2010

On the impact of Wi-Fi multimedia power save mode on the VoIP capacity of WLANs

Kwan-Wu Chin
University of Wollongong, kwanwu@uow.edu.au

Recommended Citation

Chin, Kwan-Wu: On the impact of Wi-Fi multimedia power save mode on the VoIP capacity of WLANs 2010.
https://ro.uow.edu.au/infopapers/3251

On the impact of Wi-Fi multimedia power save mode on the VoIP capacity of WLANs

Abstract

VoIP capacity is an important metric as it determines the maximum number of calls that can be supported by a Wireless Local Area Network (WLAN) before call quality degrades. To this end, researchers have conducted extensive simulation and analytical studies to determine the VoIP capacity of different WLANs. These previous works, however, assume stations are always awake during a call. In 2005, the Wi-Fi Alliance proposed a power saving mode extension that allows stations to retrieve packets from the Access Point (AP) at any time. In light of this development, this paper derives the VoIP capacity of a IEEE 802.11a WLAN where stations sleep for different time intervals. Moreover, it proposes a novel opportunistic scheduler that addresses a critical problem that arises when the power save extension is used in conjunction with a solution that improves the VoIP capacity of a WLAN by aggregating packets.

Disciplines

Physical Sciences and Mathematics

Publication Details

K. Chin, "On the impact of Wi-Fi multimedia power save mode on the VoIP capacity of WLANs," in 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA 2010), 2010, pp. 339-344.

2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops

On the Impact of Wi-Fi Multimedia Power Save Mode on the VoIP Capacity of WLANs

Kwan-Wu Chin
School of Electrical, Computer, and Telecommunications Engineering
University of Wollongong
Northfields Avenue, Australia 2522
kwanwu@uow.edu.au

Abstract— **VoIP capacity is an important metric as it determines the maximum number of calls that can be supported by a Wireless Local Area Network (WLAN) before call quality degrades. To this end, researchers have conducted extensive simulation and analytical studies to determine the VoIP capacity of different WLANs. These previous works, however, assume stations are always awake during a call. In 2005, the Wi-Fi Alliance proposed a power saving mode extension that allows stations to retrieve packets from the Access Point (AP) at any time. In light of this development, this paper derives the VoIP capacity of a IEEE 802.11a WLAN where stations sleep for different time intervals. Moreover, it proposes a novel opportunistic scheduler that addresses a critical problem that arises when the power save extension is used in conjunction with a solution that improves the VoIP capacity of a WLAN by aggregating packets.**

I. INTRODUCTION

Energy conservation is a critical issue in WLANs given the proliferation of Wi-Fi enabled, power constrained, portable devices. For example, Namboodiri et al. [1] noted that the talk time of Apple's iPhone reduces from 14 hours to eight hours when both the cellular and WLAN interface are switched on. This is not surprising as a wireless interface card draws a significant amount of power during transmission (280mA), receiving (204 mA) and idling (178 mA) [2]. In comparison, the card only draws only 14 mA when sleeping. Hence, devices must put their wireless interface card into sleep mode as long and as often as possible to extend their battery lifetime.

In legacy IEEE 802.11 WLANs, devices wake up at each beacon period to ascertain whether they have packets waiting for them via the traffic indication map. Unfortunately, the beacon interval is in the orders of hundreds of millisecond, and hence is unsuitable for VoIP calls, which typically require an inter-transmission time of 20 millisecond. To this end, in 2005, the Wi-Fi Alliance proposed a power save extension that allows a station to send a trigger to the AP at any time to retrieve packets. Note, a data frame can also be used as a trigger. Moreover, the extension supports contention free burst [3]. This reduces signaling overheads significantly as legacy IEEE 802.11 devices are required to send a poll message to retrieve each of its packet from the AP.

