Abstract—Auto rate fallback (ARF) is a link adaptation algorithm that uses layer-2 acknowledgement messages to make decisions to increase or decrease the transmission rate. The algorithm is configured by choosing thresholds to define its behavior. The ARF algorithm is applied to voice-over-IP users in an IEEE 802.11a wireless LAN system, and system simulation is used to determine the optimal ARF configuration. Configuration performance is judged using a unique voice quality metric, which is used to establish a measure of system capacity for a single access point with several voice users. Results indicate that the ARF configuration can significantly impact capacity, and one configuration of ARF provides a capacity within 13% of the available capacity.

Index Terms—Link Adaptation, voice, IEEE 802.11, WLAN

I. INTRODUCTION

Wireless LAN (WLAN) technology has become very popular for mobile and personal communication in indoor environments. The IEEE 802.11a standard permits eight transmission rates, between 6 and 54 Mbps, but does not specify a method for choosing the rate. This process, when performed adaptively based on changing channel conditions, is called link adaptation (LA). Stations benefit from employing LA by using higher transmission rates: their throughput increases (decreasing network congestion and increasing system capacity), while their transmission times are reduced (reducing delay and improving battery life). In this paper, we consider the configuration of a pure layer-2 LA algorithm [1], when the WLAN system is used for a voice application.

Traditionally, LA algorithms have been directly associated with physical layer metrics, such as received signal strength indicator (RSSI) and signal to noise ratio (SNR) measurements. Within the context of a contention-based IEEE 802.11 WLAN system, [2] and [3] have studied the performance of LA algorithms that solely base decisions on RSSI. In [4], the performance of a LA algorithm that solely bases its decision on SNR was studied.

At times, it is either necessary or desired that the LA decision be made at the data link layer, so relying on physical layer metrics like RSSI or SNR may not always be practical. Recognizing this shortcoming, there is another branch of LA research that utilizes the presence and absence of a positive acknowledgement (ACK) message as an implicit channel quality indicator. Since insufficient channel quality can cause the absence of an ACK message, [1] proposes the idea of reducing the transmission rate when a sufficient number of expected ACK messages have been missed. Similarly, the presence of a consecutive number of successful ACK messages can be used to indicate that the channel quality is good enough to support a higher transmission rate. Such an ACK-based scheme, called auto rate fallback (ARF), was used in an early WLAN system [5]. In [6], A.J. van der Vegt extends the ARF concept to include a probation state, which expedites the transmission rate fallback when it appears that the rate increase decision is unjustified. Finally, [7] considers a case where the step-up threshold (see Section II) is changed dynamically in response to an estimate of the channel’s Doppler spread.

Configuring ARF consists of choosing the thresholds for the number of received or missed ACKs required to change the rate. ARF configurations chosen in [5], [6], and [7], for example, seek to maximize data throughput in a static environment, possibly with limited hidden terminals (we consider data traffic in Section VI-E). In this paper, we evaluate ARF configurations in an effort to maximize system capacity when the primary application is voice telephony.

System capacity is defined as the number of voice users who can complete acceptable calls. A unique voice quality metric estimates call quality supplied to individual voice users. The metric is a simple, objective estimate of quality based only on the amount and pattern of packet loss during a voice call; it judges calls as acceptable or unacceptable. By modeling the system using a discrete event simulation tool, we can estimate the number of users who can experience acceptable calls given a particular system configuration.

The remainder of this paper is organized as follows. We describe the ARF algorithm and the LA algorithm used at the access point in Section II, followed by details of our model for voice users and a description of the simulation environment. Section IV provides a definition of system capacity and the voice quality metric used to determine it. We provide analysis of the impact of ARF parameter choices in Section V, and then give results from our simulation-based search. Finally, we provide concluding remarks in Section VII.

