Rate-adaptive Framing for Interfered Wireless Networks [*]

Technical Report UIUCDCS-R-2006-2743

Chun-cheng Chen[†], Eunsoo Seo[†], Haiyun Luo[†], and Nitin H. Vaidya[‡]

[†]Dept. of Computer Science
Univ. of Illinois at Urbana-Champaign
Urbana, IL 61801-2302
{chen35,eseo2}@uiuc.edu
haiyun@cs.uiuc.edu

[‡]Dept. of Electrical and Computer Engineering
Univ. of Illinois at Urbana-Champaign
Urbana, IL 61801-2302
nhv@crhc.uiuc.edu

ABSTRACT

The majority of existing wireless rate controls are based on the implicit assumption that frames are corrupted due to the random, arbitrary environmental and thermal noises. They generally reduce the channel rate on frame losses, trading lower efficiency in frequency band utilization for more robust modulation so that the current noise level may be concealed. In highly interfered wireless networks where frames are lost mainly due to interference from other wireless transceivers, simply reducing the channel rate prolongs the frame transmission time and therefore aggravates frame loss ratio. This positive feedback in the rate control loop quickly diverges the interfered transceivers into a suboptimal channel rate and drives the network into a state with high interference. In the worst case, interfered transceivers can be starved. In this paper we present RAF, the rate-adaptive framing that jointly controls the channel rate and frame size according to the observed interference patterns and noise level at the receiver. Based on the inputs from physical layer carrier sense, the receiver derives the optimal channel rate and frame size that maximize throughput, and informs the transmitter of such optimal configuration in a few bits in the per-frame acknowledgement. Through intensive simulations we show that RAF consistently outperforms ARF (automatic rate fallback), the *de facto* rate control in 802.11 networks, in all our simulated scenarios. In particular, it increases the throughput of interfered transceivers by up to six folds and completely eliminates starvation in large random networks with various traffic matrix and interference levels.

1. INTRODUCTION

The proliferation of wireless devices on unlicensed frequency bands, e.g., 802.11, Bluetooth, and UWB, has changed the landscape of wireless networking. As the spatial and temporal intensity of such communications accelerate, wireless interference becomes one dominating factor to the success or failure of a transmission. However, the majority of existing wireless rate controls, e.g, ARF [21], AARF [25], ERF [20], link adaptation [28], ONOE [2], and SampleRate [9], are based on packet losses. The implicit assumption is that packet losses signal deteriorated link conditions, and consequently the channel rate should be reduced so that the physical layer switches to a more robust modulation scheme for better noise tolerance. However, at lower channel rate it takes longer to transmit a frame. In a highly interfered wireless network where packets are lost due to interference that comes and goes, depending on the activities of interfering transceivers, a lower channel rate may cause an even higher packet loss ratio due to the prolonged packet transmission time. The increased packet loss ratio in turn further decreases the channel rate. This positive feedback in the rate control loop may quickly drive every interfered transceiver into the lowest rate possible and the entire network into the highest interference level. In the worst case, heavily interfered transceivers can be starved.

Many recent measurement studies [12, 19, 29, 31] on 802.11 wireless networks have confirmed the above phenomenon. For example, it has been shown in [29] that in an 802.11b hotspot setting most of the transmission time is spent sending at 1 Mbps, the lowest rate. The network channel rate oscillates at high frequency and only one or two frames are sent between rate switches. This problem will become worse as the autonomous installations of 802.11 home wireless networks and hotspots quickly spread. Recent reports [3, 5] show that more than 40% of 802.11 home wireless routers are operating on the same channel 6 in metropolitan areas. What's more, a maximum number of 85 802.11 wireless routers were detected in the interference range in Boston [3, 7], among which at least 28 wireless routers must directly interfere with each other since 90% of them are 802.11b/g. With the current interference oblivious rate controls, communications in unlicensed frequency band will soon become the victims of their own success.

Several recently proposed SINR (signal-to-interference-noise-ratio) based rate controls can be applied to mitigate the above problem. For example, RBAR [16] leverages 802.11's per-frame handshake, i.e., RTS/CTS, to negotiate the best rate for the DATA message given the measured SINR of the RTS message. Although RBAR achieves fine time granularity rate control, it does not address the increase of the packet loss ratio because of the longer packet transmission time at lower channel rate. More importantly, RBAR mandates the per-frame RTS/CTS handshake, which accounts for a minimum 37% or 29% overhead[1] in throughput for

[*]The research is supported in part by NSF grant ANI-0125859.

[1]Although RTS/CTS frames are small, there is constant per-frame PHY and MAC overhead.

11Mbps 802.11b or 54Mbps 802.11a/g respectively. CARA [22] differentiates packet loss due to contention from packet loss due to channel errors using RTS probing, therefore sharing the same weaknesses as RBAR does.

In this paper, we present rate-adaptive framing (RAF), a novel joint channel rate and frame size control that optimizes the throughput for interfered transceivers. RAF seeks to characterize the observed interference pattern at the receiver, and uses the pattern to determine the optimal channel rate and the frame size for the maximum throughput. It is based on the assumption that wireless interference be predictable in short term, since usually the interfering wireless transmissions or background wireless traffic is not purely random even at fine time scale[2]. In more specific, for each channel rate a receiver divides a recent time window into a series of idle and busy intervals. Each idle interval is defined as a continuous time period during which the carrier sense value, as the sum of all interference and noise, is below certain threshold. See Figure 1 for an illustration. The threshold is derived by the required minimum SINR for successful packet delivery at the channel rate and the signal strength of the last received message, assuming that the signal strength from the specific transmitter does not change significantly from the last frame. Given the set of idle intervals for a channel rate, the receiver computes the optimal frame size that maximizes the achievable throughput. The receiver then compares the maximum achievable throughput at different channel rates, and finally communicates the optimal configuration of channel rate and frame size to the transmitter in a few bits in the per-frame acknowledgement. The transmitter then applies the channel rate and frame size in the next frame transmission.

RAF's adaptation for channel rate and frame size resides at the receiver, while the control decision is enforced at the transmitter. RAF works the best with a transmitter that is able to detect in real time the beginning of each idle interval and start transmission immediately. It turns out that in the literature SELECT [11], a transmitter side self-learning collision avoidance scheme, satisfies the need. The final communications system consists of RAF receivers and SELECT transmitters. It is the RAF receiver that determines the *channel rate* and *frame size*, and the SELECT transmitter that determines *when* to transmit. Note that in all our simulated scenarios RAF by itself still consistently outperforms ARF [21], albeit at a slightly lower performance gain.

In this paper we propose the first joint rate and frame size optimal control based on the observed patterns of the interference at the receiver. Through modeling, analysis, intensive simulations, and preliminary experiments, we present thorough comparison between our proposed RAF and the representative ARF to expose the potential gain. Our results show that RAF achieves up to about 600% improvement in throughput for interfered transceivers compared with ARF, the *de facto* rate control mechanism in existing 802.11 networks. Without intelligent sender side collision avoidance (SELECT), the throughput gain of RAF alone is around 10% lower on average but still significantly higher than the throughput of ARF. Our evaluation results also show that RAF adaptation converges very fast and adapts well to the system dynamics including node mobility and traffic variations.

The rest of this paper is organized as follows. We first compare with the related work and discuss related issues in Section 2. We then motivate joint channel rate and frame size control in Section 3, using 802.11a experiments with variable rates and frame sizes. We also model and analyze the performance of ARF in a typical hidden/exposed terminal scenario when compared with a fixed rate 802.11. In Section 4 we present the details of our design and control algorithms. Section 5 presents the performance evaluation of RAF using intensive ns-2 simulations. We finally conclude with future work in Section 6.

2. RELATED WORK

The majority of existing rate controls, e.g., ARF [21], ERF [20], AARF [25], link adaptation [28], ONOE [2], and SampleRate [9], are based on packet losses. In highly interfered wireless networks, in particular those defined in the unlicensed frequency bands, reducing the channel rate on packet losses increases the contention for the shared wireless channel. It therefore aggravates interference and further increases the packet loss ratio. RAF is designed to address both interference and noise, through adaptation based on receiver side carrier sense.