Figure 1 shows the operation of Wi-Fi Multmedia (WMM)'s Power Save (PS) extension [4]. Upon waking up, a station sends a trigger frame as per IEEE 802.11e's enhanced distributed channel access (EDCA) mechanism. That is, a packet's transmission priority is determined by its Arbitrary IFS (AIFS), minimum and maximum contention window size; i.e., CW_{min} and CW_{max} respectively. The AP sends an acknowledgment packet once it receives the station's trigger frame successfully. After that, the AP contends for the channel, and begins transmitting the first data packet once it wins the channel. Notice that subsequent data packets are sent immediately following an acknowledgment, thereby removing any delays associated with channel access. Each data packet also contains a "more" flag to indicate whether it is the last packet in a burst. After receiving all packets in a burst, a station puts its wireless card to sleep.

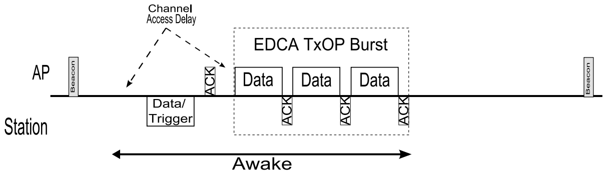

Fig. 1. Wi-FI Multimedia (WMM) Power Save (PS) extension.

The WMM-PS extension negates a key solution used to increase the VoIP capacity of a WLAN. Briefly, VoIP traffic have very high overheads and creates excessive contention. To clarify, given that VoIP packets have a small payload, the resulting overheads due to higher layer headers and MAC signaling amount to 680% if a WLAN uses IEEE 802.11b; in the best case, these overheads reduce to 200% when using IEEE 802.11g, but remain at 400% in most cases. Apart from that, real-time traffic exacerbates collisions and reduces air time as a device's rate adaptation algorithm tends to reduce its transmission rate after each collision [5]. The standard solution to reduce these high overheads is to aggregate VoIP packets into a single frame. Specifically, assuming a 20ms packetization interval, the AP multiplexes all packets that arrive in this time interval into one packet. The resulting frame is then sent as a unicast or multicast packet. As a result, instead of transmitting n different frames, the AP only needs to transmit one frame [6][5]. Unfortunately, the WMM-PS extension negates the advantages resulting from this standard solution as the AP can only aggregate packets headed to the

978-0-7695-4019-1/10 $26.00 © 2010 IEEE
DOI 10.1109/WAINA.2010.19

same station. This is because of devices' wake and sleep schedule being desynchronized, and hence, making it pointless for an AP to aggregate packets with different destinations into one packet.

Another key observation is that WMM-PS assumes a 20ms trigger interval and does not consider the delay tolerance of VoIP calls. Specifically, packets of VoIP calls can be delayed by up to handle 150ms before they experience any perceivable degradation in call quality. In other words, a device can choose to delay sending its trigger to retrieve packets from the AP, and hence allowing it to spend more time in sleep mode. Intuitively, this means a WLAN is capable of supporting more VoIP calls than if devices retrieve their packets from the AP at every packetization interval. In addition to conserving battery, sleeping provides opportunities for the devices and the AP to aggregate a higher number of packets.

In the next section, we first derive the VoIP capacity of a WLAN where all stations/devices are awake. Then in Section II-B, we used the same derivation to determine the VoIP capacity when stations sleep for a given time period. After that, in Section III, we propose an opportunistic scheduler that takes advantage of VoIP calls' delay tolerance to address the limitation that occurs when using the WMM-PS extension. In Section IV, we present the simulation methodology used to verify our analytical results and the proposed scheduler. Section V presents results pertaining to said scheduler, and Section VI concludes the paper.

II. CAPACITY ANALYSIS

In this section, we will derive the maximum number of VoIP calls supported by a IEEE 802.11a WLAN. We will consider the following cases: (i) stations remain awake at all times, and (ii) stations sleep for 20, 40 or 60 millisecond before sending the AP a trigger.