II. LINK ADAPTATION ALGORITHMS

The basic task of LA is to choose a transmission rate from a set of allowed rates based on channel conditions. In IEEE 802.11a, the possible data rates are 6, 9, 12, 18, 24, 36, 48, and 54 Mbps. Here, we describe the ARF algorithm in detail, and describe a LA technique suitable for an access point (AP) when stations use ARF.
A. Auto Rate Fallback

We assume that changes in rates occur only by moving up or down one link rate. For example, if the current transmission rate is 24 Mbps, the next transmission will occur at 18, 24, or 36 Mbps. The algorithm is summarized as follows [6]:

- If a pre-defined number (the step-up threshold, \( N_s \)) of consecutive ACK messages is seen, then the transmission rate will be increased by one step.
- If a pre-defined number (the fallback threshold, \( N_d \)) of consecutive ACK messages is missed, then the transmission rate will be decreased by one step.
- A station enters a probation state upon switching to a higher rate. If any of the next \( N_p \) ACK messages are correctly received, the station stays in the current transmission rate; otherwise, the transmission rate is reduced to the previous, slower rate. Receiving at least one ACK in the next \( N_p \) ACK messages indicates that the channel can support the rate increase decision, allowing it to exit the probation state. It is expected that \( N_p \) is set such that \( N_p < N_d \).

The probation state is an important feature of the ARF algorithm. It allows the algorithm to recover quickly from a poorly made rate-increase decision. Note that [6] describes the probation state but assumes \( N_p = 1 \), which is the most conservative setting. The notation \( N_p = 0 \) indicates that the probation state is not used.

B. Link Adaptation at the Access Point

Each mobile user (MU) will implement the LA algorithm to determine uplink (MU to AP) transmission speeds. At the AP, there are several options for choosing downlink transmission link speeds. For example, the AP may always match the rate that the MU has chosen. The AP may also implement the LA algorithm for each MU (although storage and processing requirements must be considered).

We configure the AP to choose a transmission link speed based on an “AP fallback” algorithm. For communication to a given MU, the AP initially sets its transmission speed to that used by the MU for its last successful transmission. With every retransmission that the AP must perform, the link speed is decreased by one step. For broadcast packets, the AP always uses the lowest link speed.

III. MOBILE USER MODEL

A discrete event simulation tool [8] is used to consider various configurations of the ARF algorithm and how they impact performance. In an effort to exercise the LA algorithms, a user mobility model was developed to maintain a uniform distribution of users throughout the cell’s area, while moving individual users through various degrees of coverage. A description of the simulation environment and settings is also given here.

A. User Motion

Users are initially placed uniformly throughout the cell’s area, which is a circle with a radius of 18 meters (the approximate range from the AP where reliable coverage ends). MUs are configured to move in a spiral fashion; unlike [9], their initial direction is randomly chosen to be either spiraling inward or outward. When a MU reaches the AP or the outer boundary, it begins spiraling in the opposite direction. An example user trajectory is shown in Fig. 1; the user’s starting location is circled, and the AP is located at the origin.

Based on considerations from the mobility model, all simulations in this report have duration 64 s.

B. Simulation Environment

A discrete event simulation tool was used to model an IEEE 802.11a WLAN system consisting of several MUs and an AP connected via a radio link. MUs run a voice-over-IP application that communicates with a voice gateway on the wireline network.

Voice stations use the unscheduled power save delivery (UPSD) access method [8], which is included in the IEEE 802.11e draft. UPSD is a novel power management scheme for voice-over-IP devices, allowing the MU to briefly exchange a voice frame with the AP, and quickly return to a low-power sleep mode. The voice model is as in [8], but the maximum lifetime of voice packets is 30 ms for both uplink and downlink. Additional WLAN settings are provided in Table I.

Link adaptation depends significantly on the underlying channel conditions, so accurate modeling of the expected

1Users avoid a small circular region (with radius about 3 m) at the cell’s center to prevent the radial velocity from exceeding 1.5 m/s, the constant linear velocity of all users.
deployment scenario is important. Both fast fading and lognormal shadowing are used, and parameters are given in Table II. Receiver selection diversity is disabled at all stations and the AP, so that the attenuation experienced in the downlink and uplink are more alike than in the case with receiver diversity.

IV. SYSTEM CAPACITY

We assume that a system’s failure rate, the probability that a voice user will encounter an unacceptable call, depends on the number of users, \( n \), present in the system, and we denote this probability \( p_f(n) \). In our simulations, we assume that \( p_f \) is an increasing function of \( n \), and we define a system capacity

\[
C(\epsilon) = \max\{n \mid p_f(n) < \epsilon \}.
\]

So, \( C(\epsilon) \) gives the maximum number of users that can be supported with a failure rate less than \( \epsilon \).

