Experimental VoIP Capacity Measurements for 802.11b WLANs

Moncef Elaoud, David Famolari, and Ahbrajit Ghosh
Telcordia Technologies- Applied Research
One Telcordia Drive, Piscataway, NJ 08854
Voice: +1-732-699-3132
[moncef, fam, aghosh]@research.telcordia.com

Abstract: There is an increasing interest in supporting Voice over IP (VoIP) applications over Wireless Local Area Networks (WLANs). To provide quality of service guarantee to voice traffic, an admission control mechanism must be administered to avoid overloading the network. In this paper, we present an experimental evaluation of VoIP capacity in WLANs under various data and voice loads. The obtained capacity estimates can be used in admission control engine to make an admission or a rejection decision.

In this paper, we identify a suitable evaluation metric to quantify the performance of a voice call. The metric accounts for packet losses and packet delays for each voice flow. We then define voice capacity in a WLAN and present experimental capacity measurements under various background data traffic loads. The experimental capacity measurements are immediately useful for WLAN hotspot providers and enterprise WLAN architects.

I. INTRODUCTION

Wireless Local Area Networks (WLANs) have made a huge impact inside and outside the enterprise environment. Increasingly, corporations are turning towards WLANs to network their offices and campuses. Similarly, the general consumers are increasingly installing WLAN equipment in their homes to network their home computers and appliances. Furthermore, there is a current rush to populate public spaces and commercial areas with “hotspot” WLAN coverage. While a majority of anticipated traffic in these environments will be data, there is a growing interest in supporting Voice over IP (VoIP) applications over WLANs. This type of arrangement offers reduced installation costs, increased deployment flexibility, and greater consolidation of telecommunication resources.

To provide a viable voice service over IP over WLAN, the network must administer admission control on flows that require quality of service assurance. Determining the network capacity is of paramount importance for any admission control algorithm. In this paper we present an experimental study of typical WLAN voice capacity under various data traffic loads. These results are immediately useful to admission control and network management and provisioning. First, we present metrics for defining supported voice calls that account for both packet losses and packet delays and use those metrics to experimentally measure capacity in a real-world WLAN. Second, capacity is evaluated both with and without simultaneous data traffic. Lastly, we describe our experimental methods, software and test control innovations that have enabled us to create a flexible, reliable, and automated test environment.

The remainder of this paper is organized as follows. Section II describes previous work. Section IV presents our definition of voice capacity in WLANs, including the metrics to evaluate whether or not a voice call is supported. Section V describes our experimental testbed. Experimental results are presented in section VI, while Section Error! Reference source not found. concludes the paper.

II. ACCESS MECHANISM IN IEEE 802.11

The IEEE 802.11 provides two modes of access namely the required Distributed Coordination Function (DCF) and the optional Point Coordination Function PCF [17] [18]. The DCF access mode is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) family of medium access protocols. These MAC protocols are designed to avoid collisions through a “listen-before-talk” procedure. Terminals must first determine if the medium is free before attempting to transmit. To account for the vagaries of the wireless channel and the possibility of packet loss, MAC frames must be positively acknowledged by receiving stations. Transmitting stations that do not receive such an acknowledgement consider the transmitted frame lost and retransmit.

A station that has packets to transmit first senses the medium. If the medium is determined to be free for a continuous period equal to a Distributed Inter Frame Spacing (DIFS), the station transmits the packet. Otherwise the station continues to monitor the medium until that condition is met. At this time the station enters a random backoff phase in which it chooses a random backoff timer uniformly from a collection of values known as the contention window. If multiple stations want to transmit a packet, the station with the lowest random backoff timer wins the contention for the medium.

During the backoff procedure, the station continues to monitor the medium and for every idle timeslot decrements the backoff timer. If the medium becomes busy during the countdown, the station suspends the decrement operation until the channel becomes idle again for a period of DIFS. When the backoff timer reaches zero the station transmits its packet. After the completion of a packet transmission, successfully or
unsuccessfully, the transmitting station enters a post-transmission random backoff procedure. After every unsuccessful packet transmission the size of the contention window of the failing station, and thus the average waiting time, doubles until it reaches its maximum value. However, following a successful transmission the contention window range is reset to its minimum value.

III. RELATED WORK

Much of the previous work either covered the design of MAC protocols for supporting voice traffic as in [3] and [4], or focused on the performance evaluation of voice flows in IEEE 802.11 PCF operating mode, as in [5] [6] [7] and [8]. Evaluating the performance of voice flows and the network voice capacity in the DCF mode are virtually undocumented.

