On the energy efficiency of rate and transmission power control in 802.11

Iñaki Ucar [a,*], Carlos Donato [b], Pablo Serrano [a], Andres Garcia-Saavedra [c], Arturo Azcorra [a,b], Albert Banchs [a,b]

[a] *Universidad Carlos III de Madrid, Leganés, 28911, Spain*
[b] *IMDEA Networks Institute, Leganés, 28918, Spain*
[c] *NEC Labs Europe, Heidelberg, 69115, Germany*

ARTICLE INFO

ABSTRACT

Rate adaptation and transmission power control in 802.11 WLANs have received a lot of attention from the research community, with most of the proposals aiming at maximising throughput based on network conditions. Considering energy consumption, an implicit assumption is that optimality in throughput implies optimality in energy efficiency, but this assumption has been recently put into question. In this paper, we address via analysis, simulation and experimentation the relation between throughput performance and energy efficiency in multi-rate 802.11 scenarios. We demonstrate the trade-off between these performance figures, confirming that they may not be simultaneously optimised, and analyse their sensitivity towards the energy consumption parameters of the device. We analyse this trade-off in existing rate adaptation with transmission power control algorithms, and discuss how to design novel schemes taking energy consumption into account.

1. Introduction

In recent years, along with the growth in mobile data applications and the corresponding traffic volume demand, we have witnessed an increased attention towards "green operation" of networks, which is required to support a sustainable growth of the communication infrastructures. For the case of wireless communications, there is the added motivation of a limited energy supply (i.e., batteries), which has triggered a relatively large amount of work on energy efficiency [1]. It turns out, though, that energy efficiency and performance do not necessarily come hand in hand, as some previous research has pointed out [2,3], and that a criterion may be required to set a proper balance between them.

This paper is devoted to the problem of rate adaptation (RA) and transmission power control (TPC) in 802.11 WLANs from the energy consumption's perspective. RA algorithms are responsible for selecting the most appropriate modulation and coding scheme (MCS) to use, given an estimation of the link conditions, and have received a vast amount of attention from the research community (see e.g. [4,5] and references therein). In general, the challenge lies

in distinguishing between those loses due to collisions and those due to poor radio conditions, because they should trigger different reactions. In addition, the performance figure to optimise is commonly the throughput or a related one such as, e.g., the time required to deliver a frame.

On the other hand, network densification is becoming a common tool to provide better coverage and capacity. However, densification brings new problems, especially for 802.11, given the limited amount of orthogonal channels available, which leads to performance and reliability issues due to RF interference. In consequence, some RA schemes also incorporate TPC, which tries to minimise the transmission power (TXP) with the purpose of reducing interference between nearby networks. As in the case of "vanilla" RA, the main performance figure to optimise is also throughput.

It is generally assumed that optimality in terms of throughput also implies optimality in terms of energy efficiency. However, some previous work [6,7] has shown that throughput maximisation does not result in energy efficiency maximisation, at least for 802.11n. However, we still lack a proper understanding of the causes behind this "non-duality", as it may be caused by the specific design of the algorithms studied, the extra consumption caused by the complexity of MIMO techniques, or any other reason. In fact, it could be an inherent trade-off given by the power

* Corresponding author.
E-mail addresses: inaki.ucar@uc3m.es (I. Ucar), carlos.donato@imdea.org (C. Donato), pablo@it.uc3m.es (P. Serrano), andres.garcia.saavedra@neclab.eu (A. Garcia-Saavedra), azcorra@it.uc3m.es (A. Azcorra), banchs@it.uc3m.es (A. Banchs).

consumption characteristics of 802.11 interfaces, and, if so, RA-TPC techniques should not be agnostic to this case.

This work tackles the latter question from a formal standpoint. A question which, to the best of the authors' knowledge, has never been addressed in the literature. For this purpose, and with the aim of isolating the variables of interest, we present a joint goodput (i.e., the throughput delivered on top of 802.11) and energy consumption model for single 802.11 spatial streams in the absence of interfering traffic. Packet losses occur due to poor channel conditions and RA-TPC can tune only two variables: MCS and TXP.

Building on this model, we provide the following contributions: (*i*) we demonstrate through an extensive numerical evaluation that energy consumption and throughput performance are different optimisation objectives in 802.11, and not only an effect of MIMO or certain algorithms' suboptimalities; (*ii*) we analyse the relative impact of each energy consumption component on the resulting performance of RA-TPC, which serves to identify the critical factors to consider for the design of RA-TPC algorithms; (*iii*) we experimentally validate our numerical results; and (*iv*) we assess the performance of several representative RA-TPC algorithms from the energy consumption's perspective.

The rest of this paper is organised as follows. In Section 2, we develop the theoretical framework: a joint goodput-energy model built around separate previous models. In Section 3, we provide a detailed analysis of the trade-off between energy efficiency and maximum goodput, including a discussion of the role of the different energy parameters involved. We support our numerical analysis with experimental results in Section 4. Section 5 explores the performance of RA-TPC algorithms from the energy consumption's perspective. Finally, Section 6 summarises the paper.

