

Practical Power and Rate Control for WiFi

Thomas Huehn and Cigdem Sengul

Telekom Innovation Laboratories, TU-Berlin,

Email: thomas@net.t-labs.tu-berlin.de , cigdem@net.t-labs.tu-berlin.de

Abstract—While there has been extensive theoretical work on sophisticated joint resource allocation algorithms for wireless networks, their applicability to WiFi (IEEE 802.11) networks is very limited. One of the main reasons is the limitations in changing MAC parameters in current driver implementations. To this end, in this work, we developed a general cross-layer communication interface in the Linux kernel between the IEEE 802.11 PHY and MAC to enable per packet TPC. Based on this implementation, we realize a decentralized rate-power controller (Minstrel-Piano). Our initial evaluation shows that Minstrel-Piano is able to significantly decrease the power levels while maintaining the same link performance. These results are encouraging for a better interference management and consequently, better resource allocation in WiFi networks.

Index Terms—Power control, rate control, feed-back based, measurements, WiFi.

I. INTRODUCTION

Today's ubiquitous deployments of WiFi-based wireless networks have proven the feasibility of providing Internet access anywhere anytime. However, their performance is far below the achievable limits when multiple participants share the same frequency spectrum in an uncoordinated manner. There exists significant amount of work in literature on wireless resource allocation, which use sophisticated algorithms to select PHY level parameters such as power levels, carrier-sense parameters and transmission rates. For instance, several theoretical and practical works focus on independently performing power [1]–[6], rate [7]–[12] and carrier sense control [13]–[17], which aim at adjusting the power, rate and carrier sense parameters, respectively, according to the measured link quality. As the interactions among these mechanisms lead to interesting trade-offs [18], in this paper, we focus on their joint operation. Theoretical solutions in this context typically either require changes to the MAC (e.g., introducing a new frame type or adding additional information to Beacon frames) [19], [20], or assume cross-layer information from the MAC layer such as information about wireless neighbors and medium access state per link [20]. While indeed performance improvements can be expected with these approaches, they generally remain theoretical in nature and are not widely applied and evaluated in practical systems research. Therefore, our research aims to understand joint rate and power control capabilities in a real IEEE 802.11 system and to realize an adaptive rate-power controller by enabling a cross-layer communication interface between the IEEE 802.11 PHY and MAC layers.

To this end, starting from the MAC layer, we extend the Linux mac802.11 subsystem to allow both individual power level and modulation rate annotations per data packet including

the potential retries. Our design allows the MAC subsystem to respect different hardware and driver capabilities with respect to transmit power control (TPC). It covers the range from no per-packet power control to per-packet and per-retry power level control, which respects the wide range of capabilities in today's IEEE 802.11abgn hardware. Based on this design, we implement a per-link rate and power controller that works side-by-side on Atheros 802.11a radios. For this controller design, we use the results from an extensive measurement study in BOWL (Berlin Open Wireless Lab) outdoor wireless testbed¹. For instance, the observation that external noise has a significant impact on packet detection and received signal-strength indicator (RSSI) values led us to develop a controller that does not rely on signal-to-noise or channel state information. This also frees our solution from being dependent on the non-standard vendor-specific RSSI information. Instead, our algorithm *Minstrel-Piano* uses statistical feedback of IEEE 802.11 acknowledgment (ACK) packets obtained by sampling different rate and power settings. By controlling rate and power per link, our controller reduces the power level of each link to a level that still maintains the same packet success probability, if the maximum power level were to be used. This directly decreases the generated interference, and is expected to improve resource allocation in the network.

The main contributions of our work are:

- Enable hardware-independent TPC for IEEE 802.11abgn WiFi systems using the *mac80211* Linux kernel subsystem through our implementation of a general cross-layer communication interface between PHY and MAC layer.
- Design, implementation and validation of Minstrel-Piano, a decentralized per-packet and per-link rate and power controller, which does not rely on message passing.

The rest of the paper is organized as follows. In Section II, we present the related work. Section III describes in detail the design considerations, both in hardware and software, for Minstrel-Piano as well as the descriptions of the Minstrel rate control and Piano power control algorithms. In Section IV, we discuss our initial results based on measurements from a WiFi testbed. We conclude in Section V.

II. RELATED WORK

There are a variety of approaches to resource allocation in wireless networks including PHY layer control (e.g., power, rate, and carrier sense parameters), MAC layer link scheduling, and network layer routing and congestion control. In this

¹<http://www.bowl.tu-berlin.de>

paper, we focus on power, rate and carrier sense control as these works are the most relevant. There is a significant amount of both theoretical and practical work on independently performing power [1]–[6], rate [7]–[12] and carrier sense control [13]–[17]. However, the interactions among these mechanisms cannot be overlooked, as they lead to trade-offs [18]. In theory, one can improve spatial reuse, and consequently network capacity, by reducing the power or increasing the carrier sense threshold. This intuitively means “if you want to shout, you need to listen more carefully so as not to disturb those who are whispering” [20]. In [21], a joint transmit power and carrier sense adaptation is proposed relying on a mechanism that differentiates congestion and interference related losses. It is shown that by tuning the carrier sense threshold, it becomes possible to eliminate interference related losses, when the interference signal arrives prior to the data signal. On the other hand, power control suppresses losses when interference occurs after the arrival of the data signal.