Most of the performance evaluations of the voice in the PCF mode were carried through simulations. It is not until the last few years that experimental studies have made their way into the technical literature. In [1] the authors compare experimental voice capacity performance to that predicted by analysis. The experimental setup comprised of commercial equipment, used a voice codec rate of 10ms and each conversation contained a wired and a wireless participant. The experiment revealed that 6 such conversations could be reliably supported, but that when the 7th call was added, performance for all flows in the wired-to-wireless direction suffered unacceptable losses and delays. Thus the authors concluded that capacity for a 10 ms codec is 6 terminals, however, they do not detail exactly what performance metrics constitute an “acceptable” voice call. In addition, they do not consider the impact of background data traffic on voice calls.

In contrast, [10] describes a quantifiable metric to evaluate the performance of voice in WLAN networks. However, the metric was based solely on packet loss due to collisions on the wireless channel. The capacity results presented in [10] did not account for delays encountered by voice packets. This results in overestimating the network voice capacity which may lead to over admitting voice users resulting in quality of service deterioration for the admitted flows. In the present work, we describe a testbed architecture and control procedures that allow us to accurately measure packet losses and one-way packet delays. This enables a more detailed measurement of voice capacities that incorporates both packet losses and delays.

IV. VOIP CAPACITY

Voice over IP is a time and loss critical application where a certain percentage of data packets must be delivered within a certain maximum tolerable delay. Standards bodies have suggested that a voice flow can sustain up to 2% packet loss and tolerate up to 200 milliseconds of one-way end-to-end delay and still deliver acceptable voice quality [12]. Packets that are delivered beyond the application specific deadline usually contain stale information and are generally dropped. Therefore, from the application perspective, there is no distinction between late arriving packets and lost packets. As such, the fraction of late packets must also be accounted for in calculating voice capacity. To accomplish this, we assume that each voice packet possesses a deadline by which it must be delivered. Only those packets that arrive prior to their deadlines are considered successfully delivered, while packets received afterwards are considered lost. We introduce a performance metric called the Packet Success Ratio (PSR) that measures the percentage of voice packets that are “successfully” delivered to a voice application. The PSR value details what is required for a single voice flow (or call leg) to be supported and is the basic metric for calculating capacity in our studies.

Deriving perceived voice quality based on objective traffic measurements is, in general, difficult. Some complex models exist, such as Perceptual Evaluation of Speech Quality [16] and others based on the ITU E-Model [17], that attempt to account for transients and temporal correlations in channel performance. However, as a first approximation, we have chosen a simple, straightforward model based solely on per-call packet loss and per-packet delay. Our data collection methodologies, however, do not preclude further refinement of the results via the above perceptual models and such a comparison is scheduled for future work.

Generally, voice conversations contain two call legs: one in each direction. In a WLAN environment, a voice call possesses a call leg in the uplink (terminal-to-AP) direction and one in the downlink (AP-to-terminal) direction. In such an environment, we consider a voice call to be supported if and only if both legs of the call have an acceptable PSR.

Based on this definition of a supported call, we define WLAN voice capacity as the total number of simultaneous calls that can be supported. More precisely, we define voice capacity to be the first point at which adding an additional call to the system results in at least one call not being supported. In other words, the over-capacity is defined as the first point at which the system can no longer support the entire voice population. Defining capacity in this fashion ensures that all users in a WLAN operating within capacity limits are supported. Further, this definition naturally fits with notions of system fairness and can be easily integrated into typical admission control policies.

To quantify WLAN voice capacity we need to establish a delay budget for voice. There are two kinds of delays in telecommunication networks: medium propagation delays and handling delays. Medium propagation delays, being on the order of nanoseconds, are relatively insignificant when compared with handling delays and are not considered in our analysis. Handling delays are influenced by a variety of factors including, coding/decoding at codecs, endpoint packetization, core network traversal, and wireless network access. In our experimental setup we attempt to isolate the wireless network access delays which begin the instant a voice packet is handed to the MAC layer (at either the AP or a wireless station) and ends when that packet is received by the corresponding wireless MAC layer. This delay comprises both queuing delays at the MAC and medium access delays.

Based on the one-way end-to-end delay requirements of a voice flow, we calculate an upper bound for the tolerable wireless access delay for various networking environments. We use the worst-case scenario as our baseline, where both ends of the voice communication are connected to a wide-area network through wireless hops. Voice packets in such a
conversation encounter the following delays: coding/decoding delays, δc, at both users, packetization delays, δp, at the sender, core network traversal delay, δCN, and wireless access delays, δWN, to access each wireless network. Thus, the one way end-to-end delay, ΔOW, encountered by the packet may be expressed as shown in Equation (1).

\[ Δ_{OW} = 2δ_c + δ_p + δ_{CN} + 2δ_{WN} \]  

We define \( \bar{D} \) as the maximum tolerable one-way delay or deadline by which a voice packet must arrive at its destination to be considered acceptable by the application. Then, an upper-bound for \( δ_{WN} \) can be calculated by replacing \( Δ_{OW} \) with \( \bar{D} \) and rearranging the terms in Equation (1). 