In contrast to those packet-loss based rate controls, RBAR [16] adapts the channel rate based on SINR. An RBAR receiver determines the highest data rate supported by the SINR of the RTS message, and informs the sender of the rate with the CTS message. CARA [22] uses RTS probing to differentiate packet loss due to contention from packet loss due to channel error. RAF is different in that it avoids the RTS/CTS overhead (37% and 29% overhead for 11Mbps 802.11b and 54Mbps 802.11g as shown in [10]). HRC [15] uses SINR to fine-tune loss-based rate controller. RAF is different in the way SINR is used for rate control. Link adaptation based on received signal strength [13] is relevant to our design in that its rate adaptation is based on the received signal strength at the receiver. Different from all above rate adaptations, RAF further incorporates frame size control for optimizing throughput at highly interfered or even starved transceivers.

Note that RBAR, CARA, HRC, and RAF's rate controls are all based on SINR, while recent measurements [6, 9] on existing 802.11 mesh suggest that SINR be not a good predictive tool for the successful delivery of a packet. At the first glance, our approach seems to be contradictory to the measurements in [6]. The fact is that the SINR interface exposed by existing 802.11 firmware, as a readable register of the microcontroller, only outputs the *average SINR over many received packets*[3]. Indeed as shown in [6, 9], *average* SINR may not correlate well with the success or failure of *individual* packet delivery. In contrast, RAF is based on the carrier sense of the total interference and noise level at the time granularity finer than one packet transmission time. It does put higher requirement on the physical layer *carrier sense* module, of which the performance and impact have

[2] See the literature of wireless traffic measurement, modeling and analysis, e.g., [30, 24, 26].

[3] We have confirmed this fact with the authors of [6].

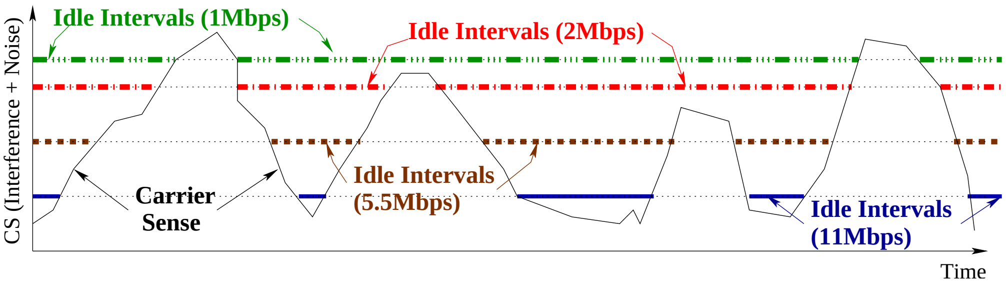


Figure 1: RAF receiver maintains multiple series of "idle" intervals. During each series the total interference and noise are less than certain threshold, which guarantees successful transmission and receiving at certain rate.

been recently evaluated [18].

To leverage the full benefits of RAF's frame size control, aggregation of small packets from upper layers is necessary, in addition to fragmentation of large packets. Large packet fragmentation is already part of the 802.11 standard. Small packet aggregation in 802.11 was studied in [23] and relevant to the way RAF enforces its frame size control.

Our modeling and analysis of ARF are based on the classic analysis of single non-interfered 802.11 WLAN [8] by Bianchi and the modeling of single rate two flow topology [14] by Garetto et al. We further extend their models to the scenarios with multiple data rates to compare the expected throughput achievable through ARF and that through fixed rate 802.11 in a typical hidden/exposed terminal scenario.

There are a number of proposals on transmission power control for optimizing network capacity in the literature. For example, Akella etc. [7] propose PARF and PERF that extend ARF [21] and ERF [20] respectively for conservative power control. We do not control the transmission power in the current design of RAF. Incorporating power control into RAF's rate and frame size control framework will be part of our future work.

3. MOTIVATION

Before we describe the details of RAF design, we present the experimental results obtained in an 802.11a testbed where a receiver is interfered by a hidden transmitter. We compare the throughput of ARF and the throughput of various fixed channel rates at different fixed frame sizes. We then model the throughput of ARF in 802.11b networks[4]. Based on the model, we derive under different levels of interference the probabilities that there exists a fixed 802.11b channel rate that out-performs ARF. Our analysis exposes the potentials of our proposed RAF throughput optimization.

3.1 802.11a Experiments

We place four Soekris [4] small-factor communication computers ∼15 meters apart from each other in a linear topology, as shown in Figure 2. Each Soekris box is equipped with an Atheros 802.11a interface running the MADWiFi [1] driver. All 802.11a interfaces transmit at 0dBm through an omini-directional 2dBi antenna. The default channel rate control distributed in MADWiFi is SampleRate [9], which implements the original ARF [21] with many improvements. It is still based on packet losses but differs in the way different rates are sampled and the way the packet loss rates are measured. All the results presented in this section are the averages over 8 experiments conducted in an indoor lab environment.

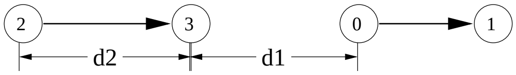


Figure 2: Interference from hidden transmitter

We first calibrate the interference at the two receivers shown in Figure 2. We found out that receiver 1 is not interfered by transmitter 2 for all the 4 link rates we tested, i.e., 6, 9, 12, 18Mpbs. We therefore run ARF at transmitter 0, with variable offered loads of CBR/UDP traffic. We change the offered loads from 2 to 12Mbps, with 2Mbps increment. For each offered load we apply four different packet sizes: 500, 800, 1100 and 1400 bytes. The variable offered loads between 0 and 1 serve as different levels of interference at receiver 3. For each interference level, we apply ARF and different fixed link rates (6-18Mbps), together with variable frame sizes between node 2 and 3. We keep node 2 always backlogged so it transmits as fast as the 802.11 link allows. We found out that 20 out of the total number of 24 different configurations of offered load and frame size at the interfering node 0, there exists a fixed link rate and frame size configuration that outperforms ARF in throughput at node 2 and 3, with the gain varying from 7% to 95%. This results clearly show the potential of intelligent channel rate and frame size control.

3.2 ARF Modeling and Analysis

We take three steps to quantify the room for throughput optimization on top of ARF in a typical hidden/exposed terminal scenario as shown in Figure 2. We first derive the throughput of an interfered receiver at different channel rates and interference levels. We then derive the expected throughput of ARF based on its state machine. We finally plug into our models realistic parameters based on published data, and show the probability that there exists a fixed 802.11 channel rate and frame size configuration outperforming ARF.

[4] We present the analysis results for 802.11b for simplicity. The analysis easily extends to 802.11a/g where more channel rates are available.

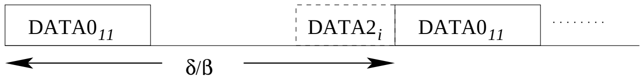


Figure 3: Channel states perceived by node 3 when node 2 transmits packets at rate iMbps

We consider the topology in Figure 2, the classic example of hidden/exposed terminal to illustrate the impact of wireless interference. Since the receiver 3 is interfered by the transmitter 0 while the receiver 1 is farther away from the transmitter 2, we focus our attention on the achievable throughput between node 2 and 3 with and without ARF. We first extend the model developed in [8, 14] to the cases of multiple rates.

The fixed rate 802.11 throughput between node 2 and 3 is modeled in [8, 14] as:

$$Thr_i = \frac{\tau_i(1-p_i)}{\tau_i(1-p_i)\overline{T}_s(i) + \tau_i p_i \overline{T}_c(i) + (1-\tau_i)\sigma_i} \qquad (1)$$

where $i \in \{1, 2, 5.5, 11\}$, representing the four 802.11b channel rates. $\overline{T}_s(i)$ and $\overline{T}_c(i)$ are the average time intervals that a packet transmission succeeds and collides respectively, at channel rate i. σ_i represents the average time interval that node 2 is idle. p_i is the packet loss probability conditioned on the fact that node 2 does send out a packet. τ_i is the probability that node 2 sends a packet after the random backoff at channel rate i, assuming node 2 is continuously backlogged. It can be expressed as a function of p_i as:

$$\tau_i = \frac{2q_i(1-p_i^{m+1})}{q_i(1-p_i^{m+1}) + W_0[1-p_i-p_i(2p_i)^{m'}(1+p_i^{m-m'}q_i)]} \qquad (2)$$

where $q_i = 1-2p_i$, W_0 is minimum 802.11 contention window size, m is maximum retry limit, and m' is the backoff stage when it reaches the maximum window size. Given that the way $\overline{T}_s$ and $\overline{T}_c$ are determined in [8] still applies here, we only need to derive the conditional frame loss probability p_i to complete the throughput model for the interfered receiver 3.