A. Stations Remain Awake

Each packet transmission incurs the following overheads. A station/device starts by transmitting a preamble (T_{pre}) followed by the Physical Layer Convergence Protocol (PLCP) (T_{PLCP}) header. The preamble is composed of 10 and two repetitions of a short and long training sequence respectively. T_{PLCP} is a single OFDM symbol in duration and also includes a SERVICE field that has a transmission time of 4μs. The duration of T_{pre} and T_{PLCP} is shown in Table I.

The next overheads are the protocol layer headers, denoted as OH_{hdr}, which comprises the RTP (12 byte), UDP (8 byte), IP (20 byte) and 802.11 (28 byte) header. This means $OH_{hdr} = 68$ bytes. Besides headers, each packet also has a four byte frame check sequence (FCS).

The size of the payload (S^{PayL}) is dependent on the codec used by the VoIP application. For example, the G.711 codec with a bit rate of 64 kbps and packetization interval of 20ms yields a payload size of 160 bytes. On the other hand, GSM 6.10 with its bit rate of 13.2 kbps generates a 33 bytes packet every 20ms.

The next set of overheads or delays are those due to MAC operation. A station first listens to the channel for an Arbitrary Interframe Spacing (AIFS) corresponding to a given traffic class (TC). For the voice TC, this is equal to DIFS [3]. After that, the station backs off for a random period before it is allowed to transmit a packet. Specifically, we have,

$$OH_{station} = DIFS + CW_{avg} \tag{1}$$

where $CW_{avg} = \text{slotTime} \times \frac{CW^{TC}_{min}}{2}$ is the average contention window (CW) size for the given TC when there are no contending stations. For the voice TC, the IEEE 802.11e specification [3] recommends $CW^{TC}_{min} = 7$ and $CW^{TC}_{max} = 15$. In our analysis, however, we used $CW^{TC}_{min} = 15$. This value better matches our simulation results as the number of devices contending for the channel exceeds seven. Note, Wang et al. [6] point out that contention overhead is negligible as compared to other overheads, and the resulting analytical VoIP capacity bound is sufficiently accurate; this is also verified by our simulation results. Similarly, for the AP, we have,

$$OH_{AP} = PIFS + CW_{avg} \tag{2}$$

where PIFS is the Point Coordination Function (PCF) Interframe Spacing.

Upon receiving a packet, a receiver then sends an acknowledgment after waiting for a Short Interframe Spacing (SIFS). To be exact, the transmission time of an acknowledgment (T_{ack}) is,

$$T_{ack} = SIFS + T_{pre} + T_{PLCP} + \frac{S^{ACK}}{D^{base}_{rate}} \tag{3}$$

where the size of the acknowledgment packet (S^{ACK}) is the sum of the MAC header (28 byte), payload (20 bytes) and FCS (4 btyes); i.e., 52 bytes. D^{base}_{rate} is the base data rate; e.g., 6 Mbps for IEEE 802.11a.

TABLE I
IEEE 802.11A PARAMETERS

Parameter	Value
slotTime	9 μs
SIFS	16 μs
PIFS	25 μs
DIFS	34 μs
PLCP Preamble (T_{pre})	16 μs
PLCP SERVICE	4 μs
OFDM Symbol	4 μs
T_{PLCP}	$8 = 4 + 4$ μs
D^{base}_{rate}	6 Mbps

Define T_{up} to be the transmission time of an uplink packet. Taking into account all the aforementioned overheads, we have,

$$\begin{aligned} T_{up} &= OH_{station} + T_{pre} + T_{PLCP} + \\ &\quad \frac{(OH_{hdr} + S^{PayL} + 4) \times 8}{D_{rate}} + T_{ack} \end{aligned}$$

Define T_{down} to be the transmission time of a downlink packet. Similar to the uplink case, we have,

$$\begin{aligned} T_{down} &= OH_{AP} + T_{pre} + T_{PLCP} + \\ &\quad \frac{(OH_{hdr} + S^{PayL} + 4) \times 8}{D_{rate}} + T_{ack} \end{aligned}$$

Lastly, define $T_{avg} = (T_{up} + T_{down})/2$ to be the average time between two consecutive packets. Hence, in one second, we have a total of $\frac{1}{T_{avg}}$ packets, or

$$\frac{1}{T_{avg}} = 2nN_p \qquad (4)$$

where n corresponds to the number of VoIP calls, and N_p is the number of packets generated by each call per second.