New trade-offs are introduced when rate control is considered in addition to power and carrier sense control [22], [23]. Here, it becomes necessary to understand the trade-off between spatial reuse and the rate that can be supported. In [24], the spatial back-off concept was introduced, which allows dynamic tuning of carrier sense threshold together with ARF (Auto-Rate Fallback) algorithm to achieve high throughput. Here, ARF switches to a lower rate if the observed losses are higher than a threshold, and switches to a higher rate, if a certain number of consecutive frames were transmitted successfully. In contrast, in [19], it is shown that in the case of discrete data rates and when there are a sufficient number of power levels, tuning the power offers several advantages compared to carrier sense control. The authors propose PRC (Power and Rate Control), that tunes the power and rate of the transmitter based on the perceived interference level at the receiver. This, however, requires the receiver to piggyback this information to the transmitter, which might be achieved by IEEE 802.11k [25], but is not implemented in any of the current device drivers. In this paper, we also focus on the interaction between power and rate control but our approach works within the current IEEE 802.11 wireless drivers.

A dynamic rate and power adjustment algorithm that is compatible with IEEE 802.11 deployments is presented in [26]. The proposed scheme relies on the reception of ACK frames, and operates using two simple adaptation strategies: (1) the highest possible rate is supported with the lowest power possible and (2) the lowest possible power is chosen, and next, the highest rate possible at this power is chosen. Similarly, in [27], PARF and PERF were proposed, where the authors extend ARF (as in [24]) and ERF (Estimated Rate Fallback) to work with TPC. Note that ERF is the SNR-based version of ARF, where each packet contains its power level and the path loss and noise estimate of the last packet received. Based on this, ERF sends estimate SNR and set the highest transmission rate that supports this SNR. The authors of [27] have observed that PARF did not perform well when the receiver decreased the power for ACK transmissions.

Essentially, this led to incorrect power decrease decisions at the transmitter when these ACK frames were lost. They obtain more stable performance with PERF, which bases the power and rate decisions on the SNR values. These results are inline with [28], which shows that SNR-based protocols are more robust compared to loss-based protocols. However, they also conclude that SNR-based protocols require in-situ training to ensure such robustness. Our measurements in the BOWL testbed also show that the RSSI values vary with external noise. Hence, in order not to rely on the specific driver-based RSSI computations, in Minstrel-Piano, we take a loss-based approach. Next, we present the design considerations and implementation of Minstrel-Piano, in detail.

III. DESIGNING A PRACTICAL POWER AND RATE CONTROLLER: MINSTREL-PIANO

Building a rate and power controller requires a clear understanding of transmission processes performed at the PHY layer for a given radio hardware. Therefore, in this section, we first present the WiFi-specific considerations that affect the design of the Minstrel-Piano. Next, we present our controller that brings together the Minstrel rate control algorithm and our power control solution, Piano.

A. WiFi Design Considerations

When considering the feasibility of rate and power control for WiFi radios, we observe that for data frames, the current radios are able to adjust their transmission rate on a per frame basis. More specifically, each IEEE 802.11 frame contains a *preamble*, which is used for signal detection by the receiving radio as well as for timing acquisition to find out when the payload actually begins. The preamble header also contains information about the used *transmission mode*, which defines a transmission rate that corresponds to a particular modulation order and channel coding rate. Note that, management frames, e.g., ACK frames, are typically sent at the lowest rate. The IEEE 802.11ag [29] standard defines also two additional higher rates, 12 and 24Mbps for ACKs.

Using the Atheros-based radios, for each data packet, it is possible to set four different rates and their corresponding number of retries in a so called multi-rate-retry (mrr) chain. If the packet is successfully transmitted, the remainder of the mrr chain is ignored. For instance, it is possible to try sending a packet first at 54Mbps 4 times, and on failures, using 24Mbps 3 times, next at 9Mbps twice, and finally, at 6Mbps again twice. Essentially, the current rate control algorithms operate based on this ability to configure mrr chain (see Section III-B for details of the Minstrel rate control algorithm).

While per-frame transmission rate control is typical, per-frame power control is not typical. Only several chipset vendors, in particular Atheros, have implementations that permit per-packet adjustments [30]. It was validated in [31] that with Atheros chipsets it is possible to perform per-packet power control with granularity of 0.5dBm and switching latency of 1ms. Our experience with the Atheros 5212 chipset also confirms this. Note that these chipsets support one power level

per mrr chain (i.e., retries are also done in the same power level). With newer Atheros 802.11n chips, such as AR938X, it is also possible to use a different power level per retry chain entry. Finally, to meet the tight timing constraints, ACKs are generated by the PHY layer of the WiFi hardware. Therefore, power setting of ACKs is restricted to using a single global power level.