In this section, we provide a description of our voice quality metric [9], followed by our method of estimating system capacity from simulations. Our quality metric allows for a simple estimate of capacity to be used as a performance measure of algorithms such as LA.

A. Voice Quality Metric

From the perspective of a single MU, voice information is received as a stream of packets (spaced \( \tau = 20 \text{ ms} \) apart, in our formulation). A gap in coverage, poor link speed choices, system capacity limitations, and many other factors may cause the downlink packet stream to be corrupted. Each packet is either received correctly and acknowledged, or is treated as being completely lost (there are no partially received packets).

We judge call quality or acceptability by first calculating two scores. The first score, \( L \), is simply the number of lost packets (at the voice application layer) throughout the call. The second, \( E \), estimates the number of audible artifacts or interruptions in the audio.

We estimate audible events, \( E \), by assuming a packet loss concealment algorithm is operating. When a burst loss of \( l \) consecutive packets occurs, our model assumes that an audible imperfection is heard with probability

\[
p(n) = \begin{cases} 
0.1l + 0.3 & 1 \leq l \leq 4 \\
1 & l \geq 5 
\end{cases}
\]

When a packet loss occurs in a burst of length \( l \), we increment \( E \) by \( p(l) \).

However, when several burst losses of lengths larger than 5 packets occur, there is a potential for artificially inflating \( E \). For example, consider a pattern of 12 lost packets, followed by 2 received packets, followed by 9 lost packets. The two received packets (40 ms of audio) will not contribute to good audio quality; this should be counted as a single event. Thus, when a burst loss occurs with length \( l \geq 5 \), \( E \) is incremented by 1, but \( E \) will not be incremented again until ten consecutive packets are received (200 ms of uninterrupted audio).

Using the \( L \) and \( E \) scores, good calls must satisfy the following tests (where the call duration is labeled \( T \)):

1) **Intelligibility:** \( \frac{L}{T} < \frac{0.2}{\epsilon} \), which requires that, on average, an error event should consist of a burst loss of length less than an average syllable (200 ms), which corresponds to 10 packets in our scenario (\( \tau = 20 \text{ ms} \)).

2) **Audible Losses:** \( E < \frac{T}{5} \), which requires that, on average, the spacing between audible error events be 5 s; this limits the number of audible events to about 12 in a one-minute call.

Note that, with these tests, the overall packet loss rate \( L_T \) is limited to 4%. An *unacceptable call* is one that fails either of the above tests. In our evaluation of LA performance, we calculate \( L \) and \( E \) for the downlink (packets received by the MU), and assume that the quality is representative of the bidirectional call quality.

B. Estimating Capacity

Simulation provides a means for estimating system capacity. A set of simulations (of a particular system configuration) can be run, varying the number of users present, \( n_j, j = 1 \ldots J \), each for several simulation seeds \( \sigma_k, k = 1 \ldots K \). The result taken from a single simulation is the ratio of unacceptable calls to \( n_j \), denoted \( r_f(n_j, \sigma_k) \). The simulated failure rate is

\[
\hat{p}_f(n_j) = \frac{1}{K} \sum_{k=1}^{K} r_f(n_j, \sigma_k),
\]

so that a capacity estimate is

\[
\hat{C}(\epsilon) = \max\{n_j \mid \hat{p}_f(n_j) < \epsilon \}.
\]

In practice, we usually limit \( K \) to less than 10, due to the amount of time required for each simulation.

As an example, we simulated a system employing an ideal LA algorithm which chooses the highest rate possible based on the current channel conditions. It requires perfect knowledge of the SNR between the AP and the MU. We simulated the system using \( K = 13 \) seeds and over the range of user sizes \( n_j = 47 \ldots 58 \). The simulated failure rate \( \hat{p}_f(n_j) \) is shown in Fig. 2. If we choose \( \epsilon = 0.04 \), for example, then the estimated system capacity is 55 voice users.