\[ δ_{WN} \leq \frac{\bar{D} - 2δ_c - δ_p - δ_{CN}}{2} \]  

Further, if we assume \( \bar{D} = 200 \text{ ms} \), as defined in [12], and the coding delays \( δ_c \) are limited to 5 milliseconds; we can express \( δ_{WN} \) as follows:

\[ δ_{WN} \leq 95 - \frac{δ_p - δ_{CN}}{2} \]  

In this study, we consider both Local Area Network (LAN) and Wide Area Network (WAN) scenarios. For the LAN analysis we assume no appreciable core network delay and for the WAN analysis we assume a core network delay of 50 ms. Solving Equation (3) for a 10 ms codec packetization value and core network delays of 0 and 50 ms yields a LAN budget of 90 ms and a WAN budget of 65 ms, respectively.

V. EXPERIMENTAL SETUP

A. Testbed Architecture and Hardware

The experimental testbed shown in Figure 1. contains three IP sub-networks: the IEEE 802.11b wireless network under study and two wired sub-networks. The wireless sub-network includes 15 laptop machines. Each laptop connects to the wireless network using an Orinoco Gold PCMCIA card. The wireless interfaces on the laptops use IP addresses in the range 192.168.1.x. In addition, each of the laptops is connected, via a second interface, to a wired sub-network used for control purposes. The wired sub-networks are joined using a Cisco AP350 Access Point. An NTP server provides time synchronization between all the machines in the testbed. We developed scripts to ensure that the time drift for any machine from the NTP server is kept within 500 microseconds.

Lastly, a controller is placed on the control network and is responsible for issuing experiment commands that start and stop trials, configure new experimental parameters and test the availability of wireless network interface cards. During the course of experimentation, our experience showed that some cards could become inactive and stop responding to commands. The wired control network facilitates an automated restarting of these disabled cards and provides other helpful diagnostic information.

B. Testbed software

All machines in the testbed have the RedHat 7.2 version of Linux running kernel version 2.4.7-10. The testbed machines contained two additional applications: MGEN and VGEN. VGEN is an in-house voice generation tool that produces network traffic corresponding to conversational speech in compliance with ITU-T recommendation P.59 [11]. VGEN instances work in pairs, one instance runs as a master while the other as a slave. Each instance simulates a participant in a two-party voice call. VGEN executes a four-state Markov model including the following states: (a) Master talking, Slave silent, (b) Master silent, Slave talking, (c) both talking, and (d) both silent. The durations of these states are Exponential random variable with means 854 ms, 854 ms, 226 ms and 456 ms, and the steady state probabilities are 0.25, 0.25, 0.3, and 0.2, respectively.

VGEN tracks the distribution of one-way packet delays and feeds the results gathering process. All VGEN control and results traffic is sent over the wired control network via source routing. For further details on VGEN and the voice model, we refer the reader to [10] and [11].

MGEN is an open-source software package from the Naval Research Laboratory [13] that is used to generate UDP data traffic. To evaluate the effects of background data on voice capacity, we generated downlink data traffic from the Core Network into the WLAN. We used a 512 byte MGEN packet to emulate typical HTTP packet size since HTTP based data downloads are typical of WLAN hotspot activity.

C. Evaluation Metrics

Since our testbed represents a small network, we can approximate the wireless access delay \( δ_{WN} \) by the total one-way delay experienced by a packet. This assumption is
justified since the wired network is a 100 Mbps Ethernet, which makes the delays within the wired segments negligible compared to the queuing and access delays in the wireless network. Thus, one-way delays measured this way closely approximate $\delta_{WN}$. Therefore, we can use the 65 ms and 90 ms values arrived at earlier for the WAN and LAN deadlines.

During the experiments, we maintained a metric called the Packet Loss Ratio (PLR) that represents the percentage of packets that never arrived at the destination. The loss of packets in the network could be the result of either buffer overflow at the transmitting stations or due to wireless collisions and/or channel effects. We also measured the one-way delay for all voice packets traveling in the network. Packets that arrive at the destination prior to their deadlines are considered successful. All other packets are considered unsuccessful. Hence the PSR is calculated as the percent of all transmitted packets that reach their destination prior to their deadline.

VI. EXPERIMENTAL RESULTS

The results presented in this paper were collected from a set of experiments in which a downlink data rate was selected from the following set of values: 0, 1, 2, and 4 Mbps. Each experiment consisted of 10 trials. In each trial we varied the number of bidirectional voice calls from 1 to 15 to determine the network voice capacity under each of the data loads. Voice calls were established between wireless hosts and wireline counterparts. Each call lasted 3 minutes and used the G711 voice codec with a packetization of 10 milliseconds. Results were collected after each trial and tabulated to determine the voice capacity under the various core network deadline assumptions. Finally, the capacities for each experiment were averaged across trials to arrive at the final number of supported calls.