Solving for n in Equ. 4 using the data rates supported by IEEE 802.11a, Table II shows the VoIP capacity for popular codecs. The capacity shown are much higher than IEEE 802.11b; e.g., the authors of [6] reported 10.2 VoIP G.711 calls at 11 Mbps. The key reason for the increased in the number of VoIP calls is due to the high data rates supported by IEEE 802.11a. Note, the value reported for each data should be considered an upper bound. Indeed, the VoIP capacity obtained via simulation is slightly less than our analytical result. The main reason for this discrepancy is the number of retransmission retries afforded by a station and collisions.

TABLE II

VoIP capacity for data rates supported by a IEEE 802.11a WLAN. G.711 has a bit rate of 64 kbps, payload size of 160 bytes. G.720 runs at 8 kbps with a payload size of 10 bytes. GSM 6.10 has a bit rate 13.2 kbps and a payload size of 33 bytes.

Codec	6 Mbps	12 Mbps	24 Mbps	54 Mbps
Analytical				
G.711	21.04	31.19	41.10	49.91
G.729	36.32	45.32	51.72	56.13
GSM 6.10	32.68	42.37	49.75	55.08
Simulation				
G.711	21	29	40	51
G.729	34	43	50	54
GSM 6.10	32	42	49	55

B. Stations with WMM-PS

In the above analysis, each station sends a packet at every packetization interval. For example, the G.711 has a packetization interval of 20ms, which corresponds to 50 packets. To conserve energy, a station, can sleep for an extended period of time, thereby reducing its transmission rate. Effectively, a station conserves its energy by extending its packetization interval. The tradeoff, however, is the bigger payload size, and possible drop in voice quality as the end-to-end delay may exceed 150ms.

To study the impact of sleeping on the number of VoIP calls, we modify the analysis in Section II-A to consider extended packetization intervals. Table III shows the payload size and transmission rate resulting from extended packetization or sleep intervals.

TABLE III

Increased packetization or sleep interval on payload size and transmission rate.

Codec	*Sleep Time (ms)*	*Payload (bytes)*	*Tx Rate (pkt/s)*
G.711	20	160	50
	40	320	25
	60	480	17
G.729	20	20	25
	40	40	13
	60	60	8
GSM 6.10	20	33	50
	40	66	25
	60	99	17

Using Table III, we recalculate the number of VoIP calls in a IEEE 802.11a WLAN, but with stations sleeping for 20, 40 and 60 milliseconds. The VoIP capacity shown in Figure 2 is clearly higher than the case where stations are awake at all times. Note, we only show the result for stations transmitting at 6 Mbps because the VoIP capacity for higher data rates follows a similar trend. From Figure 2, we see that sleeping for an extra packetization interval is useful. For example, if stations using G.711 transmit a packet every other packetization interval, i.e., 40ms, the WLAN can support an additional seven calls. The main reason for the increased in capacity is due to the smaller number of packets being transmitted by each station. More importantly, the increased payload size does not negatively impact the number of VoIP calls.

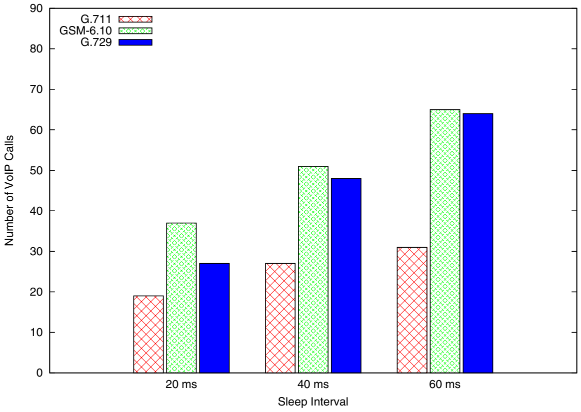

Fig. 2. VoIP capacity with WMM-PS. All stations transmit at 6 Mbps.