V. IMPACT OF PARAMETER CHOICE

The ARF algorithm, as described in Section II, requires three parameters: the number of consecutive ACK messages needed to increase the rate \( \left(N_i\right) \), the number of consecutive missed ACKs that trigger a rate decrease \( \left(N_d\right) \), and \( N_p \), which replaces \( N_d \) as the rate-decrease threshold immediately following a rate increase. The shorthand notation \( \left(N_i, N_d, N_p\right) \) can be used to describe a system configuration. In this section, we discuss three considerations for selecting \( N_i \) and \( N_d \) for ARF, under the assumption \( N_p = 1 \).
1) Lower Limit for $N_i$: The first consideration is primarily applicable to a throughput-application, where there are a small number of users with a lot of traffic to transmit. In this environment, $N_i$ should not be too small. Consider a static environment where the channel is not changing and the MU has already reached its steady-state transmission rate. After $N_i$ consecutive transmissions, the MU will try to increase its transmission rate, which will be unsuccessful because the optimum transmission rate has already been reached. As a result, one packet loss will occur every $N_i + 1$ transmissions, leading to a minimum raw packet loss rate of $1/(N_i + 1)$. As an example, with $N_i = 2$, the raw packet loss rate will be 33% even in a lightly loaded network with good channel conditions. For a voice application, this criterion is less important than the following two, as will be explained later.

2) Net Bias Toward Rate Increase: In a voice application, there may be many low-traffic users trying to access the channel, each with an aggressive CWmin setting of 3 slots (Table I). This CWmin value could result in a collision probability as high as 43% in a fully-loaded network [8]. Consider a situation where the MU is using a transmission rate lower than its optimal rate; i.e., the transmission rate can still be increased and corruption by noise is insignificant. Let the collision probability be given by $P_{coll}$, a constant. The probability that the MU will increase the transmission rate equals the probability that it does not experience any collisions in $N_i$ transmissions, or $(1 - P_{coll})^{N_i}$. Additionally, the probability that the MU will unfavorably decrease the transmission rate equals the probability that it experiences $N_d$ consecutive collisions, or $(P_{coll})^{N_d}$. Because the MU is currently using a transmission rate lower than the optimal rate, we want a positive bias to increase the rate in this environment, or $(1 - P_{coll})^{N_i} > (P_{coll})^{N_d}$.

This bias is illustrated in Fig. 3, where we use an example $P_{coll}$ value of 43%. The z-axis represents the net bias, while the x-axis and y-axis represent different $N_i$ and $N_d$ values. A small $N_i$ is desirable: the net bias reaches its highest value when $N_i = 1$ and $N_d \geq 5$. Although $N_i = 1$ would lead to a 50% raw packet loss rate established in the first consideration, that loss rate is comparable to the 43% collision probability, reducing its relative impact on overall network performance.

3) Effect of Retransmission Limit: An isolated packet loss, according to the voice quality metric model, would result in an error event with 40% probability (Section IV-A), which will lead to a higher probability of an unacceptable call. The leading cause of packet loss is when the retransmission limit ($7$; see Table I) is reached. Consider a situation where the MU is using a transmission rate higher than its optimal rate, i.e., the transmission rate must be decreased. With $(1, 6, 1)$, it only has one remaining attempt to deliver the packet after the rate reduction because six failed transmissions are needed before the rate decrease. Hence, the probability that the retransmission limit will be reached in this scenario equals $P_{coll}^{(7 - N_d)}$, neglecting the corruption probability by noise. This packet drop probability equals 43%, 18%, and 8% with $(1, 6, 1)$, $(1, 5, 1)$, and $(1, 4, 1)$, respectively. The bias toward rate increase is nearly the same for $(1, 6, 1)$ and $(1, 5, 1)$, so choosing the smaller $N_d$ of 5 should reduce dropped packets. Subsequent simulations will provide additional insights into parameter selection.

VI. ARF PARAMETER SEARCH

Before conducting simulations of systems with varying ARF configurations, we consider some ARF configurations discussed in [6], and compare the results to the performance of the ideal algorithm.