To control the per-packet rate levels, the Linux kernel uses a control structure (in addition to the packet buffer) which carries the rate and retry annotations of a given data packet on its way to the driver for transmission (tx-path). We extended this control structure to support annotating four power levels, covering the full retry chain. On receiving the control structure, the driver, in our case, the Atheros device driver, forms a tx-descriptor that represents the rate and power settings to be used by the WiFi hardware. After a packet is sent, this structure is overwritten with the status annotation (tx-feedback), which contains at which rate and after how many retries the packet succeeded (if at all).

B. Minstrel-Piano

Our controller Minstrel-Piano controls both transmission rate and power. To control the transmission rate per-packet, the rate and retry annotations on the *tx-path* are done by Minstrel. We propose the Piano algorithm to perform the per-packet power annotations. In the rest of the section, we explain these two algorithms and how they operate together.

1) *Minstrel Rate Control*: The core of Minstrel rate control algorithm is setting the mrr chain, which contains the number of retries for the next transmission rate when packet delivery fails at the current rate. In Minstrel, these the retry rate preferences are named as best throughput, next best throughput, best probability of success, and lowest baserate. Each of these preferences is associated with a specific transmission rate. To determine these rate sets, Minstrel uses a per-link (i.e., per-neighbor) table that keeps the probability of success and achievable throughput for each rate. Each 100ms, Minstrel evaluates this statistics table and uses a smoothing mechanism, called Exponential Weighted Moving Average (EWMA) to process the success history of each rate, R , as follows:

$$p_{success}^+[R] = ((1-\alpha) \times \frac{success[R]}{attempt[R]}) + (\alpha \times p_{success}[R]), \quad (1)$$

where $p_{success}^+[R]$ is the new success probability, $p_{success}[R]$ is the old success probability, $success[R]$ is the number of packets sent successfully in the current interval, and $attempt[R]$ is the number of attempts. Here, α is the smoothing factor. Given $p_{success}^+[R]$, the maximum achievable throughput is calculated as $R \times p_{success}^+[R]$. While we used the default α value, 0.75, for our experiments, dynamically adapting α can improve performance by 19% [32]. Nevertheless, even with default settings, Minstrel was shown to achieve better throughput performance compared to other rate control algorithms (e.g., Onoe, SampleRate [7]) [9].

To calculate per-rate success probabilities, Minstrel takes a fixed amount, 10%, out of all data packets per destination

TABLE I
HOW DOES MINSTREL FILL THE RETRY CHAIN?

Retry chain	Sampling packet		Data
	Random < Best T	Random > Best T	
Rate 1	Best T	Random	Best T
Rate 2	Random	Best T	2nd Best T
Rate 3	Best Prob	Best Prob	Best Prob
Rate 4	Basic rate	Basic rate	Basic rate

and assign a randomly chosen transmission rate. As mentioned earlier, the Atheros chipset supports four rate and retry preferences per packet. Minstrel fills in these values using the heuristic in Table I, where *Best T* refers to best throughput rate, and *Best prob* denotes the rate with the best success probability. *Random* is a randomly chosen rate. According to the table, while Minstrel, by default, assigns the 3rd rate as the best probability rate, and the 4th rate as the lowest basic rate, the 1st and 2nd rates are chosen such that unnecessary sampling of lower performance rates are avoided.

2) *Piano Power Control*: The goal of the Piano power control algorithm is to support the same rate, and hence, the throughput that Minstrel achieves, with the lowest power possible. Therefore, we first try to understand the relationship between the Atheros power settings and the set of WiFi rates through measurements. As an example, Fig. 1 shows the throughput results from one link in our 5GHz outdoor WiFi network [33]. For this scenario, we used 2 different carrier sense settings possible in the ath5k driver: (1) the highest receiver sensitivity setting with energy and preamble detection and (2) the lowest receiver sensitivity setting with only energy detection [29]. Switching between these two extremes allows us to emulate different channel conditions at the receiver based on its sensitivity and understand how Piano behaves under these conditions. The figure shows that, when (1) is used, the transmission power can be reduced to 1dBm when the rates are 6 – 18Mbps. On the other hand, with (2), the power can be reduced to \approx 16dBm without affecting the throughput. For 24Mbps, these power levels are 3dBm and 16dBm, respectively. The figure also shows that this link cannot support rates higher than 24Mbps. These results confirm that throughput is a non-decreasing function of transmission power, and that it is possible to maintain the same throughput at lower power levels. Albeit the different carrier-sense conditions may mandate different minimum power levels. We design the Piano algorithm based on these principles.

To decide at which power level to operate, Piano uses 3 types of packets: (1) Reference packets are sent at a reference power (typically a high power level). (2) Sample packets are sent at different power levels for exploration. (3) Data packets are sent at a power level equal to sample power level plus a constant, Δ ($\Delta = 2\text{dB}$ in our experiments). By sending packets at different power levels, Piano aims to understand the impact of transmission power changes on throughput.