The introduction of downlink data significantly impacted capacity even without considering delay budgets. This is illustrated by the PLR data in Figure 3. Since voice enjoys no priority with respect to data, a number of voice packets are lost as a result of overflows in access point queues [10]. Thus, while about 10 calls can be supported in the absence of downlink data, only about half that number can be supported in the presence of 2 Mbps downlink data. Additionally, once the data load is increased to 4 Mbps, no voice calls are supportable.

Delay budgets further reduce the number of supported calls in a mixed voice and data system. Consider the case of 2 Mbps downlink data and a 98% PSR threshold shown in Figure 3. Here, for both the LAN and the WAN deadline, the number of supported calls drops from more than 6 to less than 4. Furthermore, no calls can be supported alongside a 4 Mbps downlink data flow at any deadline or PSR threshold.

The capacity measurements are a result of averaging over multiple, independent trials. Therefore, it is natural to investigate the level of confidence associated with the results.

<table>
<thead>
<tr>
<th>Data Rate (Mbps)</th>
<th>Avg. Capacity (95% Confidence Interval)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Downlink</td>
</tr>
<tr>
<td>0</td>
<td>6.9 (6.71, 7.08)</td>
</tr>
<tr>
<td>1</td>
<td>3.7 (2.91, 4.49)</td>
</tr>
<tr>
<td>2</td>
<td>2.8 (2.34, 3.26)</td>
</tr>
</tbody>
</table>

Delay budgets further reduce the number of supported calls in a mixed voice and data system. Consider the case of 2 Mbps downlink data and a 98% PSR threshold shown in Figure 3. Here, for both the LAN and the WAN deadline, the number of supported calls drops from more than 6 to less than 4. Furthermore, no calls can be supported alongside a 4 Mbps downlink data flow at any deadline or PSR threshold.

The capacity measurements are a result of averaging over multiple, independent trials. Therefore, it is natural to investigate the level of confidence associated with the results.

TABLE I. compiles the average capacity, and the 95%
The table verifies that the downlink is the limiting factor in determining WLAN voice capacity. Throughout the data ranges considered, the uplink outperformed the downlink. This discrepancy is largest in the 0 and 1 Mbps downlink data cases, in which the uplink supported close to 2 more users than the downlink. We can attribute this to the highly asymmetric loads present in a WLAN voice system. All downlink data packets must pass through a single AP queue, yet the AP competes for the medium on roughly equal footing with other stations. Therefore, the AP can empty the downlink queue at roughly the same rate that the wireless stations can empty their uplink queues. However, the downlink queue fills at a much faster rate and excessive queue buildup at the AP is inevitable. Thus downlink packets are more readily extended beyond their deadlines resulting in small PSR metrics. The queue buildup at the AP, and the resultant downlink packet delays, are the limiting factors for WLAN voice capacity.

TABLE I. also shows that the experimental method obtains acceptable confidence levels in terms of capacity, with the 95% confidence intervals smallest for the case of no downlink data. Introducing data increases these intervals, but the resulting confidence is still acceptable for quantifying voice capacity. This larger variability is due to a larger standard deviation of voice capacity in the presence of data flows. In fact, we observed that downlink data significantly increases the standard deviation of downlink voice delays, while having little or no effect on uplink voice delays. This results because the AP, despite the increased load, does not achieve a significantly higher share of the medium and thus does not disrupt the variability of uplink packet delays.

The performance difference between the uplink and the downlink is further illustrated in Figure 4., which plots the cumulative density functions (CDFs) of voice packet delays for various numbers of simultaneous users and no background data traffic. In the figure, the curves are labeled Tx, where x is the number of voice users in the network. When the system is operating within capacity limits, the CDF curves for both the uplink and the downlink are heavily concentrated in the upper-left, indicating high PSR values. However, as the system attempts to accommodate more users, the CDFs begin a downward shift to the right, with the downlink separating at a faster rate. At eight simultaneous users (T8) a noticeable separation has occurred in the downlink indicating that capacity has been exceeded. Further addition of users quickly deteriorates the packet delay CDFs. The uplink, however, resists separation until well after eight simultaneous users.

VII. CONCLUSION

In this work, we described a quantifiable performance metric for voice sessions that accounts for both packet losses and delays. Using this metric, we presented a definition for WLAN voice capacity as the first voice population at which adding an additional voice user results in at least one user not being supported. We then presented experimental capacity measurements with and without background data traffic. We evaluated the impact of delay deadlines on the uplink and downlink call legs as well as on the overall system capacity.
End-to-end Quality of Service in TIPHON systems; Part 1: General aspects of Quality of Service (QoS)