2. Joint goodput-energy model

In this section, we develop a joint goodput-energy model for a single 802.11 spatial stream and the absence of interfering traffic. It is based on previous studies about goodput and energy consumption of wireless devices. As stated in the introduction, the aim of this model is the isolation of the relevant variables (MCS and TXP) to let us delve in the relationship between goodput and energy consumption optimality in the absence of other effects such as collisions or MIMO.

Beyond this primary intent, it is worth noting that these assumptions conform with real-world scenarios in the scope of recent trends in the IEEE 802.11 standard development, namely, the amendments 11ac and 11ad, where device-to-device communications (mainly through beamforming and MU-MIMO) are of paramount importance.

2.1. Goodput model

We base our study on the work by Qiao et al. [8], which develops a robust goodput model that meets the established requirements. This model analyses the IEEE 802.11a Distributed Coordination Function (DCF) over the assumption of an AWGN (Additive White Gaussian Noise) channel without interfering traffic.

Let us briefly introduce the reader to the main concepts, essential to our analysis, of the goodput model by Qiao *et al.*. Given a packet of length l ready to be sent, a frame retry limit $n_{\max}$ and a set of channel conditions $\hat{s} = \{s_1, \ldots, s_{n_{\max}}\}$ and modulations $\hat{m} = \{m_1, \ldots, m_{n_{\max}}\}$ used during the potential transmission attempts, the expected effective goodput $\mathcal{G}$ is modelled as the ratio between the expected delivered data payload and the expected transmission time as follows:

$$\mathcal{G}(l, \hat{s}, \hat{m}) = \frac{\mathbb{E}[\text{data}]}{\mathbb{E}[\mathcal{D}_{\text{data}}]} = \frac{\Pr[\text{succ} \mid l, \hat{s}, \hat{m}] \cdot l}{\mathbb{E}[\mathcal{D}_{\text{data}}]} \tag{1}$$

where $\Pr[\text{succ} \mid l, \hat{s}, \hat{m}]$ is the probability of successful transmission conditioned to $l, \hat{s}, \hat{m}$, given by Eq. (5) in [8]. This model is valid as long as the coherence time is equal or greater than a single retry, i.e., the channel condition s_i is constant.

The expected transmission time is defined as follows:

$$\mathbb{E}[\mathcal{D}_{\text{data}}] = \left(1 - \Pr[\text{succ} \mid l, \hat{s}, \hat{m}]\right) \cdot \mathcal{D}_{\text{fail} \mid l, \hat{s}, \hat{m}} + \Pr[\text{succ} \mid l, \hat{s}, \hat{m}] \cdot \mathcal{D}_{\text{succ} \mid l, \hat{s}, \hat{m}} \tag{2}$$

where

$$\begin{aligned}
\mathcal{D}_{\text{succ} \mid l, \hat{s}, \hat{m}} = \sum_{n=1}^{n_{\max}} \Pr[n\text{succ} \mid l, \hat{s}, \hat{m}] \cdot \Bigg\{ & \sum_{i=2}^{n_{\max}} \left[\overline{T}_{\text{bkoff}}(i) \right. \\
& \left. + T_{\text{data}}(l, m_i) + \overline{\mathcal{D}}_{\text{wait}}(i)\right] \\
& + \overline{T}_{\text{bkoff}}(1) + T_{\text{data}}(l, m_1) + T_{\text{SIFS}} \\
& + T_{\text{ACK}}(m'_n) + T_{\text{DIFS}} \Bigg\}
\end{aligned} \tag{3}$$

is the average duration of a successful transmission and

$$\mathcal{D}_{\text{fail} \mid l, \hat{s}, \hat{m}} = \sum_{i=1}^{n_{\max}} \left[\overline{T}_{\text{bkoff}}(i) + T_{\text{data}}(l, m_i) + \overline{\mathcal{D}}_{\text{wait}}(i+1)\right] \tag{4}$$

is the average time wasted during the $n_{\max}$ attempts when the transmission fails.

$\Pr[n\text{succ} \mid l, \hat{s}, \hat{m}]$ is the probability of successful transmission at the nth attempt conditioned to $l, \hat{s}, \hat{m}$, and $\overline{\mathcal{D}}_{\text{wait}}(i)$ is the average waiting time before the i-th attempt. Their expressions are given by Equations (7) and (8) in [8]. The transmission time (T_{data}), ACK time (T_{ACK}) and average backoff time ($\overline{T}_{\text{bkoff}}$) are given by Eq. (1)–(3) in [8]. Finally, T_{SIFS} and T_{DIFS} are 802.11a parameters, and they can be found also in Table 2 in [8].

2.2. Energy consumption model

The selected energy model is our previous work of [9], which has been further validated via ad-hoc circuitry and specialised hardware [10] and, to the best of our knowledge, stands as the most accurate energy model for 802.11 devices published so far, because it accounts not only the energy consumed by the wireless card, but the consumption of the whole device. While classical models focused on the wireless interface solely, this one demonstrates empirically that the energy consumed by the device itself cannot be neglected as a device-dependent constant contribution. Conversely, devices incur an energy cost derived from the frame processing, which may impact the relationship that we want to evaluate in this paper.