To relate the achievable throughput to the distances d_1 and d_2, we apply the ideal joint free-space and two-ray ground signal propagation model [27]:

$$Pr(d) = \begin{cases} \frac{P_t \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 \cdot d^2 \cdot L} & \text{if } d \le d_{cross} \text{ (free-space)} \\ \frac{P_t \cdot G_t \cdot G_r \cdot h_t^2 \cdot h_r^2}{d^4 \cdot L} & \text{if } d > d_{cross} \text{ (two-ray ground)} \end{cases}$$

where P_r is the received signal strength, P_t is the transmission power, and d is the distance between the transmitter and the receiver. G_t and G_r are the antenna gains at the transmitter and the receiver respectively, h_t and h_r are the heights of the antennas, $L\,(L \ge 1)$ is the system loss factor, λ is the wave length in meters, and d_{cross} is the crossover distance defined as $d_{cross} = (4\pi\, h_t\, h_r)/\lambda$. For simplicity we further assume an unrealistic binary channel model where SINR threshold for successful packet delivery is set to α_i for rate i. Due to the lack of space we omit the analysis based on a more realistic SINR-BER (bit error rate) curve in this paper, since it leads to similar conclusion.

Assuming a CBR-alike traffic between the interfering transmitter 0 and receiver 1, with offered load β and fixed frame size δ. The channel status observed at the interfered receiver 3 is shown in Figure 3. Note that since receiver 1 is not interfered, the data rate at node 0 is always the highest 11Mbps under ARF. At the interfered receiver 3, the delivery of a packet only succeeds if the transmission completely falls into the interval between two consecutive transmissions at the interfering transmitter 0. Assuming node 2 initiates the transmission uniformly randomly during the period δ/β, the packet loss ratio p_i between node 2 and 3 can be characterized as:

$$p_i = \begin{cases} \frac{DATA2_i + DATA0_{11}}{\frac{\delta}{\beta}}, \\ \quad \text{if } Pr(d_2) \leqslant Pr(d_1)\,\alpha_i, \frac{\delta}{\beta} \geqslant \eta, DATA2_i + DATA0_{11} < \frac{\delta}{\beta} \\ 1, \\ \quad \text{if } Pr(d_2) \leqslant Pr(d_1)\,\alpha_i, \frac{\delta}{\beta} \geqslant \eta, DATA2_i + DATA0_{11} \geqslant \frac{\delta}{\beta} \\ 1 - \frac{2}{W_0[2T_s + (W_0-1)\sigma]\sum_{i=0}^{W_0-1} max(0, D+i\sigma)}, \\ \quad \text{if } Pr(d_2) \leqslant Pr(d_1)\,\alpha_i, \frac{\delta}{\beta} < \eta \\ 0, \\ \quad \text{if } Pr(d_2) > Pr(d_1)\,\alpha_i \end{cases}$$

where $DATA2_i$ is the data transmission duration at rate iMbps for node 2, similarly $DATA0_{11}$ is the data transmission duration at rate 11Mbps for node 0. $Pr(d_2)$ is the signal strength at node 3 when node 2 is transmitting. $Pr(d_1)$ is the signal strength at node 3 when node 0 is transmitting. $T_s = DATA0_{11} + SIFS + ACK + DIFS$, $\eta = DATA0_{11} + SIFS + ACK + DIFS + CWmin \cdot \sigma$, and $D = ACK + DIFS - DATA2_i + SIFS$. The first case represents the scenario where the offered load at node 0 is light, and frame size at node 2 is small enough to fit into the idle intervals. The second case represents the scenario where node 2's frame size becomes too large to fit into any idle interval and all channel access attempts will fail. When the offered load at node 0 is heavy we can derive p_i based on the results from [14]. It is shown as the third case and we again omit the details for concise representation. The last case represents the scenario where the SINR is always above the threshold (because node 2 and 3 are very close, node 3 and 0 are farther away, or both) and receiver 3 is not interfered at all. With the packet loss ratio p_i finally derived we can obtain the throughput between node 2 and 3 at different channel rates (i) and interference levels (d_1, d_2, P_r, δ/β, α_i).

Next we model the throughput of ARF. ARF reduces the channel rate to the next lower level when there are two consecutive packet losses, and increases the channel rate to the next higher level when there are ten consecutive successful packet deliveries. It turns out that the ARF state machine easily translates into a 34-state Markov model (we omit the details for concise presentation). Given the throughput derived above, the expected ARF throughput follows under different interference levels, as a function of the distances (d_1 and d_2), offered load at the interfering nodes (δ and β), and the SINR threshold (α_i). We have verified the multiple rate and interference level model through intensive simulations in ns-2. Figure 2 shows one example topological setting where transmitter 2 is always backlogged and the offered load at the interfering node 0 is 2.2Mbps with 1300B frame size. For fixed rate 802.11b, we plot the highest throughput at rate 11Mbps. We can see from the figure that the model matches the simulations well, and indeed one fixed-rate 802.11b sig-

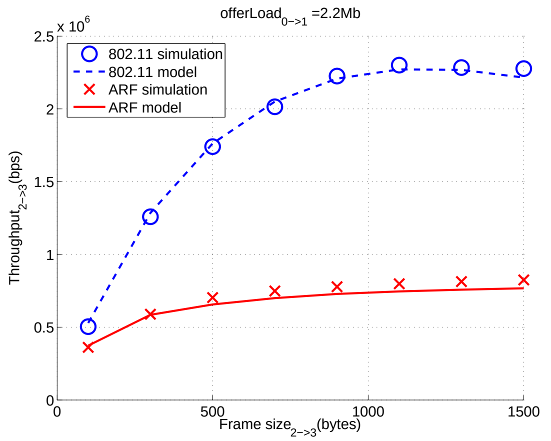


Figure 4: ARF and fixed rate 802.11b throughput

SINR	Probability
$(0, \alpha_1]$	100%
$(\alpha_1, \alpha_2]$	36%
$(\alpha_2, \alpha_{5.5}]$	28%
$(\alpha_{5.5}, \alpha_{11}]$	11%
(α_{11}, ∞)	0%

Table 1: 802.11 outperforming ARF probability

nificantly outperforms ARF.

We finally systematically quantify the potentials for throughput improvement over ARF in terms of the probability that there exists a fixed rate and frame size leading to a higher throughput than ARF. By varying d_1 and d_2, SINR at node 3 ($\frac{Pr(d2)}{Pr(d1)}$) can be categorized into five ranges, i.e., (α_{11}, ∞), $(\alpha_{5.5}, \alpha_{11}]$, $(\alpha_2, \alpha_{5.5}]$, $(\alpha_1, \alpha_2]$, $(0, \alpha_1]$, corresponding to required SINRs for the four 802.11b channel rates plus zero. Assuming frame sizes are uniformly distributed in $(0, 1500]$ bytes and offerLoad$_{0\rightarrow1}$ uniformly distributed in $(0, 6.4]$ Mbps (6.4Mbps is the saturation throughput when frame size is 1500B), we can calculate the probability that ARF can be outperformed for SINRs in each of the five ranges.

We set the SINR threshold α_i according to the published data in [7]. The final results are summarized in Table 1. For example, when the SINR at node 3 is in the range of $(\alpha_1, \alpha_2]$, 802.11b could have outperformed ARF with a fixed one of the higher channel rates (2Mbps, 5.5Mbps, and 11Mbps) in 36% of scenarios of various offered interfering loads and frame sizes. When the SINR at node 3 falls in $(\alpha_2, \alpha_{5.5}]$, 802.11b outperforms ARF in 28% of the scenarios if the channel rate is fixed at 5.5Mbps or 11Mbps. These figures show that the improvement we project above is not just for some specific niche scenarios. It is actually highly probable given a random network and traffic configuration.

4. RATE-ADAPTIVE FRAMING

The design of RAF resides at the datalink layer. It first maintains a recent history of fine-grained carrier sense from the physical layer. RAF then computes the optimal channel rate and frame size. Finally RAF communicates the optimal configuration to potential transmitters, where the optimal channel rate and frame size are enforced. We present the design details of the above procedure in the rest of this section.