Using these three different types of packets, the algorithm operates in two parts, as shown in Fig. 2: (1) It records

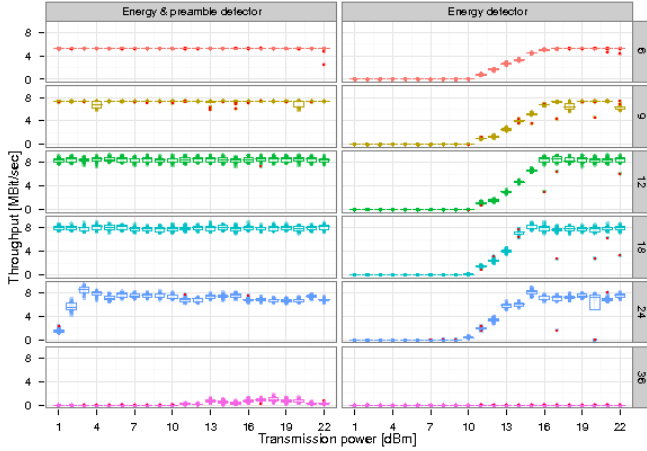


Fig. 1. Throughput, rate and power relationship under two carrier sense settings: (1) energy and preamble detector and (2) only energy detector.

number of successes and attempts obtained from the transmission feedback. (see COLLECT_STATS in Fig. 2), and (2) if enough packets are sampled for a given rate (determined by the min_update parameter), an update mechanism adjusts the power levels for each type of packet (i.e., $P_ref[R]$, $P_sample[R]$, $P_data[R]$) based on this history of attempts and successes (see UPDATE_STATS in Fig. 2). To make these adjustments, similar to Minstrel, Piano also maintains an EWMA of reference, sample, and data success probabilities, which are denoted as $p_{success}^{ref}[R]$, $p_{success}^{sample}[R]$, $p_{success}^{data}[R]$, respectively.

The main idea of the update mechanism is to keep the sample and reference success probabilities close (see UPDATE_STATS, comment (1) in Fig. 2). Therefore, the sample power is incremented if $p_{success}^{sample}[R]$ is below $p_{success}^{ref}[R]$ by a δ_{inc} threshold. The power increment is given as Δ_{inc} and the maximum power level that can be set is P_{max} . On the other hand, Piano reduces the sample power if $p_{success}^{sample}[R]$ is higher than $p_{success}^{ref}[R]$ minus a δ_{dec} threshold. The power decrement is denoted as Δ_{dec} and the minimum power level that can be set is P_{min} . Note that, here, the comparison is made between data and reference success probability, as the power of data packets are determined based on sample packets. Essentially, $P_data[R] = P_sample[R] + \Delta$ (UPDATE_STATS, comment (3)), where Δ is the power separation between data and sample packets. Piano updates the reference power similarly, but uses 1 as the comparison success probability (UPDATE_STATS, comment (2)).

When sampling different power levels, Piano does not interfere with Minstrel sampling. Hence, only when the packet is not a Minstrel sampling packet, Piano decides whether to send this packet as a reference, sample or a data packet. Since the power level used for sampling depends on the rate, in Piano, we alternate between the best and 2nd best throughput rates. If the 2nd best throughput rate is selected for power sampling, then we also change the rate order in the retry chain of the data packet. This helps to build a better history of power

Algorithm III.1: PIANO POWER CONTROL()

global $min_update, min_P, max_P, \Delta_{inc}, \Delta_{dec}, \Delta, \delta_{inc}, \delta_{dec}$

procedure INIT()

$\forall R :$

$P_ref[R], P_data[R] \leftarrow max_P$

$P_sample[R] \leftarrow P_ref[R] - \Delta$

procedure COLLECT_STATS($tx_feedback$)

comment: For each rate R in the retry chain

for each $R \in tx_feedback.retry_chain$
do

$success \leftarrow tx_feedback.get_ACK(R);$

if $tx_feedback.power == ref_power[R]$

then $\{ attempt_ref[R] ++;$

$success_ref[R] += success;$

else if $tx_feedback.power == data_power[R]$

then $\{ attempt_data[R] ++;$

$success_data[R] += success;$

else if $tx_feedback.power == sample_power[R]$

then $\{ attempt_sample[R] ++;$

$success_sample[R] += success;$

if $attempt_sample[R] > min_update$ **or**
 $attempt_ref[R] > min_update$

then UPDATE_STATS(R)

procedure UPDATE_STATS(R)

EWMA($\alpha, p_{success}^{ref}[R], \frac{success_ref[R]}{attempt_ref[R]}$)

EWMA($\alpha, p_{success}^{data}[R], \frac{success_data[R]}{attempt_data[R]}$)

EWMA($\alpha, p_{success}^{sample}[R], \frac{success_sample[R]}{attempt_sample[R]}$)

comment: (1) Update sample power P_sample

INC_POWER($P_sample[R], p_{success}^{sample}[R], p_{success}^{ref}[R]$)

DEC_POWER($P_sample[R], p_{success}^{data}[R], p_{success}^{ref}[R]$)

comment: (2) Update ref power P_ref

INC_POWER($P_ref[R], p_{success}^{ref}[R], 1$)