III. OPPORTUNISTIC SCHEDULING

A key problem when using WMM-PS is that it negates the performance benefits reported in [6] and [5]. Specifically, the AP is only able to aggregate packets on a per station basis

TABLE IV

VoIP capacity of a IEEE 802.11a WLAN using WMM-PS with all stations transmitting at 6 Mbps. We exclude the results for other data rates as they exhibit similar trend.

Sleep Interval	20ms	40ms	60ms
Analytical			
G.711	21.04	29.04	32.61
G.729	69.28	121.97	182.75
GSM 6.10	32.68	49.26	66.54
Simulation			
G.711	19	27	30
G.729	69	120	181
GSM 6.10	32	50	66

because stations have different wake-up periods. In the worst case, the AP may only transmit a frame containing one VoIP packet upon receiving a trigger from a station. Hence, all downlink transmissions will incur high signaling overheads. Fortunately, this is rare as packets have different arrival times as they traverse the Internet.

We propose an opportunistic scheduler to address the aforementioned problem. The scheduler's main aim is to group stations together such that they are able to receive an aggregated packet from the AP at a specific time. The challenge, however, is to group stations in a manner that does not violate their respective *delay budget*. Here, delay budget is defined as the remaining time before a station's packet misses its playout deadline. One can also interpret a station's delay budget as the time before it experiences a discernible drop in voice quality. For example, if we assume the end-to-end delay of a voice call to be 100ms and a tolerable delay of 150ms, the station making the call would then inform its AP that its link budget is 50ms, which accounts for the arrival time of its packet at the AP and transmission delay incurred over the wireless link.

Figure 3 gives an overview of the proposed scheduler. There are five stations, labeled A-E, each with a VoIP call. Periodically, each station sends their data or trigger to the AP in order to be forwarded to their respective peer. Each trigger also contains the station's delay budget, which is shown as a right arrow. Note, each station calculates its delay budget from the start of each trigger. The AP records each station's delay budget, and determines the best time to transmit an aggregated frame. We can see that transmitting at time t_1 enables three packets to be aggregated and is also within station A's delay budget. Similarly, transmitting at time t_2 allows the AP to aggregate two packets. Lastly, the AP informs a station the chosen transmission time, i.e., t_1 or t_2, via the ACK corresponding to the station's trigger.

Next, the AP runs Algorithm 1. In essence, the algorithm exploits the delay budget of each station to maximally aggregate packets, and thus prolongs the lifetime of stations, and in turn allow a WLAN to have a higher VoIP capacity. Initially, the AP does not have any set time to transmit a frame. Upon receiving a trigger, say from station A of Figure 3, the AP records the station's delay budget and inform the station to wake up at t_1; a time before the conclusion of the delay budget. Note, the AP needs to ensure transmission and contention delay do not cause the packet to miss its deadline. Therefore, the AP sets itself to transmit, as controlled by the parameter (δ), earlier than a station's link budget expiration time. When the next trigger arrives, e.g., from station B, it determines whether station B's delay budget exceeds t_1 and whether there is sufficient room in the aggregate packet; for stations using G.711, an IEEE 802.11 payload is able to hold approximately 14, 160 bytes packets. If so, it informs station B to wake-up at t_1. Otherwise, the AP informs station B to wake-up at a pre-defined time before the end of its delay budget. The AP carries out the same process after receiving station C's trigger. At time t_1, the AP transmits the frame containing packets belonging to station A, B and C. Note, the frame aggregation process is similar to [5]. That is, the AP prepends an aggregation header describing the set of packets and their corresponding packet length. Also, the aggregated frame is unicast to one of the stations randomly, and hence solicits only one ACK frame.