A. Example ARF Configurations

The WaveLAN-II system was designed with ARF parameters $(10, 2, 1)$, although a rate increase is also triggered by a timer [5]. In [6], Van der Vegt considers $(5, 4, 0)$ and $(5, 4, 1)$, as well as $(11, 4, 1)$. A dynamic version of ARF is described in [7], where $N_d = 1$, but $N_i$ is either 3 or 10, based on the performance in a modified probation state. If the transmission following a rate increase is successful, $N_i$ is set to 3; if it fails, $N_i$ is set to 10, but the current rate is maintained.

A system capacity estimate was performed for parameters from [6] using $K = 5$ seeds. We chose $c = 0.04$. The $(11, 4, 1)$ configuration resulted in a 32-user capacity. The $(5, 4, 0)$ and $(5, 4, 1)$ cases have the same capacity, 33 voice users, but the packet loss rate ($L_p$) is slightly higher in the $(5, 4, 0)$ case for 34 users, so there is likely a benefit to using the probation state. We will use $N_p = 1$ in our search.
indicate that decisions typically impact only individual users (shallow curve for conditions, packet losses will occur.

A gap in capacity results between the ideal SNR LA algorithm and those listed above, it is possible that tuning the ARF algorithm by proper parameter selection may yield better results when voice is the primary application. A thorough search was performed using the simulation environment described in Section III.

**B. Simulation-based Parameter Search**

Knowing that 55 voice users are supported when an ideal LA algorithm is used, the search for an optimal ARF configuration begins by fixing the system size at 40 users. The simulated failure rate will be observed across multiple ARF configurations and simulation seeds. Certain configurations will be eliminated due to unacceptable failure rates, so we may simulate the system again with more voice users, but a smaller set of candidate configurations. This process will repeat until only one configuration remains.

Simulation results for \( N_i \leq 5 \) are depicted in Fig. 4. Each curve represents a different \( N_i \) value, while the horizontal axis represents each \( N_d \) tested. In this figure, each point represents averaged results over 4 simulations. The vertical axis provides the mean packet loss rate, which is the mean value of \( L \hat{p} \), per call, per simulation. Finally, circled points or dashed lines indicate that \( \hat{p}_f(40) > 4\% \).

Consider \( N_i = 1 \) and 2: if the value of \( N_d \) is too low, the link speed falls back too quickly, and average transmission speed is reduced. For low values of \( N_d \), the system capacity is less than 40 users, due to the lower average speed. In these cases, packet losses accumulate rapidly, as some users have difficulty accessing the channel. On the other hand, as \( N_d \) increases, the algorithm becomes slow to respond to changes in channel conditions. If the rate remains too high for conditions, packet losses will occur\(^2\). Poor link adaptation decisions typically impact only individual users (shallow curve with \( N_d \geq 7 \)), while the system exceeding capacity will affect all users (steep curve with \( N_d < 7 \)). (Note the increase in packet loss when \( N_d \) reaches the retransmission limit of 7; in these cases, the station may be unable to decrease the transmission rate until it has exceeded the retransmission limit, causing a packet loss.)

The \( N_i = 3 \) curve is quite different, as systems configured with \( N_d < 9 \) appear unable to support 40 users. Since the algorithm is more reluctant to increase the rate, only a small range of \( N_d \) values provides acceptable call quality; when \( N_d \geq 13 \), the performance is hindered by being slow to adapt to changes.

Curves for \( N_i = 4 \) and \( N_i = 5 \) are also shown in Fig. 4, where in every case, the simulated failure rate was greater than 4%. Simulations with \( N_i = 6 \) (not shown) followed a similar pattern. The pattern of increasing packet losses with increasing \( N_i \) indicates that the system is being stretched beyond capacity, so further simulations with \( N_i > 6 \) are not needed. Thus, we continue the search by considering those parameters that resulted in simulations with \( \hat{p}_f(n_j) < 4\% \): \( N_i = 1 \) with \( N_d = 3\ldots 6 \), \( N_i = 2 \) with \( N_d = 4\ldots 10 \), and \( N_i = 3 \) with \( N_d = 10\ldots 12 \).