DEC_POWER($P_ref[R], p_{success}^{sample}[R], 1$)

comment: (3) Update data power P_data

$P_data[R] = P_sample[R] + \Delta$

procedure INC_POWER($P, p1, p2$)

if $p1 < p2 - \delta_{inc}$

then $P = \min(max_P, P + \Delta_{inc})$

procedure DEC_POWER($P, p1, p2$)

if $p1 > p2 - \delta_{dec}$

then $P = \max(min_P, P - \Delta_{dec})$

Fig. 2. Piano power control algorithm

Algorithm III.2: PIANO-MINSTREL()

```

main
  if sending Pkt
    if  $Pkt! = Minstrel\_sample\_pkt$ 
       $R1 \leftarrow Minstrel\_rate\_BestT$ 
       $R2 \leftarrow Minstrel\_rate\_2ndBestT$ 
       $R \leftarrow (sample\_best?R1 : R2)$ 
      comment: Alternate R1 and R2
      if  $sample\_best$ 
        then  $sample\_best = false$ 
        else  $\{sample\_best = true$ 
           $type \leftarrow IS\_SAMPLE\_OR\_REF()\}$ 
          if  $type == Sample$ 
            then  $Send\_P = P\_sample[R]$ 
            else if  $type == Ref$ 
              then  $Send\_P = P\_ref[R]$ 
              else  $Send\_P = P\_data[R]$ 
          comment: Don't touch Minstrel samples
          break
      else if received tx-feedback
        then COLLECT_STATS( $tx\_feedback$ )

```

Fig. 3. Piano-Minstrel interaction.

sampling, especially when Minstrel is consistently switching between the two different rates. Based on the selected rate R , the power level of the packet, $Send_P$, is set respectively as reference power, $P_ref[R]$, sample power, $P_sample[R]$, or data power, $P_data[R]$. For each sent packet, Piano uses the tx-feedback to collect statistics. The pseudo-code of Minstrel-Piano interaction is presented in Fig. 3.

IV. MEASUREMENTS FOR VALIDATION

The goal of our validation is to show that Piano reduces the power while maintaining the same throughput. To this end, we focused on a single link scenario, and confirmed that Piano performs well both with fixed transmit rate settings as well as alongside Minstrel. Our evaluations also take into account two different carrier-sense settings: (1) energy and preamble detection (ED & PD) and (2) only energy detection (ED). We look at two metrics: (1) SNR (dB) and (2) throughput (Bits/s) at the receiver. These values are reported for all frames (i.e., sample and reference frames as well as data frames).

A. Measurement Set-up

All the measurements were performed in the BOWL testbed [33], an outdoor wireless mesh network deployed on the rooftop of the TU-Berlin campus. The network comprises two separate networks, a 50-node network equipped with Avila Gateworks GW2348-4 motherboard, and a 13-node network with Asus WL-500GP routers. For this study, we used the

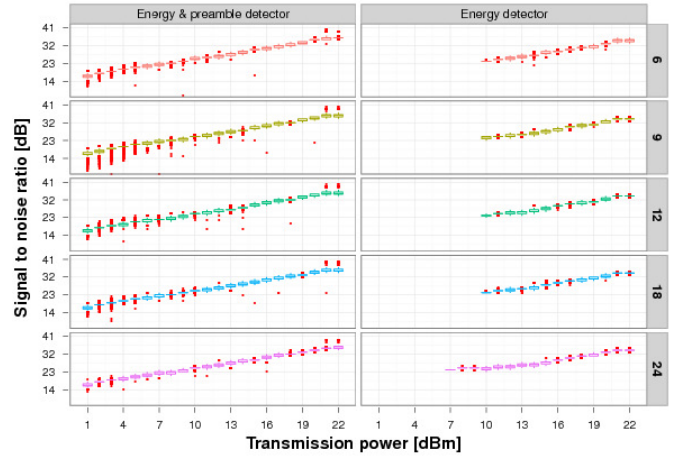


Fig. 4. SNR at the receiver based on different transmission power and carrier sense settings: (1) Energy and preamble detector (2) only energy detector.

Asus network. Each Asus node has a MIPS 266MHz CPU, 32MB RAM, 8MB flash 1 miniPCI port and 2xUSB interfaces. They are also attached 12dB omni-directional antennas. We have both wireless and wired access to the nodes: the wireless interfaces are Atheros DCMA-82 miniPCI cards with an 5212 chipset. We use the 100Mbps Ethernet connection for measurement collection.

Nodes run a customized version of the OpenWrt operating system, a GNU/Linux distribution for embedded devices [34] and the ath5k driver. Nodes communicate in the ad-hoc mode using the IEEE 802.11a standard. Diversity and Atheros ambient noise immunity (ANI) are switched off. Dynamic frequency selection (DFS) is also disabled. For traffic generation, we used Iperf version 2.0.5 to generate UDP traffic with a constant datagram size of 1420B. The sending rate is chosen 32Mbps to ensure all lower layers are always saturated.

B. Validation

In this section, we first validate Piano using fixed rates. Next, we show that Piano is also able to work with Minstrel well. To serve as a baseline, we ran experiments on a single link (i.e., with one sender and one receiver). Fig. 4 presents the SNR at the receiver for different power settings for rates 6–24Mbps. (The corresponding receiver throughput per power level was shown earlier in Fig. 1). Fig. 4 shows that SNR at the receiver increases linearly with increasing power. Note that the higher SNR does not necessarily mean higher throughput, and with ED & PD, it is possible to obtain the same high throughput with a power level as low as 0dBm for rates 6–18Mbps.