4.1 Fine-grained carrier sense

RAF is based on the assumption that wireless interference be predictable in short term, since usually the interfering wireless transmissions or background wireless traffic is not purely random even at fine time scale [30, 24, 26]. RAF first maintains the history of the physical carrier sense in the forms of multiple series of idle and busy intervals. Each idle interval is defined as a continuous time period during which the carrier sense, as the sum of all interference and noise, is below certain threshold. A time period is busy if it is not idle. See Figure 1 for an illustration.

Note that in existing 802.11 radio a single carrier sense threshold is hardwired, and the physical layer signals whenever the carrier sense reading moves across the threshold[5]. See [18] for a brief summary of how carrier sense is implemented. RAF follows the carrier sense design logic, but requires that the physical layer be able to accept multiple configurable carrier sense thresholds (e.g., the four horizontal lines in Figure 1), each corresponding to the maximum interference and noise that are tolerable for communications at certain channel rate. Furthermore, the physical layer must signal in real time whenever the carrier sense reading moves across any of the thresholds.

With the input from the physical layer carrier sense RAF maintains each series of idle/busy intervals in a simple FIFO circular buffer (implemented as an array and a pointer pointing to the end of the series). Each buffer item simply records the length of an idle/busy interval. Note that RAF does not maintain the detailed carrier sense reading. RAF also implicitly controls the length of the maintained carrier sense history by simply bounding the size of the circular buffer, eliminating the need to maintain time-stamps. This simple control method automatically adapts to the dynamics of the channel status, since old records are overwritten faster when the channel status is volatile.

Note that the carrier sense thresholds that RAF submits to the physical layer may be different for different transmitters, depending on their signal strength. Even for the same transmitter, the thresholds may change over time, because its signal strength changes over time due to the dynamics in node mobility and channel fading. In RAF we use the transmitter's most recent signal strength as the reference, and calculate the carrier sense thresholds as the quotients of the transmitter's most recent signal strength and the SINR thresholds. These carrier sense thresholds are then passed to the physical layer.

4.2 Optimal channel rate and frame size

Given the idle intervals $idle_k$, the receiver calculates the optimal channel rate and packet size by maximizing the fol-

[5] With necessary hysteresis to avoid thrashing.

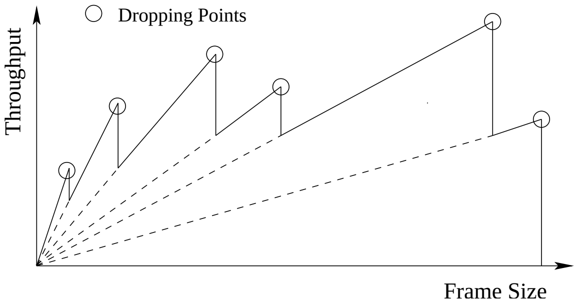


Figure 5: $f(s)$ versus frame sizes

lowing throughput function $f(s)$:

$$\max_{i\in R, s\in S} f(s) \tag{3}$$

$$f(s) = s\sum_{k=1}^{n_i}\left\lfloor\frac{\max(idle_k - \text{InitBackoff}, 0)}{s/i + OH + C}\right\rfloor. \tag{4}$$

where R is the set of channel rates, S is the set of frame sizes, n_i is the number of idle intervals for rate i maintained at the receiver, OH is the per-frame PHY/MAC overhead, and C is the inter frame overhead ($\text{CW}_{min}/2\cdot$ aSlotTime + SIFS + ACK + DIFS). To determine the initial backoff InitBackoff at the beginning of each idle interval, we assume the sender is able to detect when the channel becomes idle and resumes its backoff timer immediately. That is, the frame delivery fails only near the end of an idle interval, since the sender cannot predict when the idle interval ends before it starts transmitting the last frame. It turns out that in the literature SELECT [11], a transmitter side self-learning collision avoidance scheme, satisfies the need. Assuming such a transmitter the expected initial backoff is $\text{InitBackoff} = \text{CW}_{min}\cdot\text{aSlotTime}$, since the contention window size always doubles to 2CW_{min} after the last frame is lost at the end of the previous idle interval. Our analysis below is based on this initial backoff setting. If RAF is not working with a SELECT transmitter, the expected initial backoff could be larger, up to $\text{CW}_{max}/2\cdot\text{aSlotTime}$ in the worst case. We find that the inaccuracy in initial backoff slightly decreases the throughput gain by around 10% in our simulations (Section 5).

One naive search for the optimal throughput is to enumerate all possible packet sizes, e.g., from 1 to 1500 bytes, resulting a running time of $O(|R|\,|S|\,n_{max})$, where $n_{max} = \max_i n_i$, for $i \in R$. In the rest of this subsection, we describe two methods to reduce the computation overhead.

Our first method is based on the observation that $f(s)$ be a saw-tooth shaped function, with each segment a linear function extending to the origin. See Figure 5 for an illustration. Furthermore, the slope of a linear function, i.e., the sum of the floor functions in Eqn. 4, monotonically decreases as the frame size increases. Since the slopes are all positive, the maximum throughput within a segment must appear at the right end, followed by a sudden drop to the next linear segment with a lower slope. We therefore can safely skip all intermediate frame sizes in our search for the one leading to the maximum throughput. In specific, we start from MIN_PKT_SIZE, calculate the corresponding throughput $f(s)$ and the next dropping point $s_{nextDrop}$, proceed to the next packet size which is equal to either the current packet size plus default increment or the $s_{nextDrop}$, whichever is larger.

To calculate $s_{nextDrop}$ given s, we take advantage of the fact that when $f(s)$ increases linearly with frame size s within a segment, the dropping point must occur when frame size s is just enough to fit one or multiple frame transmission time ($\frac{s}{i} + OH + C$) into some idle interval $idle_{nextDrop}$, with no residual left. In fact, $idle_{nextDrop}$ is the idle interval with minimum normalized residual, the residual normalized by the number of frames already fit in the interval. Let the normalized residual for idle interval $idle_k$ and frame size s be $\xi(idle_k, s)$, $idle_{nextDrop}$ can be expressed as follows:

$$\begin{aligned} idle_{nextDrop} &= \text{argmin}_{idle_k\in H_i}\xi(idle_k, s) \qquad (5)\\ \xi(idle_k, s) &= \frac{D - \lfloor D\rfloor}{\lfloor D\rfloor} \qquad (6)\\ \text{where }\ D &= \frac{\max(idle_k - \text{CW}_{min}\ \cdot\ \text{aSlotTime}, 0)}{\frac{s}{i} + OH + C} \end{aligned}$$

Note that the $idle_{nextDrop}$ is readily available when computing $f(s)$. After identifying $idle_{nextDrop}$ the frame size at the next dropping point $s_{nextDrop}$ can be calculated by:

$$s_{nextDrop} = \left(\frac{adjIdle_{nextDrop}}{E} - OH - C\right)\cdot i \tag{7}$$

$$adjIdle_{nextDrop} = \max(idle_{nextDrop} - \text{CW}_{min}\ \cdot\ \text{aSlotTime}, 0)$$

$$E = \begin{cases} \left\lfloor\dfrac{adjIdle_{nextDrop}}{\frac{s}{i}+OH+C}\right\rfloor, \text{ if } \left\lfloor\frac{adjIdle_{nextDrop}}{\frac{s}{i}+OH+C}\right\rfloor \neq \frac{adjIdle_{nextDrop}}{\frac{s}{i}+OH+C} \\ \dfrac{adjIdle_{nextDrop}}{\frac{s}{i}+OH+C} - 1, \text{ if } \left\lfloor\frac{adjIdle_{nextDrop}}{\frac{s}{i}+OH+C}\right\rfloor = \frac{adjIdle_{nextDrop}}{\frac{s}{i}+OH+C} \end{cases}$$

The pseudo-codes are shown in line 8-12 and line 24-32 in Figure 6.