Next, the system was simulated \((K = 7)\) with 43 voice users. The results showed degraded performance for \( N_i = 3 \), with \( \hat{p}_f(n_j) \) between 4% and 13% for the three cases tested. For \( N_i = 1 \) and \( N_i = 2 \), most parameters had low (or zero) failure rate, although in one of seven simulations of the \((2, 5, 1)\) configuration, all calls were unacceptable, so \((2, 5, 1)\) is eliminated. The configuration with the lowest average \( L \) was \((1, 3, 1)\) with about 2 lost packets per call.

Advancing the system size to 46 voice users, the parameter \((2, 8, 1)\) is the only parameter with \( N_i = 2 \) that maintained a low failure rate. For \( N_i = 1 \), the parameters with \( N_i = 4\ldots 6 \) supported acceptable calls. Increasing by one to a 47-user size, \((1, 6, 1)\) is eliminated. When the system size is again increased to 48 voice users, only configuration \((1, 5, 1)\) remains.

**C. System Capacity**

The best configuration of ARF for voice traffic appears to be \((1, 5, 1)\), with a system capacity of approximately 48 voice users; simulations with 49 voice users showed about a 17% failure rate for this configuration. To verify this result, some
additional simulations were run using the (1, 5, 1) configuration. Fig. 5 provides the simulated failure rate \( \hat{p}_f(n_j) \) for \( n_j = 44 \ldots 51 \); \( K = 10 \) seeds were used for each point. The additional simulations verify the capacity result.

If background traffic is present due to additional data users running an FTP application, for example, the system capacity will decrease slightly. Data users are modeled using a 100 kbps FTP service (with 75% of the bandwidth in the downlink direction), and 24 such users are added to the system in addition to the voice users.

All users again use ARF (1, 5, 1) for link adaptation, and several simulations were run, using \( K = 9 \) seeds and a range of user sizes, \( n_j = 42 \ldots 50 \). The simulated failure rate \( \hat{p}_f(n_j) \) is also shown in Fig. 5. The estimated capacity is 45 voice users (plus 24 data users), since \( \hat{p}_f(45) < 0.01 \) and \( \hat{p}_f(46) > 0.07 \).

D. Layer-2 Packet Loss Rate

We consider the layer-2 packet loss rate for the (1, 5, 1) configuration and compare it to the (5, 4, 1) and (11, 4, 1) configurations from [6]; this is the loss rate before packet retransmission (note that the \( L \) metric is the loss rate at the application layer). Each point in Fig. 6 represents \( K = 6 \) simulation seeds.

Note that for each configuration tested, the packet loss rate increases steadily with additional system load. When the systems operate below capacity (such as below 30 voice users), packet losses are influenced by the aggressiveness of the ARF configuration. The conservative (11, 4, 1) configuration results in loss rates of about 15% to 20%, while the (1, 5, 1) configuration causes packet loss to approach 35%.

E. Data-only Throughput

The (1, 5, 1) configuration has performed well for a system where voice is the primary application. Here, we consider a system with only data users running the FTP application, to determine if the (1, 5, 1) configuration provides good throughput. The system consists of 24 data users, each modeled using a 3.2 Mbps FTP service, with equal bandwidth in the uplink and downlink directions. The retransmission limit is 4, due to the longer packet sizes. Stations use RTS/CTS in order to transmit.

Simulations \( (K = 6) \) were run for three ARF configurations and the results are shown in Fig. 7. The figure depicts the moving average (3 s window) of traffic received at the TCP layer in MBytes per second. The (1, 5, 1) configuration provides good throughput compared to the settings from [6]. The scenario was also considered with RTS/CTS disabled, but the results were similar. We also tested the (2, 8, 1) configuration of ARF, but it performed similarly to the (1, 5, 1) configuration.

VII. CONCLUSION

When auto rate fallback LA is used as a method of increasing system capacity for voice users in an 802.11a WLAN system, parameter choice drastically affects performance. Parameters cited in [6] have a voice capacity of approximately 33 users, whereas the potential capacity (judged using an ideal LA algorithm) may be as high as 55 voice users; only 60% of the capacity is attained. By choosing the (1, 5, 1) configuration, a voice capacity of 48 voice users is possible, 87% of the available capacity.

REFERENCES