1) *Piano with Fixed Rates:* To confirm that Piano reduces the transmission power as long as there is no penalty in throughput, we switched the sender to use different transmissions rates, while Piano is running. Fig. 5 shows the SNR at the receiver over time. We observe that with both carrier sensing settings, Piano is indeed able to reduce the power while maintaining similar throughput (see Fig. 6). For the

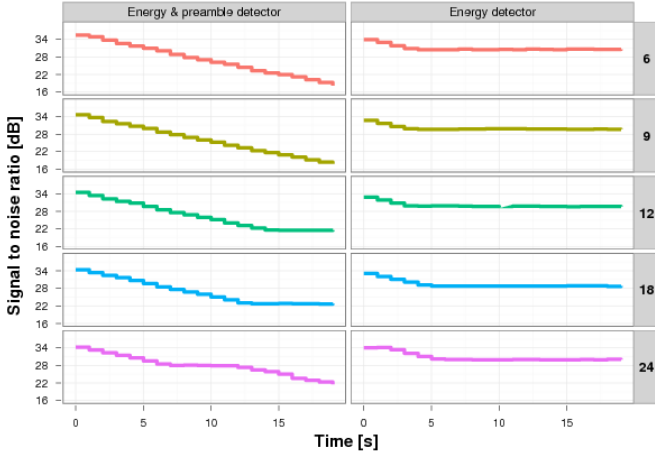


Fig. 5. SNR at the receiver: (1) Piano with energy and preamble detector (2) Piano with energy detector.

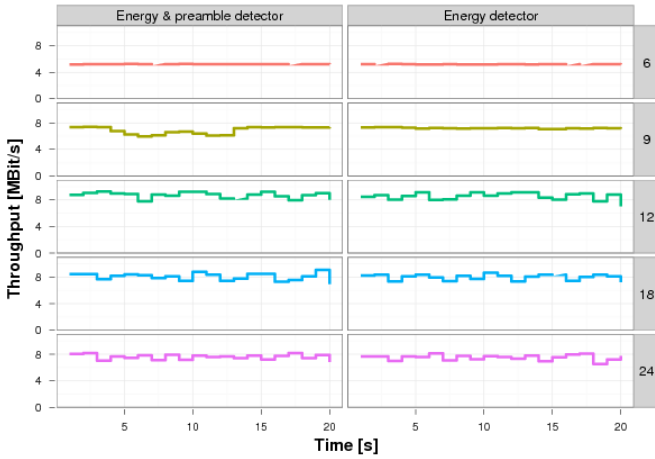


Fig. 6. Receiver throughput: (1) Piano with energy and preamble detector (2) Piano with energy detector.

case with ED & PD, the reduction is more significant (as expected from Fig. 4). For the rates 6–24Mbps, the saturation throughput is achieved when the SNR values are ≈ 16 –22dB. (This can be seen from looking up the power level from Fig. 1 and checking the SNR value for this power level in Fig. 4). We see that Piano successfully reduces the power level to attain these SNR levels at the receiver (see Fig. 5). On the other hand, with only energy detector, the power reduction opportunities are limited. Here, Piano maintains the SNR values around 26–28dB, which corresponds to the point where the receiver can attain higher throughput (see again Figs. 4 and 1).

2) *Minstrel-Piano*: Next, we validate the joint operation of Piano and Minstrel. To this end, we first start Minstrel with maximum transmission power using only energy detector. The SNR and the throughput at the receiver during this time are shown in Figs. 7 and 8. We observe that Minstrel throughput is quite stable. In our experiment after 120s, Piano is started. We see that with Piano, the SNR values at the receiver reduce to ≈ 30 dB, which is the expected operation level with

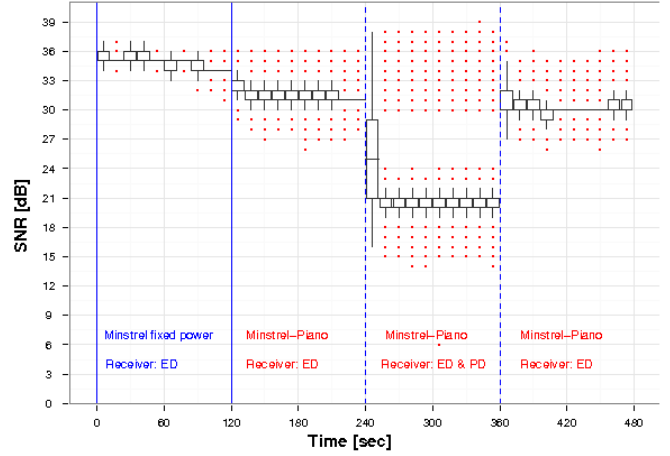


Fig. 7. SNR at the receiver: (1) Minstrel-full power with ED (0-120s) (2) Minstrel-Piano with ED (120-240s), (3) Minstrel-Piano with ED & PD (240-360s) and (4) Minstrel-Piano back to only using ED (360-480s).