The energy model is a multilinear model articulated into three main components:

$$\overline{P}(\tau_i, \lambda_i) = \rho_{\text{id}} + \underbrace{\sum_{i \in \{\text{tx}, \text{rx}\}} \rho_i \tau_i + \sum_{i \in \{\text{g}, \text{r}\}} \gamma_{\text{x}i} \lambda_i}_{\text{classical model}} \tag{5}$$

where the first two addends correspond to the classical model and the third is the contribution described in [9]. These components are the following:

- A platform-specific baseline power consumption that accounts for the energy consumed just by the fact of being powered on, but with no network activity. This component is commonly referred to as *idle consumption, ρ_{id}*.
- A component that accounts for the energy consumed in transmission, which linearly grows with the airtime percentage τ_{tx}, i.e., $\overline{P}_{\text{tx}}(\tau_{\text{tx}}) = \rho_{\text{tx}} \tau_{\text{tx}}$. The slope ρ_{tx} depends linearly on the radio transmission parameters MCS and TXP.

- A component that accounts for the energy consumed in reception, which linearly grows with the airtime percentage τ_{rx}, i.e., $\overline{P_{rx}}(\tau_{rx}) = \rho_{rx}\tau_{rx}$. The slope ρ_{rx} depends linearly on the radio transmission parameter MCS.
- A new component, called *generation cross-factor* or γ_{xg}, that accounts for a per-frame energy processing toll in transmission, which linearly grows with the traffic rate λ_g generated, i.e., $\overline{P_{xg}}(\lambda_g) = \gamma_{xg}\lambda_g$. The slope γ_{xg} depends on the computing characteristics of the device.
- A new component, called *reception cross-factor* or γ_{xr}, that accounts for a per-frame energy processing toll in reception, which linearly grows with the traffic rate λ_r received, i.e., $\overline{P_{xr}}(\lambda_r) = \gamma_{xr}\lambda_r$. Likewise, the slope γ_{xr} depends on the computing characteristics of the device.

Therefore, the average power consumed $\overline{P}$ is a function of five device-dependent parameters (ρ_i, γ_{xi}) and four traffic-dependent ones (τ_i, λ_i).

2.3. Energy efficiency analysis

Putting together both models, we are now in a position to build a joint goodput-energy model for 802.11a DCF. Let us consider the average durations (3) and (4). Based on their expressions, we multiply the idle time ($\overline{\mathcal{D}}_{wait}$, $\overline{T}_{bkoff}$, T_{SIFS}, T_{DIFS}) by ρ_{id}, the transmission time (T_{data}) by ρ_{tx}, and the reception time (T_{ACK}) by ρ_{rx}. The resulting expressions are the average energy consumed in a successful transmission $\mathcal{E}_{succ|l,\hat{s},\hat{m}}$ and the average energy wasted when a transmission fails $\mathcal{E}_{fail|l,\hat{s},\hat{m}}$:

$$
\begin{aligned}
\mathcal{E}_{succ|l,\hat{s},\hat{m}} = \sum_{n=1}^{n_{max}} \Pr[nsucc \mid l, \hat{s}, \hat{m}] \cdot \Bigg\{ \sum_{i=2}^{n_{max}} \Big[\rho_{id}\overline{T}_{bkoff}(i) \\
+ \rho_{tx}T_{data}(l, m_i) + \rho_{id}\overline{\mathcal{D}}_{wait}(i) \Big] \\
+ \rho_{id}\overline{T}_{bkoff}(1) + \rho_{tx}T_{data}(l, m_1) + \rho_{id}T_{SIFS} \\
+ \rho_{rx}T_{ACK}(m'_n) + \rho_{id}T_{DIFS} \Bigg\}
\end{aligned}
\tag{6}
$$

$$
\begin{aligned}
\mathcal{E}_{fail|l,\hat{m}} = \sum_{i=1}^{n_{max}} \Big[\rho_{id}\overline{T}_{bkoff}(i) \\
+ \rho_{tx}T_{data}(l, m_i) + \rho_{id}\overline{\mathcal{D}}_{wait}(i+1) \Big]
\end{aligned}
\tag{7}
$$

Then, by analogy with (2), the *expected energy consumed per frame transmitted*, $\mathbb{E}[\mathcal{E}_{data}]$, can be written as follows:

$$
\begin{aligned}
\mathbb{E}[\mathcal{E}_{data}] = \gamma_{xg} + \big(1 - \Pr[succ \mid l, \hat{s}, \hat{m}]\big) \cdot \mathcal{E}_{fail|l,\hat{s},\hat{m}} \\
+ \Pr[succ \mid l, \hat{s}, \hat{m}] \cdot \mathcal{E}_{succ|l,\hat{s},\hat{m}}
\end{aligned}
\tag{8}
$$

It is noteworthy that the receiving cross-factor does not appear in this expression because ACKs (acknowledgements) are processed in the network card exclusively, and thus its processing toll is negligible.