Our second method is based on the observation in our simulations that the number of idle intervals within certain time window, i.e., the time window (MAX_WINDOW_SIZE) that bounds the history of idle/busy intervals, could be very small, especially for lower channel rates. It turns out that when there is only a single idle interval, we can find the optimal frame size in constat time without going through all the dropping points. We start from the following proposition:

Proposition 1. *With one idle interval $f(s)$ at the dropping points are strictly increasing.*

Proof. Let m and $m-1$, $m > 0$, be the slops for the two consecutive dropping points. Let s_1 and s_2 be the corresponding dropping-point frame sizes, $s_1 < s_2$. With some algebraic manipulation, it is easy to see that $s_1 = \frac{(idle - m(OH+C))\ i}{m}$ and $s_2 = \frac{(idle-(m-1)(OH+C))\ i}{m-1}$. Also, we have:

$$\begin{aligned} (m-1)\ (OH+C)\ i &< m\ (OH+C)\ i\\ \Rightarrow\ (idle - m(OH+C))\ i &< (idle-(m-1)(OH+C))\ i\\ \Rightarrow\ m\frac{(idle - m(OH+C))\ i}{m} &< (m-1)\frac{(idle-(m-1)(OH+C))\ i}{m-1}\\ \Rightarrow\ m\ s_1 &< (m-1)\ s_2\\ \Rightarrow\ f(s_1) &< f(s_2) \end{aligned}$$

```
getRateAndPktSize($H_1, \ldots, H_{|R|}$)
$H_i$, $i \in R$: the idle busy history
$R$: the set of available channel rates
$S$: the set of available pkt sizes
 1: $windowSize \leftarrow$ getMinLength($H_1, \ldots, H_{|R|}$)
 2: if $windowSize >$ MAX_WINDOW_SIZE then
 3:    $windowSize \leftarrow$ MAX_WINDOW_SIZE
 4: $maxThr \leftarrow -\infty$
 5: for all $i \in R$ do
 6:    cut $H_i$ to length $windowSize$
 7:    if $H_i$ has more than one idle interval then
 8:       for $s \leftarrow$ MIN_PKT_SIZE, $s \leq$ MAX_PKT_SIZE do
 9:          $(thr, s_{nextDrop}) =$ getThr($s, H_i$)
10:          if $maxThr < thr$ then
11:             $maxThr \leftarrow thr$   $bestRate \leftarrow i$   $bestPktSz \leftarrow s$
12:          $s \leftarrow max(s_{nextDrop}, s + default\_increment)$
13:    else
14:       $s_{lastDrop}$ ←pktLastDrop($idle$, MAX_PKT_SIZE , $i$){Eq.
          8}
15:       $thr1$ ←calcThr($idle, s_{lastDrop}, i$){from Eq. 4}
16:       $thr2$ ←calcThr($idle$, MAX_PKT_SIZE, $i$){from Eq. 4}
17:       if $maxThr < max(thr1, thr2)$ then
18:          $maxThr \leftarrow max(thr1, thr2)$   $bestRate \leftarrow i$
19:          if $thr1 > thr2$ then
20:             $bestPktSz \leftarrow s_{lastDrop}$
21:          else
22:             $bestPktSz \leftarrow$ MAX_PKT_SIZE
23: return $bestRate$, $bestPktSz$
getThr($s, H_i$)
$s$: pkt size
$H_i$: the idle busy history for rate $i$
24: $minResidual \leftarrow \infty$
25: $thr \leftarrow 0$
26: for all $idle_k \in H_i$ do
27:    $thr \leftarrow thr +$ calcThr($idle_k, s, i$) {from the floor func. term in
       Eq. 4}
28:    $normResidual$ ←calcNormResidual($idle_k, s, i$) {from Eq.
       6}
29:    if $normResidual > minResidual$ then
30:       $minResidual \leftarrow normResidual$   $dropIndex \leftarrow k$
31: $s_{nextDrop} \leftarrow$ pktNextDrop($idle_{dropIndex}, s, i$) {from Eq. 7}
32: return $thr, s_{nextDrop}$
```

Figure 6: Optimal rate and pkt size calculation

□

Given the above proposition, we can start the search for optimal frame size from MAX_PKT_SIZE and move backward for the last dropping point $s_{lastDrop}$ for the optimal frame size.

$$s_{lastDrop} = \left(\frac{adjIdle}{F} - OH - C \right) \cdot i \tag{8}$$

$$adjIdle = \max(idle - \mathrm{CW}_{min} \cdot \mathrm{aSlotTime}, 0)$$

$$F = \begin{cases} \left\lceil \dfrac{adjIdle}{\frac{s_{max}}{i} + OH + C} \right\rceil, \text{ if } \left\lceil \frac{adjIdle}{\frac{s_{max}}{i} + OH + C} \right\rceil \neq \frac{adjIdle}{\frac{s_{max}}{i} + OH + C} \\ \dfrac{adjIdle}{\frac{s_{max}}{i} + OH + C} + 1, \text{ if } \left\lceil \frac{adjIdle}{\frac{s_{max}}{i} + OH + C} \right\rceil = \frac{adjIdle}{\frac{s_{max}}{i} + OH + C} \end{cases}$$

Line 14 to 22 in Figure 6 show the pseudo-codes for the above computation. In our simulated scenarios the two methods reduce the computation overhead by at least 75%.

4.3 Frame aggregation and fragmentation

To achieve the maximum throughput, an RAF transmitter may aggregate small packets or fragment large packets from upper layer to fit into the designated optimal frame size. Note that fragmentation is already part of the 802.11 standard and applies here. To enable aggregation, an RAF transmitter precedes every aggregated upper layer packet in the frame payload with two bytes. These two bytes specify the length of the following aggregated packet, as shown in Figure 7. When an RAF receiver receives a frame, it reads the first two bytes of the frame payload and extracts the first packet. If the read pointer does not hit the end of the frame yet the RAF receiver interprets the next two bytes as the length of the next aggregated packet. It then extracts the next packet accordingly. This process continues until all aggregated packets in the frame are extracted.

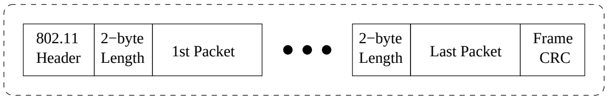


Figure 7: RAF frame format

Note that the above aggregation works fine even if the last fragment of the previous packet has to be combined in a single frame with the following packets. The 802.11 header for the frame will contain the necessary fragmentation information for the fragment, located at the beginning of the frame, to be de-fragmented at the receiver. Other packets in the same frame payload can be extracted following the same procedure as described above. To avoid further complexity we do not include more than one fragment in a single frame[6], since there is only one 802.11 header with room for the fragmentation information of one fragment only.

4.4 Optimal configuration update

An RAF receiver piggybacks the optimal channel rate and frame size in the per-frame 802.11 ACK message, and an RAF transmitter applies the updated configuration to the transmission of the next frame. We start by introducing the 802.11 ACK frame. Figure 8(a) shows the format of an 802.11 ACK message. The Duration field is set to zero unless the More Fragments bit in the Frame Control is set to 1, in which case the Duration field contains the remaining time in microseconds to finish transmitting the entire fragmented packet.

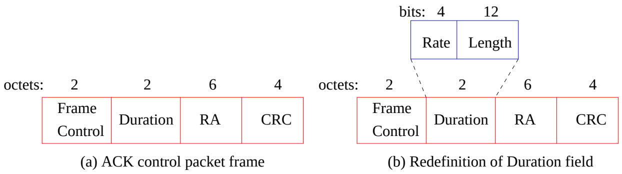


Figure 8: Original and redefined 802.11 ACK format

RAF redefines the 16-bit Duration field to carry the updated channel rate and frame size information. It divides the 16-bit Duration field into two subfields, one 4-bit Channel Rate subfield and the other 12-bit Frame Length subfield, as shown in Figure 8(b). All the other nodes overhearing the ACK frame will disregard the information in the duration field when More Fragments bit in the Frame Control is set

[6] In fact, there could be at most two fragments in the one frame, and they must appear at the beginning and the end of the frame payload respectively.

to 0. When the bit is set to 1, the duration of the next fragment transmission can be easily calculated given the channel rate and frame size encoded in the frame header.

5. PERFORMANCE EVALUATION

We implement RAF in *ns-2* simulator version *2.29*. For comparison we also implement ARF as described in Section 3.2. The 802.11 physical layer in ns-2.29 is overly simplified. Nodes receive packets only when the signal from the sender is greater than the receive threshold. However, the impact of any signal with strength less than the carrier sense threshold is completely ignored—no matter how many those signals are. We replace this part of 802.11 functions with the ones developed in [17], so that all signals are taken into account at receiver, and the combined SINR is used to determine if an incoming signal can interfere or be received/captured. We adopted the capture threshold as listed in [7] so that the SINR has to be greater than 3, 4, 8, 12dB in order for receiver to respectively receive the frame at 1, 2, 5.5, and 11Mbps or above. We use Two-Ray Ground radio propagation model, and the transmitting power is set so that the communication range is 115m, and the carrier sensing range is set to 200m. We use 2Mbps basic rate and 11Mbps channel rate based on IEEE 802.11b. Each simulation runs for 45 seconds unless otherwise specified.