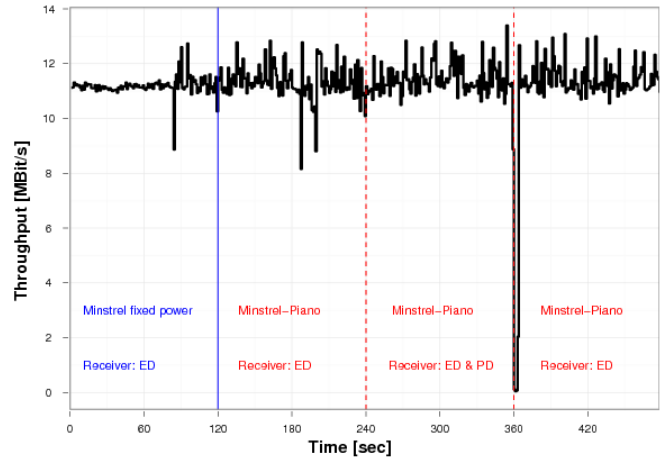


Fig. 8. Receiver throughput: (1) Minstrel-full power with ED (0-120s) (2) Minstrel-Piano with ED (120-240s), (3) Minstrel-Piano with ED & PD (240-360s) and (4) Minstrel-Piano back to only using ED (360-480s).

only energy detector for 24Mbps without negatively impacting the throughput. Again, after 120s, the preamble detector is activated. In this setting, the SNR further reduces to ≈ 21 dB, while the same throughput is maintained. SNR values increase back to ≈ 30 dB, when the preamble detector is disabled at around 360s. We see that this causes a short glitch and leads to high variation in power levels as well as a throughput drop. However, Minstrel-Piano recovers fast and Piano bumps up the power to maintain the same throughput as before. Finally, Fig. 9 depicts the number of times the data packets were received at different rates at different intervals. Note that in all intervals 24Mbps is the dominant rate (Fig. 9 is in log-scale). Note that Minstrel also tries the higher rate 36Mbps at times, which leads to consequent throughput degradation (e.g., around 90s). These results confirm that Piano is able to reduce the power without affecting the rate selection in Minstrel, and throughput.

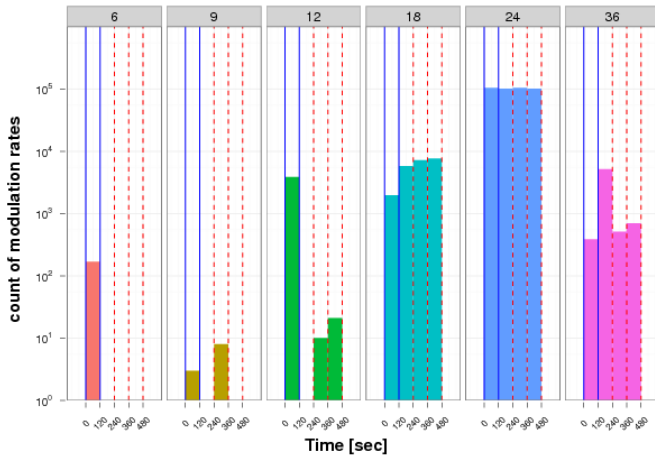


Fig. 9. Rate distribution of received data packets. (1) 0-120s: Minstrel-full power with ED, (2) 120-240: Minstrel-Piano with ED, (3) 240-360: Minstrel-Piano with ED & PD, and (4) 360-480: Minstrel-Piano again only using ED. The y-axis is in log-scale.

V. CONCLUSION

As more and more WiFi Access Points are deployed and recent IEEE 802.11n devices potentially using twice the channel width, efficient resource allocation is expected to become even more important to reduce the negative effects of interference. Minstrel-Piano controller has the potential to achieve efficient use of the shared spectrum, as it finds the necessary power level to provide the same link performance with only rate control. Currently, we are investigating the Minstrel-Piano performance in larger network scenarios with full and limited deployment to quantify its benefits. To support the open-source and research community, this work and all implementations will be released as GPL kernel code.

REFERENCES

- [1] A. Muqattash and M. Krunz, "A single-channel solution for transmission power control in wireless ad hoc networks," in *MobiHoc '04*, pp. 210–221.
- [2] A. Sheth and R. Han, "Shush: reactive transmit power control for wireless MAC protocols," in *1st Intl. Conf. on Wireless Internet, 2005*, 2005, pp. 18 – 25.
- [3] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks," in *1st IEEE Conf. on Sensor and Adhoc Communication and Networks*, 2004, pp. 289–298.
- [4] M. Sha, G. Xing, G. Zhou, S. Liu, and X. Wang, "C-MAC: model-driven concurrent medium access control for wireless sensor networks," in *IEEE INFOCOM*, 2009, pp. 1845 –1853.
- [5] N. Ahmed and S. Keshav, "Smarta: a self-managing architecture for thin access points," in *CoNEXT '06*, pp. 9:1–9:12.
- [6] V. Shrivastava, D. Agrawal, A. Mishra, S. Banerjee, and T. Nadeem, "Understanding the limitations of transmit power control for indoor w lans," in *IMC '07*, pp. 351–364.
- [7] R. T. Morris, J. C. Bicket, and J. C. Bicket, "Bit-rate selection in wireless networks," MS thesis, MIT, Tech. Rep., 2005.
- [8] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: a practical approach," in *MSWiM '04*, pp. 126–134.
- [9] W. Yin, P. Hu, J. Indulska, and K. Bialkowski, "Performance of mac80211 rate control mechanisms," in *MSWiM '11*, pp. 427–436.
- [10] W. Yin, K. Bialkowski, J. Indulska, and P. Hu, "Evaluations of madwifi MAC layer rate control mechanisms," in *IWQoS*, 2010, pp. 1–9.