Finally, we define the *expected effective energy efficiency* μ as the ratio between the expected delivered data payload and the expected energy consumed per frame, which can be expressed in *bits per Joule* (bpJ):

$$
\mu(l, \hat{s}, \hat{m}) = \frac{\mathbb{E}[data]}{\mathbb{E}[\mathcal{E}_{data}]}
\tag{9}
$$

3. Numerical results

Building on the joint model presented in the previous section, here we explore the relationship between optimal goodput and energy efficiency in 802.11a. More specifically, our objective is to understand the behaviour of the energy efficiency of a single spatial stream as the MCS and TXP change following our model to meet the optimal goodput.
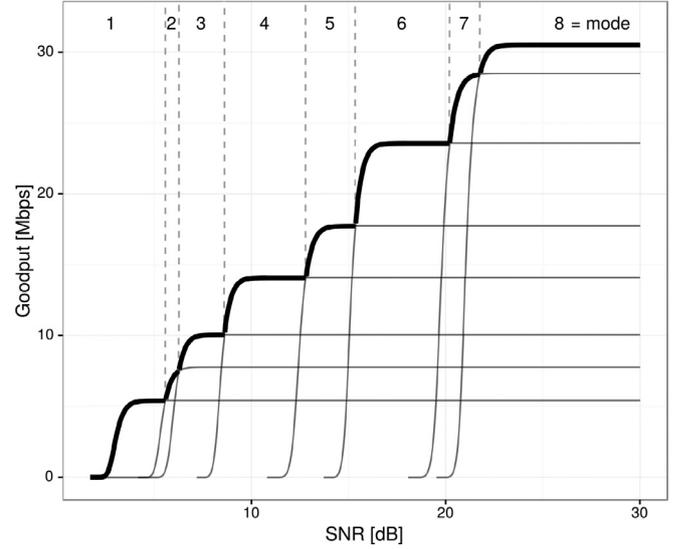

Fig. 1. Optimal goodput (bold envelope) as a function of SNR.

Table 1
Modes of the IEEE 802.11a PHY.

Mode Index	1	2	3	4	5	6	7	8
MCS (Mbps)	6	9	12	18	24	36	48	54

3.1. Optimal goodput

We note that the main goal of RA, generally, is to maximise the effective goodput that a station can achieve by varying the parameters of the interface. In terms of the model discussed in the previous section, a rate adaptation algorithm would aspire to fit the following curve:

$$
\max \mathcal{G}(l, \hat{s}, \hat{m})
\tag{10}
$$

We provide the numerical results for this goodput maximisation problem in Fig. 1, which are in good agreement with those obtained in [8]. For the sake of simplicity but without loss of generality we fix $l = 1500$ octets and $n_{max} = 7$ retries, and assume that the channel conditions and the transmission strategy are constant across retries ($\hat{s} = \{s_1, \ldots, s_1\}$ and $\hat{m} = \{m_1, \ldots, m_1\}$).

Fig. 1 illustrates which mode (see Table 1) is optimal in terms of goodput, given an SNR level. We next address the question of whether this optimisation is aligned with energy efficiency maximisation.

### 3.2. Extension of the energy parametrisation

The next step is to delve into the energy consumption of wireless devices. Serrano et al. [9] provides real measurements for five devices: three AP-like platforms (Linksys WRT54G, Raspberry Pi and Soekris net4826-48) and two hand-held devices (HTC Legend and Samsung Galaxy Note 10.1). Two of the four parameters needed are constant (ρ_{id}, γ_{xg}), and the other two (ρ_{tx}, ρ_{rx}) depend on the MCS and the TXP used. However, the characterisation done in [9] is performed for a subset of the MCS and TXP available, so we next detail how we extend the model to account for a larger set of operation parameters.

A detailed analysis of the numerical figures presented in [9] suggests that ρ_{rx} depends linearly on the MCS, and that ρ_{tx} depends linearly on the MCS and the TXP (in mW). Based on these observations, we define the following linear models:

$$
\rho_{tx} = \alpha_0 + \alpha_1 \cdot MCS + \alpha_2 \cdot TXP
\tag{11}
$$

Table 2
Linear Regressions.