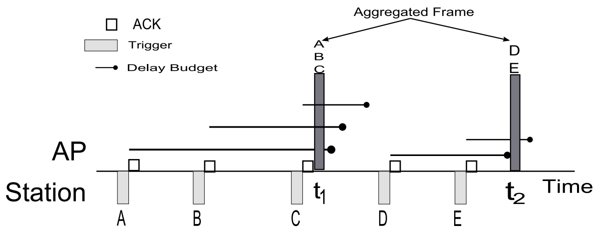

Fig. 3. Opportunistic scheduling.

IV. Simulation Methodology

We used *ns-2* (v2.33) [7] to investigate the VoIP capacity of a IEEE 802.11a WLAN. A key feature which we employed in this version of *ns-2* is the new IEEE 802.11 MAC and physical layer extensions implemented by Chen et al. [8]. Specifically, these extensions accurately model (i) the noise floor experienced by each station, (ii) transmission and processing of preamble and PLCP of each packet, and (iii) a frame reception process that considers capture when receiving either the preamble or a frame's body. Other than that, Chen et al. address the incorrect backoff and Extended IFS handling in the current IEEE 802.11 MAC implementation. In our simulation, all stations transmit using the same data rate: 6, 12, 36 or 54 Mbps. Also, there are no bit errors. This means all packet loss over the wireless channel are due to collision only. To clarify, a station discards a packet after attempting to transmit a packet three times. Another source of packet loss is when packets missed their playout time, which we set to 150 millisecond in all our experiments.

Each station emits one VoIP flow. Hence, the number of stations correspond to the number of VoIP calls. Each VoIP call consists of two Constant Bit Rate (CBR) flows; one that originates from the station, and the other from the AP. Both CBR flows start randomly, and emit a packet of a given size

```
/* Delay budget, station address. */
input : $D_{sta}$, $A_{sta}$
output: $A_{sta}$'s wake up time.
/* Initialize associative array. The
   transmission time is the index used
   to retrieve the set of stations that
   are awake to receive an aggregated
   frame. */
AggrTxTime ⟵ ∅
begin
    if AggrTxTime is empty then
        /* t is the current time. */
        AggrTxTime[$t + D_{sta} - \delta$] ∪ $A_{sta}$
    else
        /* Get the biggest key that is
           smaller than the station's
           delay budget, and is not full.
           */
        BigK = FindKey(AggrTxTime, $t + D_{sta} - \delta$)
        if found then
            AggrTxTime[BigK] ∪ $A_{sta}$
        else
            AggrTxTime[$t + D_{sta} - \delta$] ∪ $A_{sta}$
end
```

Algorithm 1: Opportunistic scheduler.

at a specific interval; e.g., the CBR flows generate a 160 bytes packet every 20ms for experiments involving the G.711 codec. To study the impact of varying network delays, we randomly delay the packets originating from the AP by 50 to 150 millisecond. Lastly, we determine the VoIP capacity by increasing the number of VoIP calls or stations until all calls experience a 1% packet loss.

V. RESULTS

Figure 4 shows the VoIP capacity of a WLAN using our scheduler. Here, we experimented with stations sleeping for 20, 40 and 60 millisecond. The VoIP capacity for each data rate is higher than the value shown in Table II. This is mainly due to the benefits of packet aggregation, which has the effect of reducing the load at the AP, and hence minimizing the number of packets that had to be dropped after missing their playout time. This can be seen on Figure 5, which shows the number of messages transmitted by the AP. We can see several orders of magnitude reduction in the number of packets transmitted by the AP. If the AP only uses WMM-PS as is, it is still able to aggregate packets on a per-station basis. However, the AP has many more opportunities, and hence more savings, if it is allowed to aggregate packets headed to different stations. From Figure 4, we also see the resulting VoIP capacity when stations sleep for a longer period of time. In other words, instead of sleeping 20ms before sending a trigger, they send a trigger every 40ms or 60ms. The net effect of this is that there is less channel contention, thereby allowing the AP and other stations that are awake to send their respective packets. In some cases, we see a 20% jump in the number of VoIP calls supported by the WLAN if stations sleep 20ms longer.