5.1 Simple topology

We first evaluate the throughput of the simple hidden/exposed terminal topology shown in Figure 2. In this case, sender 0 and 2 are outside the interference range of each other. Client 3 is an exposed receiver since it is placed in the interference range of client 0, which is associated with another access point (node 1) in a neighboring BSS. Notice that in this configuration flow 0→1 will always succeed in the channel contention because its receiver (client 1) is not interfered by flow 2→3. We therefore vary the offered load (CBR/UDP rate) of flow 0→1 serving as variable interference level, while keeping interfered sender 2 always backlogged (with a 11Mbps CBR). We compare the throughput of ARF and RAF.

5.1.1 Variable interference levels

When the distance (d_1) between node 3 and 0 is equal to 120 m, flow 0→1's transmission will always interfere flow 2→3 even if the lowest 1Mbps channel rate is applied at node 2. Therefore, ARF throughput is close to zero no matter what frame size is applied, as shown in Figure 9. In contrast, RAF achieves significantly higher throughput than ARF. This is because when interfered, ARF's convergence to lower channel rate prolongs the transmission time, which in turn further increases the chances of interference. In the worst case, ARF may stuck at the lowest rate due to aggravated interference - leading to the starvation of flow 2→3. RAF, on the other hand, stays at higher rates with much better channel bandwidth sharing. We also plot the throughput of RAF with a SELECT transmitter and with a regular 802.11 transmitter, both enforcing the RAF receiver's channel rate and frame size control. As shown in Figure 9 again, the throughput of RAF with a regular 802.11 transmitter decreases only slightly and remains consistently higher than the throughput of ARF.

We then increase the distance between the two flows to reduce the level of interference at node 3. Figure 10-12 again show the throughput of flow 2→3 with varied offer loads at flow 0→1 as the distance (d_1) between node 3 and 0 increases from 135m, 160m, to 180m. Under these three distances the maximum channel rate at node 3 that is not interfered becomes 1Mbps, 2Mbps, and 5.5Mbps respectively. Again RAF consistently outperforms ARF at all different $\text{offerLoad}_{0\to1}$. Note that although the throughput of ARF is improved when larger frame size is used (due to reduced MAC/PHY overhead), RAF still outperforms ARF. This is because ARF keeps trying higher channel rates for every ten continuous successful packet deliveries, while RAF stabilizes at the optimal channel rate and frame size. We also observe that when the $\text{offerLoad}_{0\to1}$ is low, RAF improves the throughput by transmitting at the highest channel rate and large frame size, despite the existence of interference. When $\text{offerLoad}_{0\to1}$ increases, RAF reduces the frame size but stays at the highest channel rate. When the $\text{offerLoad}_{0\to1}$ increases up to certain point, RAF uses large frame size and switches to the highest possible channel rate that will not be interfered. In specific, Figure 14 shows the maximum, minimum, median, upper quartile, lower quartile of the optimal calculated frame size at node 3. When the offer load of flow 0→1 is less than 3Mbps, flow 2→3 uses 11Mbps rate and large frame sizes. When the offer load increases channel rate keeps 11Mbps but with smaller frame sizes. When offer load is larger than 3.5Mbps, node 3 decides to chose rate 1Mbps and frame size as large as possible. Note that there are several data points identified as outliers by the default box plot function. These data points are the calculated at the start of the simulation due to lack of information. We summarize the throughput improvement when d_1 is 135 m, 160 m, and 180 m for different values of $\text{offerLoad}_{0\to1}$ in Figure 13. We omit the throughput improvement when d_1 is 120 m because the ARF throughput is zero most of the time. From the figure, RAF's throughput gain ranges from 31% to 596%.

5.1.2 Dynamic traffic pattern

We then study how RAF adapts when the traffic pattern of the interfering flow 0→1 changes. By fixing d_1 to 135 m, we initialize $\text{offerLoad}_{0\to1}$ to 5Mbps. At time 20 second, $\text{offerLoad}_{0\to1}$ changes to 2Mbps, then changes back to 5Mbps at time 35 second. Node 2 remains backlogged throughput the simulations. Figure 15 shows the instantaneous throughput over 1 second period normalized by 11Mbps - the highest available channel rate. From the figure, it is clear that within 2 seconds RAF adjusts itself and achieves throughput 4 times as before. On the other hand, ARF stays at the same throughput level due to intermittent interferences, no matter how the interferences level changes. Figure 16 shows the corresponding channel rate and frame sizes used by flow 2→3 throughout the simulation. From the figure, node 2 switches to higher channel rate (11Mbps) when the interference level becomes mild, and switches to 1Mb when the interference becomes strong. Note that the optimal frame sizes quickly stabilize within 2∼3 seconds after traffic change. From Figure 15 and 16, it is easy to see RAF successfully adjusts the channel rate and frame size according to the changing interference level and achieve consistent throughput improvement.

5.1.3 Node mobility

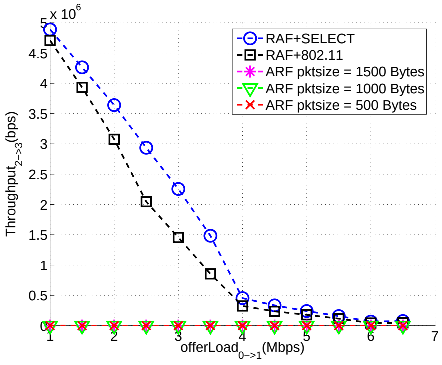


Figure 9: Throughput comparison between RAF and ARF for different offerLoad$_{0\rightarrow1}$(d_1 = 120m)

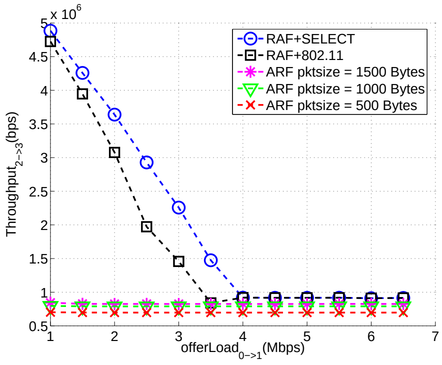


Figure 10: Throughput comparison between RAF and ARF for different offerLoad$_{0\rightarrow1}$(d_1 = 135m)

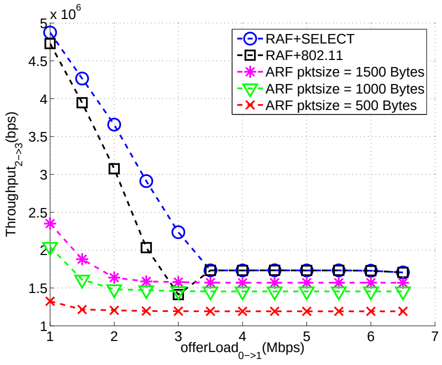


Figure 11: Throughput comparison between RAF and ARF for different offerLoad$_{0\rightarrow1}$(d_1 = 160m)

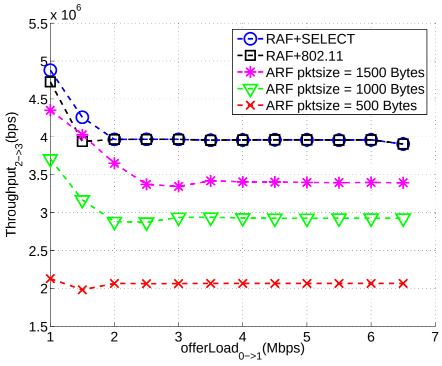


Figure 12: Throughput comparison between RAF and ARF for different offerLoad$_{0\rightarrow1}$(d_1 = 180m)

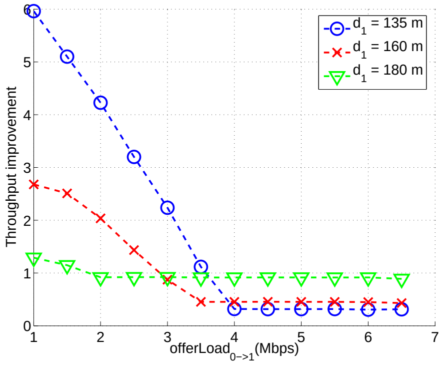


Figure 13: Throughput improvement of RAF over ARF for different offerLoad$_{0\rightarrow1}$, (d_1 = 135m, 160m, 180m)