Device	ρ_{tx} model estimates				ρ_{rx} model estimates		
	α_0 [W]	α_1 [W/Mbps]	α_2 [W/mW]	adj. r^2	β_0 [W]	β_1 [W/Mbps]	adj. r^2
HTC Legend	0.354(14)	0.0052(3)	0.021(3)	0.97	0.013(3)	0.00643(11)	> 0.99
Linksys WRT54G	0.540(12)	0.0028(2)	0.075(3)	0.98	0.14(2)	0.0130(7)	0.96
Raspberry Pi	0.478(19)	0.0008(4)	0.044(5)	0.88	$-0.0062(14)$	0.00146(5)	0.98
Galaxy Note 10.1	0.572(4)	0.0017(1)	0.0105(9)	0.98	0.0409(10)	0.00173(4)	0.99
Soekris net4826-48	0.17(3)	0.0170(6)	0.101(7)	0.99	0.010(8)	0.0237(3)	> 0.99
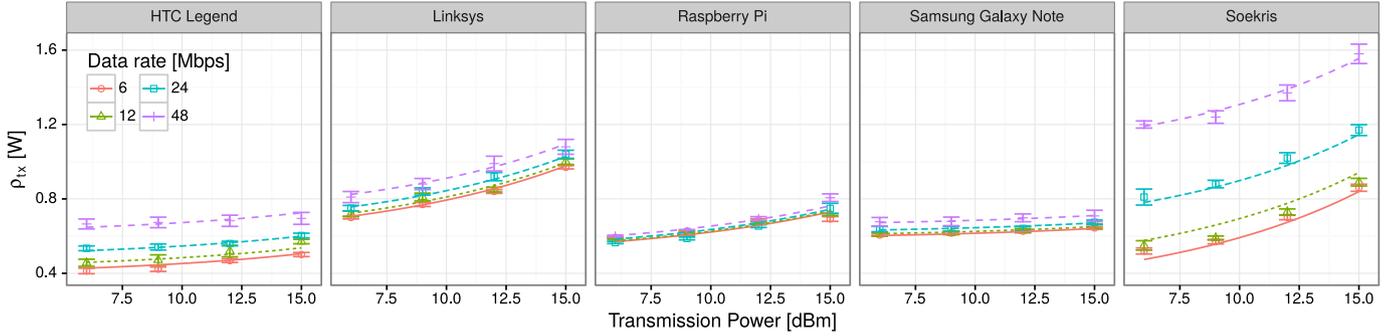
(a) ρ_{tx} fit as a function of MCS and TXP.
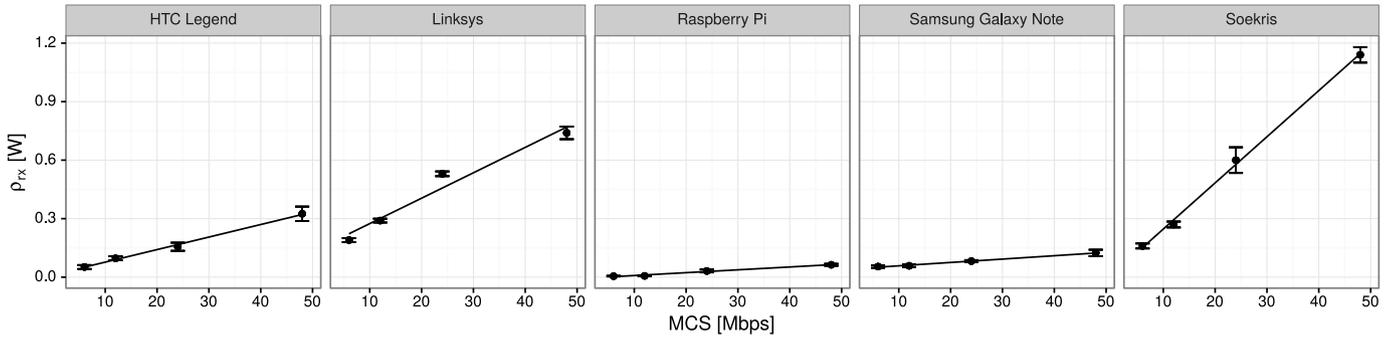
(b) ρ_{rx} fit as a function of MCS.

Fig. 2. Linear regressions.

$$\rho_{rx} = \beta_0 + \beta_1 \cdot \text{MCS} \qquad (12)$$

The models are fed with the data reported in [9], and the resulting fitting is illustrated in Fig. 2(a) and (b), while Table 2 collects the model estimates for each device (with errors between parentheses), as well as the adjusted r-squared. Since these linear models show a very good fit, they support the generation of synthetic data for the different MCS and TXP required.

3.3. Energy consumption

To compute the energy consumption using the above parametrisation, first we have to define the assumptions for the considered scenario. We assume for simplicity a device-to-device communication, with fixed and reciprocal channel conditions during a sufficient period of time (i.e., low or no mobility). As we have discussed before, our primary goal is to isolate MCS and TXP as variables of interest, but we must not forget that these are also reasonable assumptions in scenarios targeted by recent 802.11 standard developments (11ac, 11ad).

For instance, given channel state information from a receiver, the transmitter may decide to increase the TXP in order to increase the receiver's SNR (and thus the expected goodput), or to decrease it if the channel quality is high enough. Although the actual relationship between TXP and SNR depends on the specific channel model (e.g., distance, obstacles, noise), without loss of generality, we choose a noise floor of $N = -85$ dBm in an office scenario with a link distance of $d = 18$ m in order to explore numerically the whole range of SNR while using reasonable values of TXP. The ITU model for indoor attenuation [11] gives a path loss of $L \approx 85$ dBm. Then, we can use (8) to obtain the expected energy consumed per frame and MCS mode, with TXP being directly related to the SNR level.