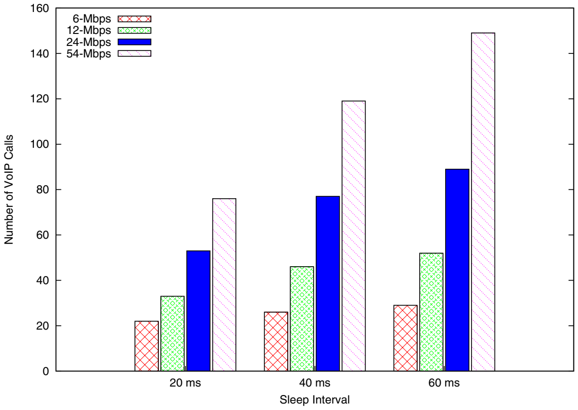

Fig. 4. VoIP capacity with opportunistic scheduling using the G.711 codec.

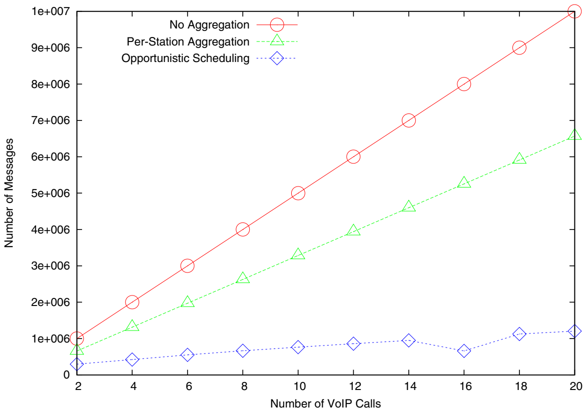

Fig. 5. Number of messages transmitted with and without opportunistic scheduling in 10 simulation seconds.

VI. CONCLUSION

This paper is the first to study the impact of Wi-Fi's multimedia power saving extension on the VoIP capacity of WLANs. Our analysis and simulation results show that with every additional 20 millisecond of sleep time, the VoIP capacity of a WLAN increases by approximately 20%. Apart from that, we identified a key problem that arises when this extension is used with a solution that aggregates VoIP packets in order to reduce packet overheads. To address this problem, we proposed an opportunistic scheduler that aggregates packets and groups stations so that they wake at the same time. Our results show the proposed scheduler to be very effective in reducing the load of the AP, and thereby, allows a WLAN to admit more calls.

REFERENCES

[1] V. Namboodiri and L. Gao, "Towards energy efficient VoIP over wireless LANs," in *ACM MobiHoc*, (Hong Kong, China), ACM, May 2008.

[2] L. M. Feeney and M. Nilsson, “Investigating the energy consumption of a wireless network interface in an ad hoc networking environment,” in *IEEE INFOCOM*, (Alaska, USA), May 2001.
[3] IEEE, “Part 11: Wireless lan medium access control (MAC) and physical layer (phy) specifications amendment 8: Medium access control (mac) quality of service enhancements.” IEEE Std 802.11e-2005, 2005.
[4] WiFi Alliance, “WMM power save for mobile and portable Wi-Fi CERTIFIED devices,” Dec. 2005.
[5] P. Verkaik, Y. Agarwal, R. Gupta, and A. C. Snoeren, “Softspeak: Making VoIP play well in existing 802.11 deployments,” in *6th USENIX Symposium on Networked Systems Design and Implementation*, (Boston, USA), USENIX, April 2009.
[6] W. Wang, S. C. Liew, and V. O. Li, “Solutions to performance problems in VoIP over a 802.11 wireless LAN,” *IEEE Transactions on Vehicular Technology*, vol. 54, pp. 366–376, January 2005.
[7] “The Network Simulator NS-2.” http://www.isi.edu/nsnam/ns/.
[8] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, “Overhaul of IEEE modeling and simulation in ns-2,” in *ACM MSWiM*, (Crete Island, Greece), ACM, October 2007.