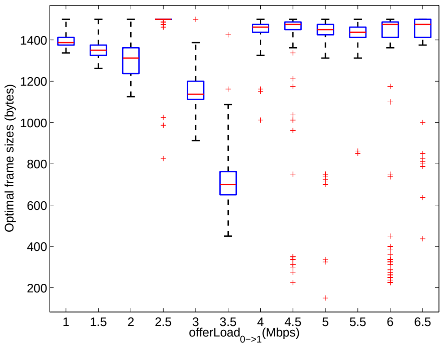


Figure 14: Optimal frame sizes for different offerLoad$_{0\rightarrow1}$ (d_1 = 135m). Optimal rate is 11Mbps for offer load < 4Mbps and 1Mbps otherwise

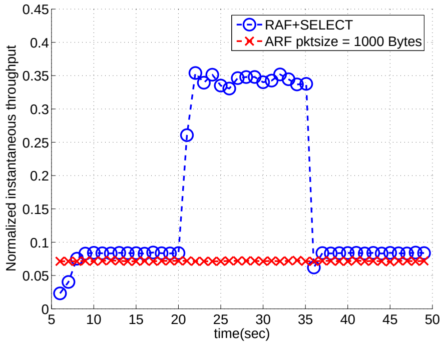


Figure 15: Throughput$_{2\rightarrow3}$ comparison between RAF and ARF when traffic pattern of flow$_{0\rightarrow1}$changes

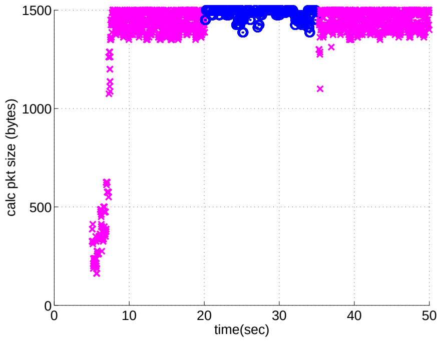


Figure 16: Channel rate and frame size calculated at node 3 when traffic pattern of flow 0→1 changes (○: 11Mbps, ×: 1Mbps)

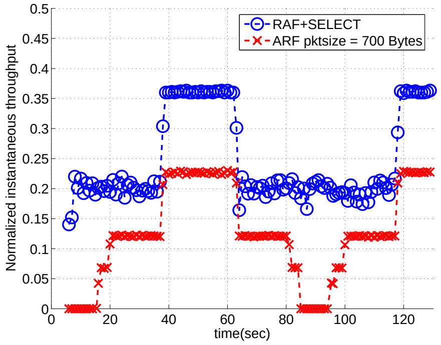
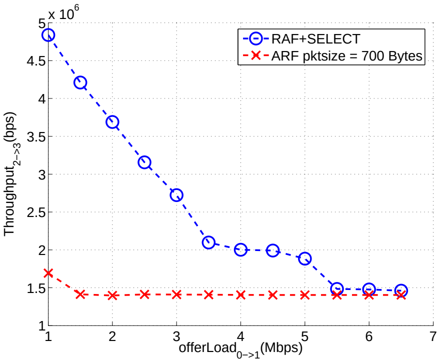
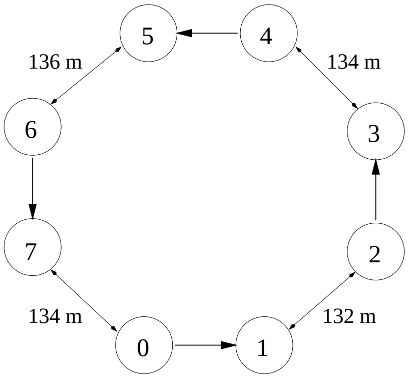


Figure 17: Instantaneous throughput of flow$_{2\rightarrow3}$ over 1 sec period when node 0 is moving toward/away from node 1

We further study RAF's adaptivity when the interference level at node 3 continuously changes. The simulation runs from 5 sec to 130 sec. Node 0 is initially placed 120m away from node 3. At time 10sec, it starts moving back and forth from node 3 up to 200 m at 2 m/s, as shown in Figure 24. Figure 17 again shows the one-sec normalized instantaneous throughput for flow 2→3 when offer load of flow 0→1 is 3Mbps. Indeed, ARF shows its ability to adapt when interference level changes, but not to the level that optimizes the throughput as RAF does. As a result, RAF outperforms ARF during all the time periods. Note that during two intervals 5∼15sec and 85∼94sec, ARF's throughput drops to





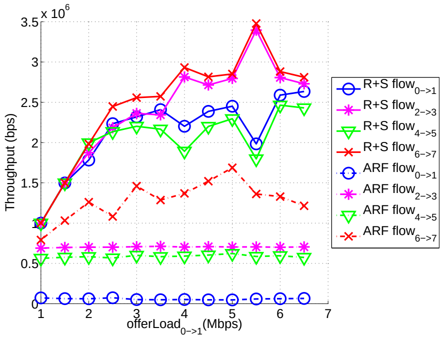
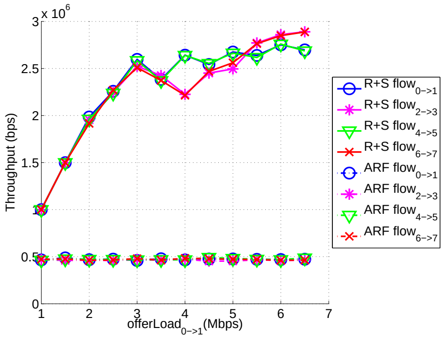
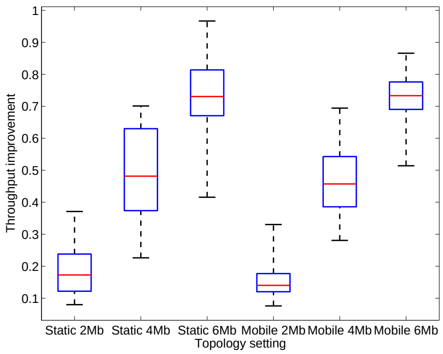


Figure 18: Throughput of flow$_{2\to3}$ when node 0 is moving toward/away from node 1, offerLoad$_{0\to1}$ ranges from 1Mb to 6.5Mb

Figure 19: Four-flow topology

Figure 20: Throughput of the four flows in Fig 19, offer load of each flow ranges from 1Mb to 6.5Mb





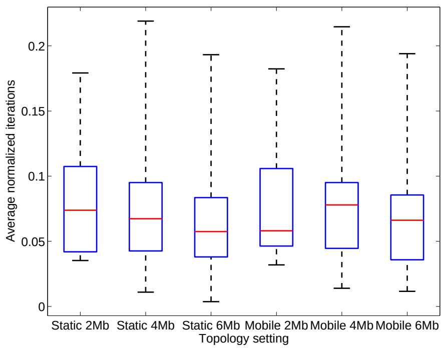


Figure 21: Throughput of the four flows for perfectly symmetric topology similar to Fig 19, offer load of each flow ranges from 1Mb to 6.5Mb

Figure 22: RAF's throughput gain over ARF in 29 random static/mobile topologies. Offer load ranges from 2Mb to 6Mb

Figure 23: Avg. normalized iterations for optimal rate and frame size calculation for 29 random static/mobile topologies, offer load ranges from 2Mb to 6Mb

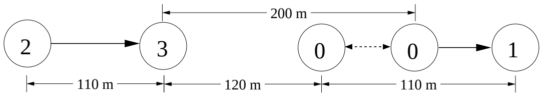


Figure 24: Topology where node 0 moves in an oscillatory manner

zero. This is due to the strong interference resulting from the short distance between node 0 and 3, and ARF's improper selection of low channel rate which aggravates the effect of hidden terminal interferences. Finally, Figure 18 shows the throughput when the offer load of flow 0→1 is varied from 1Mbps to 6.5Mbps. RAF again, performs better than ARF for all the interfering offer loads and improves by up to 298%.