The results are reported in Fig. 3. As the figure illustrates, consumption first falls abruptly as the TXP increases for all modes, which is caused when the SNR reaches a sharp threshold level such that the number of retransmissions changes from 6 to 0 (i.e., no frame is discarded). From this threshold on, the consumption increases with TXP because, although the number of retransmissions is 0, the wireless interface consumes more power. We note that the actual value of the TXP when the consumption drops depends on the specifics of the scenario considered, but the qualitative conclusions hold for a variety of scenarios.
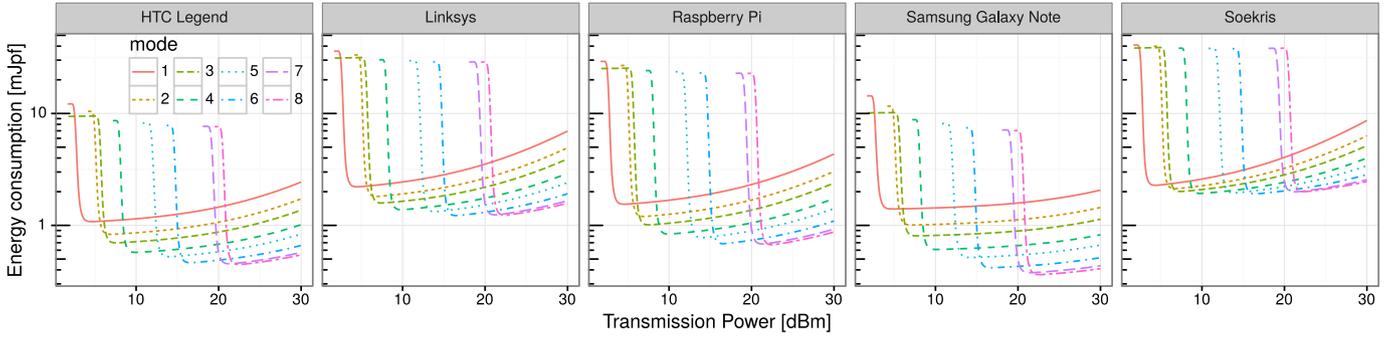
Fig. 3. Expected energy consumption per frame in millijoules per frame (mJpf) under fixed channel conditions.
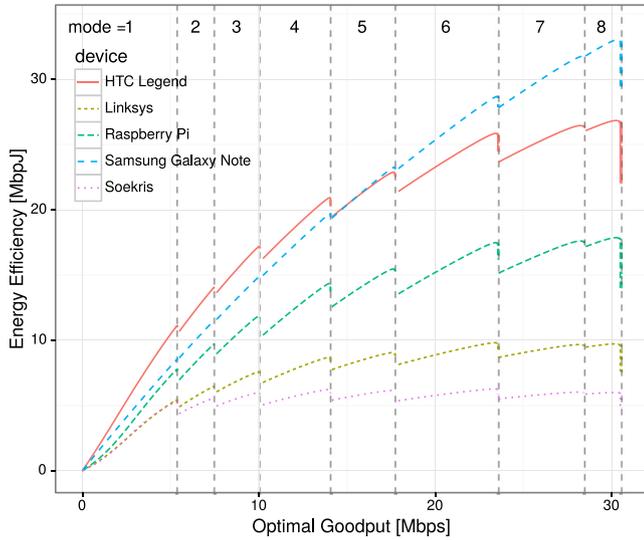

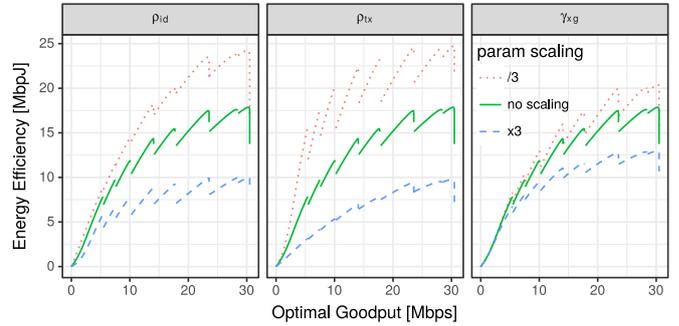**Fig. 4.** Energy efficiency vs. optimal goodput under fixed channel conditions.

3.4. Energy efficiency vs. optimal goodput

We can finally merge previous numerical analyses and confront energy efficiency, given by (9), and optimal goodput, given by (10), for all devices and under the aforementioned assumptions. To this aim, we plot in the same figure the energy efficiency for the configuration that maximises goodput given an SNR value vs. the obtained goodput, with the results being depicted in Fig. 4. We next discuss the main findings from the figure.

First of all, we can see that the energy efficiency grows sublinearly with the optimal goodput (the optimal goodput for each SNR value) in all cases. We may distinguish three different cases in terms of energy efficiency: high (Samsung Galaxy Note and HTC Legend), medium (Raspberry Pi) and low energy efficiency (Linksys and Soekris). Furthermore, for the case of the Soekris, we note that the "central modes" (namely, 4 and 5) are more efficient in their optimal region than the subsequent ones.

Another finding (more relevant perhaps) is that it becomes evident that increasing the goodput does not always improve the energy efficiency: there are more or less drastic leaps, depending on the device, between mode transitions. From the transmitter point of view, in the described scenario, this can be read as follows: we may increase the TXP to increase the SNR, but if the optimal goodput entails a mode transition, the energy efficiency may be affected.

As a conclusion, we have demonstrated that optimal goodput and energy efficiency do not go hand in hand, even in a single spa-

tial stream, in 802.11. There is a trade-off in some circumstances that current rate adaptation algorithms cannot take into account, as they are oblivious to the energy consumption characteristic of the device.

