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

Follow this and additional works at: https://ro.uow.edu.au/infopapers

Part of the Physical Sciences and Mathematics Commons

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

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
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

This conference paper is available at Research Online: https://ro.uow.edu.au/infopapers/3251
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 milliseconds, and hence is unsuitable for VoIP calls, which typically require an inter-transmission time of 20 milliseconds. 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 arrivals in this time interval into one packet. The resulting frame is then sent as a unicast or multicast packet. As a result, the extension negates the advantages resulting from this standard solution as the AP can only aggregate packets headed to the

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
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,

$$OH_{station} = DIFS + CW_{avg}$$

where $CW_{avg} = \text{slotTime} \times \frac{CW_{TC}^{max}}{2}$ 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,

$$OH_{AP} = PIFS + CW_{avg}$$

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,

$$T_{ack} = SIFS + T_{pre} + T_{PLCP} + \frac{S^{ACK}}{D_{rate}^{base}}$$

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>slotTime</td>
<td>9 μs</td>
</tr>
<tr>
<td>SIFS</td>
<td>16 μs</td>
</tr>
<tr>
<td>PIFS</td>
<td>25 μs</td>
</tr>
<tr>
<td>DIFS</td>
<td>34 μs</td>
</tr>
<tr>
<td>PLCP Preamble ($T_{pre}$)</td>
<td>16 μs</td>
</tr>
<tr>
<td>PLCP SERVICE</td>
<td>4 μs</td>
</tr>
<tr>
<td>OFDM Symbol</td>
<td>4 μs</td>
</tr>
<tr>
<td>$T_{PLCP}$</td>
<td>8 = 4 + 4 μs</td>
</tr>
<tr>
<td>$D_{rate}^{base}$</td>
<td>6 Mbps</td>
</tr>
</tbody>
</table>

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

$$T_{up} = \frac{OH_{station} + T_{pre} + T_{PLCP} + (OH_{hdr} + S^{PayL} + 4) \times 8}{D_{rate}^{base}} + T_{ack}$$
Define $T_{down}$ to be the transmission time of a downlink packet. Similar to the uplink case, we have,

$$T_{down} = \frac{OH_{AP} + T_{pre} + T_{PLCP} + \left( OH_{hdr} + S_{PayL} + 4 \right) \times 8}{D_{rate}} + T_{ack}$$

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

$$\frac{1}{T_{avg}} = 2nN_p$$

(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**

<table>
<thead>
<tr>
<th>Codec</th>
<th>Sleep Time (ms)</th>
<th>Payload Size (bytes)</th>
<th>Tx Rate (pkts/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.711</td>
<td>20</td>
<td>160</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>320</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>480</td>
<td>17</td>
</tr>
<tr>
<td>G.729</td>
<td>20</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>40</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>60</td>
<td>8</td>
</tr>
<tr>
<td>GSM 6.10</td>
<td>20</td>
<td>33</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>66</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>99</td>
<td>17</td>
</tr>
</tbody>
</table>

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.

**TABLE III**

<table>
<thead>
<tr>
<th>Codec</th>
<th>Sleep Time (ms)</th>
<th>Payload Size (bytes)</th>
<th>Tx Rate (pkts/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.711</td>
<td>20</td>
<td>160</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>320</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>480</td>
<td>17</td>
</tr>
<tr>
<td>G.729</td>
<td>20</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>40</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>60</td>
<td>8</td>
</tr>
<tr>
<td>GSM 6.10</td>
<td>20</td>
<td>33</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>66</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>99</td>
<td>17</td>
</tr>
</tbody>
</table>

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
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 remaining 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 informs the station to wake up at a time before the conclusion of the delay budget. Therefore, the AP sets itself to transmit, as controlled by the parameter \( \delta \), 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.

### IV. Simulation Methodology

We used \textit{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 \textit{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
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
      AggrTxTime[t + D_{sta} − δ] ∪ A_{sta}
  else
      BigK = FindKey(AggrTxTime, t + D_{sta} − δ)
      if found then
          AggrTxTime[BigK] ∪ A_{sta}
      else
          AggrTxTime[t + D_{sta} − δ] ∪ 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.

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