- [11] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *SIGCOMM '09*, pp. 3–14.
- [12] K. LaCurtis and H. Balakrishnan, "Measurement and analysis of real-world 802.11 mesh networks," in *IMC '10*, pp. 123–136.
- [13] J. Zhu, B. Metzler, X. Guo, and Y. Liu, "Adaptive csma for scalable network capacity in high-density wlan: A hardware prototyping approach," in *IEEE INFOCOM*, 2006, pp. 1–10.
- [14] X. Yang and N. Vaidya, "On physical carrier sensing in wireless ad hoc networks," in *IEEE INFOCOM*, vol. 4, 2005, pp. 2525 – 2535.
- [15] J. Zhu, X. Guo, L. Yang, and W. Conner, "Leveraging spatial reuse in 802.11 mesh networks with enhanced physical carrier sensing," in *IEEE ICC*, vol. 7, 2004, pp. 4004 – 4011.
- [16] K.-J. Park, J. Hou, T. Basar, and H. Kim, "Noncooperative carrier sense game in wireless networks," *IEEE Trans. on Wireless Commun.*, vol. 8, no. 10, pp. 5280 –5289, Oct. 2009.
- [17] H. Ma, R. Vijayakumar, S. Roy, and J. Zhu, "Optimizing 802.11 wireless mesh networks based on physical carrier sensing," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1550 –1563, Oct. 2009.
- [18] T. Huehn, R. Merz, and C. Sengul, "Joint transmission rate, power, and carrier sense settings: An initial measurement study," in *WIMESH*, 2010, pp. 1 –6.
- [19] T.-S. Kim, H. Lim, and J. C. Hou, "Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks," in *MobiCom '06*, pp. 366–377.
- [20] V. P. Mhatre, K. Papagiannaki, and F. Baccelli, "Interference mitigation through power control in high density 802.11 w lans," in *IEEE INFOCOM*, 2007, pp. 535 –543.
- [21] H. Ma, J. Zhu, S. Roy, and S. Y. Shin, "Joint transmit power and physical carrier sensing adaptation based on loss differentiation for high density IEEE 802.11 WLAN," *Comput. Netw.*, vol. 52, pp. 1703–1720, June 2008.
- [22] F. A. Tobagi and M. M. Hira, "Joint optimization of physical layer parameters and routing in wireless mesh networks," in *Med-Hoc-Net*, 2010, pp. 1 –8.
- [23] K. Jamieson, B. Hull, A. K. Miu, and H. Balakrishnan, "Understanding the Real-World Performance of Carrier Sense," in *ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND)*, 2005.
- [24] X. Yang and N. H. Vaidya, "A spatial backoff algorithm using the joint control of carrier sense threshold and transmission rate," in *SECON*, 2007, pp. 501–511.
- [25] "IEEE standard for information technology telecommunications and information exchange between systems local and metropolitan area networks specific requirements part 11: Wireless lan medium access control (MAC) and physical layer (phy) specifications amendment 1: Radio resource measurement of wireless lans," 2008.
- [26] P. Chevillat, J. Jelitto, and H. L. Truong, "Dynamic data rate and transmit power adjustment in ieee 802.11 wireless lans," *Intl. Journal of Wireless Information Networks*, vol. 12, pp. 123–145, 2005.
- [27] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, "Self-management in chaotic wireless deployments," in *MobiCom '05*, pp. 185–199.
- [28] J. Camp and E. Knightly, "Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation," *IEEE/ACM Trans. Netw.*, vol. 18, no. 6, pp. 1949–1962, Dec. 2010.
- [29] "IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications," 2007.
- [30] F. B. Abdesslem, L. Iannone, M. D. de Amorim, K. Kabassanov, and S. Fdida, "On the feasibility of power control in current IEEE 802.11 devices," in *4th annual IEEE Intl. Conf. on Pervasive Computing and Communications Workshops*, 2006.
- [31] K. Kowalik, M. Bykowski, B. Keegan, and M. Davis, "Practical issues of power control in ieee 802.11 wireless devices," in *Intl. Conf. on Telecommunications (ICT)*, 2008, pp. 1–5.
- [32] W. Yin, P. Hu, J. Indulska, and K. Bialkowski, "A method to improve adaptability of the minstrel MAC rate control algorithm," in *7th Intl. Conf. on Ubiquitous intelligence and computing*, 2010, p. 504518.
- [33] T. Fischer, T. Hühn, R. Kuck, R. Merz, J. Schulz-Zander, and C. Sengul, "Experiences with bowl: Managing an outdoor wifi network (or how to keep both internet users and researchers happy?)," in *Usenix LISA*, 2011, pp. 287–292.
- [34] <http://openwrt.org/>.