5.2 Ring topology

We move on to a more complicated topology shown in Figure 19. In this topology, the senders of the flows are outside the carrier sensing range of each other, but their receivers are interfered by the next sender in the circle. Note that although the distance between each sender and receiver is fixed at 110 m, we intentionally disturb the inter-flow distances to avoid perfectly symmetrical topology, which is impossible in reality. Since each flow is suffering from the hidden terminal problem, we set the offer load of every flow equal. Figure 20 shows the throughput of each flow for RAF and ARF with offer load varied from 1Mbps to 6.5Mbps. As expected, RAF improves the throughput for every flow regardless of their complex interactions. Note that RAF's throughput for flow 0→1 and 4→5 tend to align with each other when offer load increases, so do flow 2→3 and 6→7. This is because for the current topology, each flow will be dominated by the next flow, while dominating the previous one. Better performance of one flow will result in worse performance of the adjacent two flows. Note also that ARF throughput for flow 0→1 is almost zero while flow 6→7's is the highest among the four. This is due to the asymmetric inter-flow distances, i.e. interference at node 1 is the most severe compared with the interference at other receivers. When flow 0→1 suffers from strong interferences, flow 6→7 benefits. In contrast, RAF avoids the starvation of flow 0→1, improves the throughput for all the flows consistently. For comparison, we plot the throughput of each flow in Figure 21 when the topology is unrealistically symmetric. In this case, both

ARF and RAF achieves almost perfect fairness among all competing flows, although RAF flow throughput is consistently higher than ARF flows by up to 622% when the total offered load saturates the channel.

5.3 Large random topologies

We finally study RAF's performance in large scale random topologies. Specifically, we randomly place 40 nodes in a 1000 m by 1000 m area, emulating a chaotic deployment of AP's and wireless stations as reported in the measurements [7]. We choose half of the 40 nodes to transmit/receive on the same channel, accommodating the potential growth. Besides, the 20 chosen nodes are organized into 10 flows. We generate 29 random topologies and the simulation runs for 20 seconds for each experiment. Our experiments can be categorized into 6 groups. The first 3 groups are static random topologies with offer load of each flow set to 2Mbps, 4Mbps, and 6Mbps respectively. For the other 3 groups, we enable node mobility and move each receiver back and forth, between 30m and 110m, from the sender at 2 m/s. Note that under the same moving speed this moving pattern causes the highest signal strength variation compared with the popular random way point mobility model.

As shown in Figure 22, we again use box plot to show the aggregate throughput improvement of RAF over ARF for the 29 topologies in each of the 6 groups. We can easily see the maximum, minimum, median, upper quartile, lower quartile of RAF improvements among these 6 groups. Indeed, RAF outperforms ARF for all the 29 topologies among all the three different rates for both static and mobile topologies. As a general trend for both static and mobile topologies, the higher the offered load (or the higher the interference in the network), the higher the performance gain RAF is able to achieve.

We finally show the effectiveness of our two proposed methods for the search of optimal rate and frame size given a recent history of channel idle intervals, as presented in Section 4.2. For each experiment, we calculate the average number of iterations, and normalize the averages by the total number of enumerations. Again we use box plot to show the average normalized number of iterations for each of the 29 topologies in the 6 groups. As shown in Figure 23, all the medians of the computation overhead in the 6 groups are within 9% of the brute-force enumerations (around 11 times faster), and we reduce the number of iterations by at least 75%.

6. CONCLUSION

As wireless devices defined in the unlicensed frequency bands proliferate, interference is becoming a dominating factor to the success or failure of a transmission. However, the majority of existing rate controls are interference oblivious. They often lead the system into a state where all interfered wireless transceivers operate at low data rate and the overall contention for the wireless channel stays high. In this paper we present rate adaptive framing (RAF), a joint optimal channel rate and frame size control that address both interference and noise for maximal throughput. The design of RAF leverages patterns of the interference, as a result of the spatial and temporal correlations of wireless traffic, and derives the optimal channel rate and frame size. An RAF transmitter obtains such optimal configuration from the ACK message, and applies it in the transmission of the next frame. Through analysis, intensive simulations and preliminary experiments we have shown that RAF consistently outperforms the *de facto* rate control ARF by up to six folds in throughput, under various levels of interference and traffic patterns. We are now adding power control into the RAF framework to realize better fairness between interfered and the interfering wireless transceivers.

7. REFERENCES

[1] Madwifi: Multiband Atheros driver for WiFi. `http://madwifi.org/`.

[2] ONOE. `http://madwifi.org/browser/trunk/ath_rate/onoe`.

[3] Place Lab. http://www.placelab.org/.

[4] Soekris net4801 communication computers. `http://www.soekris.com/`.

[5] WiFi Maps. http://www.wifimaps.com/.

[6] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *Proceedings of ACM SIGCOMM*, 2004.

[7] A. Akella, G. Judd, P. Steenkiste, and S. Seshan. Self management in chaotic wireless deployments. In *Proceedings of ACM MobiCom*, 2005.

[8] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, March 2000.

[9] J. C. Bicket. Bit-rate selection in wireless networks. Master's thesis, Department of EECS, MIT, February 2005.

[10] C. Chen and H. Luo. The case for heterogeneous wireless macs. In *Proceedings of HotNets*, 2005.

[11] C. Chen, E. Seo, and H. Luo. Self-learning collision avoidance for wireless networks. In *Proceedings of IEEE INFOCOM*, 2006.

[12] S. Choi, K. Park, and C. kwon Kim. On the performance characteristics of wlans: Revisited. In *Proceedings of ACM SIGMETRICS*, 2005.

[13] J. del Prado Pavon and S. Choi. Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement. In *Proceedings of ICC*, 2003.

[14] M. Garetto, J. Shi, and E. Knightly. Modeling media access in embedded two-flow topologies of multi-hop wireless networks. In *Proceedings of ACM MobiCom*, 2005.

[15] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips. Hybrid rate control for IEEE 802.11. In *Proceedings of ACM MobiWac*, 2004.

[16] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *Proceedings of ACM MobiCom*, 2001.

[17] C. Hu and J. C. Hou. A reactive channel model for expediting wireless network simulation. In *ACM SIGMETRICS Poster*, 2005.

[18] K. Jamieson, B. Hull, A. K. Miu, and H. Balakrishnan. Understanding the real-world performance of carrier sense. In *Proceedings of ACM SIGCOMM Workshop on experimental approaches to wireless network design and analysis*, 2005.

[19] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer. Understanding link-layer behavior in highly congested ieee 802.11b wireless networks. In *Proceedings of ACM SIGCOMM Workshop on experimental approaches to wireless network design and analysis*, 2005.

[20] G. Judd and P. Steenkiste. Using emulation to understand and improve wireless networks and applications. In *Proceedings of NSDI*, 2005.

[21] A. Kamerman and L. Monteban. WaveLAN-II: a high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, 2(3):118–133, August 1997.

[22] J. Kim, S. Kim, S. Choi, and D. Qiao. CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs. In *Proceedings of IEEE INFOCOM*, 2006.

[23] Y. Kim, S. Choi, K. Jang, and H. Hwang. Throughput enhancement of IEEE 802.11 WLAN via frame aggregation. In *Proceedings of IEEE VTC*, 2004.

[24] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. In *Proceedings of ACM MobiCom*, 2002.

[25] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 rate adaptation: A practical approach. In *Proceedings of ACM MSWiM*, 2004.

[26] X. G. Meng, S. H. Wong, Y. Yuanz, and S. Lu. Characterizing flows in large wireless data networks. In *Proceedings of ACM MobiCom*, 2004.

[27] J. Proakis. *Digital Communications*. McGraw-Hill, 2000.

[28] D. Qiao and S. Choi. Fast-responsive link adaptation for IEEE 802.11 WLANs. In *Proceedings of IEEE ICC*, 2005.

[29] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan. Measurement-based characterization of 802.11 in a hotspot setting. In *Proceedings of ACM SIGCOMM Workshop on experimental approaches to wireless network design and analysis*, 2005.

[30] A. Sang and S. qi Li. A predictability analysis of network traffic. In *Proceedings of IEEE INFOCOM*, 2000.

[31] Z. Wu, S. Ganu, I. Seskar, and D. Raychaudhuri. Experimental investigation of PHY layer rate control and frequency selection in 802.11-based ad-hoc networks. In *Proceedings of ACM SIGCOMM Workshop on experimental approaches to wireless network design and analysis*, 2005.