3.5. Sensitivity to energy parameter scaling

We next explore how the different energy parameters affect the energy efficiency vs. optimal goodput relationship. For this purpose, we selected the Raspberry Pi curve from Fig. 4 (results are analogous with the other devices) and we scale up and down, one at a time, the four energy parameters ρ_{id}, ρ_{tx}, ρ_{rx}, and γ_{xg}. The scaling up and down is done by multiplying and dividing by 3, respectively, the numerical value of the considered parameter. One of the first results is that the impact of ρ_{rx} is negligible, a result somehow expected as the cost of receiving the ACK is practically zero. From this point on, we do not consider further this parameter.

We show in Fig. 5(a) the overall effect of this parameter scaling. The solid line represents the base case with no scaling (same

(a) Overall effect.

(b) Impact on mode transitions.

Fig. 5. Impact of energy parameter scaling on the energy efficiency.

curve as in Fig. 4), and in dashed and dotted lines the corresponding parameter was multiplied or divided by a factor of 3, respectively. As expected, larger parameters contribute to lower the overall energy efficiency. However, the impact on the energy efficiency drops between mode transitions is far from being obvious, as in some cases transitions are more subtle while in others they become more abrupt.

To delve into these transitions, we illustrate in Fig. 5(b) the "drop" in energy efficiency when changing between modes. As it can be seen, the cross-factor γ_{xg} is the less sensitive parameter of the three, because its overall effect is limited and, more importantly, it scales all the leaps between mode transitions homogeneously. This means that a higher or lower cross-factor, which resides almost entirely in the device and not in the wireless card, does not alter the energy efficiency vs. optimal goodput relationship (note that this parameter results in a constant term in (8)). Thus, the cross-factor is not relevant from the RA-TPC point of view, and energy-aware RA-TPC algorithms can be implemented by leveraging energy parameters local to the wireless card.

On the other hand, ρ_{id} and ρ_{tx} have a larger overall effect, plus an inhomogeneous and, in general, opposite impact on mode transitions. While a larger ρ_{id} contributes to larger leaps, for the case of ρ_{tx}, the larger energy efficiency drops occur with smaller values of that parameter. Still, the reason behind this behaviour is the same for both cases: the wireless card spends more time in idle (and less time transmitting) when a transition to the next mode occurs, which has a higher data rate.

This effect is also evident if we compare the Samsung Galaxy Note and the HTC Legend curves in Fig. 4. Both devices have ρ_{id} and ρ_{tx} in the same order of magnitude, but the HTC Legend has a larger ρ_{id} and a smaller ρ_{tx}. The combined outcome is a more dramatic sub-linear behaviour and an increased energy efficiency drop between mode transitions.

3.6. Discussion

We have seen that the energy efficiency vs. optimal goodput relationship shows a signature "sawtooth" pattern when RA and TPC are considered for a single 802.11 spatial stream. This sawtooth shape presents a growing trend in the central part of each mode, but there are energy efficiency drops between mode transitions, which conceal a trade-off.

Parameter scaling has diverse effects on the final consumption signature, but overall, the qualitative behaviour (i.e., the shape) remains the same. The cross-factor produces a homogeneous scaling of the sawtooth. Thus, a first conclusion is that the trade-off depends on the energy parameters local to the wireless card, which means that a properly designed energy-aware RA-TPC algorithm can be device-agnostic.

Moreover, an energy-aware RA-TCP algorithm may also be card-agnostic. This is because the inefficiencies are always constrained at mode transitions, which are exactly the points at which RA-TPC algorithms take decisions. Therefore, there is no need of knowing the exact energy parametrisation, nor the instantaneous power consumption of the wireless card, in order to take energy-efficient decisions.

An RA-TPC algorithm moves along the sawtooth shapes of Fig. 4 in two directions, namely, "up" (towards higher throughput) and "down" (towards lower throughput). In this way, an algorithm requires different policies to make a decision: (*i*) the *upwards* policy, in which mode transitions take place by increasing MCS and TXP (to achieve more goodput), and (*ii*) the *downwards* policy, in which mode transitions take place by decreasing MCS and TXP.

(i) In the *upwards* direction, a sensitive policy would be to remain in the left side of the leaps, to prevent falling into the
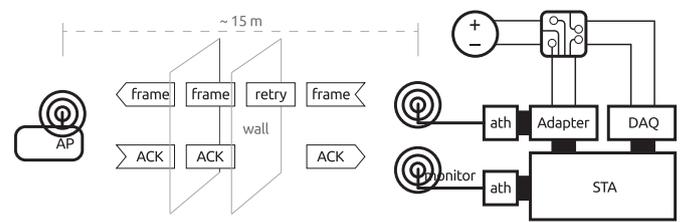

Fig. 6. Experimental setup.

efficiency gaps, until the link is good enough to move to a higher MCS with at least the same efficiency. An heuristic for the *upwards* policy may be the following: whenever an algorithm chooses a higher MCS, it may hold the decision for some time and, if it persists, then trigger the MCS change (however, if this delay is too long, the algorithm will incur in inefficiencies, too).

(ii) In the *downwards* direction, a sensitive policy would be to try to reach the left side of the leaps as soon as possible. However, it should be noted that this *downwards* policy is much more challenging, as it implies predicting quality drops to trigger early MCS/TXP changes.

In summary, our results suggest that one of the key points of an energy-aware RA-TPC algorithm is the management of mode transitions. A good algorithm should be *conservative* at mode transitions, in the sense that it should prefer a lower MCS and TXP until a higher MCS can be guaranteed.

4. Experimental validation

This section is devoted to experimentally validate the results from the numerical analysis and, therefore, the resulting conclusions. To this aim, we describe our experimental setup and the validation procedure, first specifying the methodology and then the results achieved.

4.1. Experimental setup

We deployed the testbed illustrated in Fig. 6, which consists of a station (STA) transmitting evenly-spaced maximum-sized UDP packets to an access point (AP). The AP is an x86-based Alix6f2 board with a Mini PCI Qualcomm Atheros AR9220 wireless network adapter, running Voyage Linux with kernel version 3.16.7 and the `ath9k` driver. The STA is a desktop PC with a Mini PCI Express Qualcomm Atheros QCA9880 wireless network adapter, running Fedora Linux 23 with kernel version 4.2.5 and the `ath10k` driver. We also installed at the STA a Mini PCI Qualcomm Atheros AR9220 wireless network adapter to monitor the wireless channel.

The QCA9880 card is connected to the PC through a *x1 PCI Express to Mini PCI Express* adapter from Amfeltec. This adapter connects the PCI bus' data channels to the host and provides an ATX port so that the wireless card can be supplied by an external power source. The power supply is a Keithley 2304A DC Power Supply, and it powers the wireless card through an *ad-hoc* measurement circuit that extracts the voltage and converts the current with a high-precision sensing resistor and amplifier. These signals are measured using a National Instruments PCI-6289 multifunction data acquisition (DAQ) device, which is also connected to the STA. Thanks to this configuration, the STA can simultaneously measure the instant power consumed by the QCA9880 card,[1] and the goodput achieved.

---

[1] Following the discussion on Section 3.5 the device's cross-factor is not involved in the trade-off, thus we will expect to reproduce it by measuring the wireless interface alone.
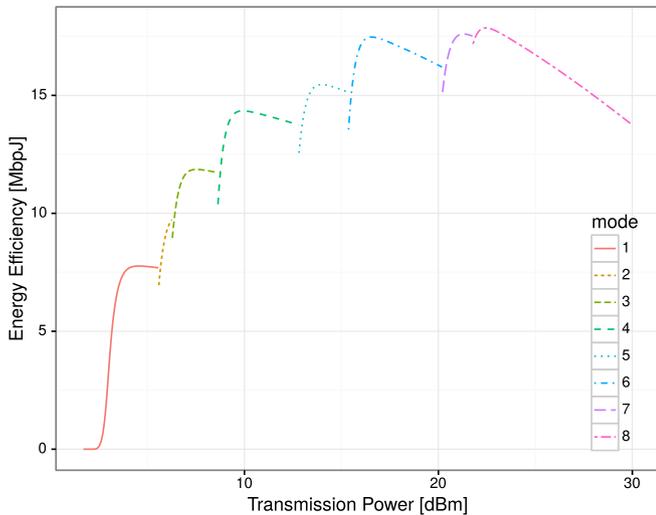
Fig. 7. Energy Efficiency vs. Transmission Power under fixed channel conditions for the Raspberry Pi case.

As Fig. 6 illustrates, the STA is located in an office space and the AP is placed in a laboratory 15 m away, and transmitted frames have to transverse two thin brick walls. The wireless card uses only one antenna and a practically-empty channel in the 5-GHz band. Throughout the experiments, the STA is constantly back-logged with data to send to the AP, and measures the throughput obtained by counting the number of received acknowledgements (ACKs).

4.2. Methodology and results

In order to validate our results, our aim is to replicate the qualitative behaviour of Fig. 4, in which there are energy efficiency "drops" as the optimal goodput increases. However, in our experimental setting, channel conditions are not controllable, which introduces a notable variability in the results as it affects both the *x*-axis (goodput) and the *y*-axis (energy efficiency). To reduce the impact of this variability, we decided to change the variable in the *x*-axis from the optimal goodput to the transmission power —a variable that is directly configured in the wireless card—. In this way, the qualitative behaviour to replicate is the one illustrated in Fig. 7, where we can still identify the performance "drops" causing the loss in energy efficiency.

Building on Fig. 7, we perform a sweep through all available combinations of MCS (see Table 1) and TXP.[2] Furthermore, we also tested two different configurations of the AP's TXP at different times of the day, to confirm that this qualitative behaviour is still present under different channel conditions. For each configuration, we performed 2-s experiments in which we measure the total bytes successfully sent and the energy consumed by the QCA9880 card with sub-microsecond precision, and we compute the energy efficiency achieved for each experiment.

The results are shown in Fig. 8. Each graph corresponds to a different TXP value configured at the AP, and depicts a single run (note that we performed several runs throughout the day and found no major qualitative differences across them). Each line type represents the STA's mode that achieved the highest goodput for each TXP interval, therefore in some cases low modes do not appear because a higher mode achieved a higher goodput. Despite the inherent experimental difficulties, namely, the low granularity
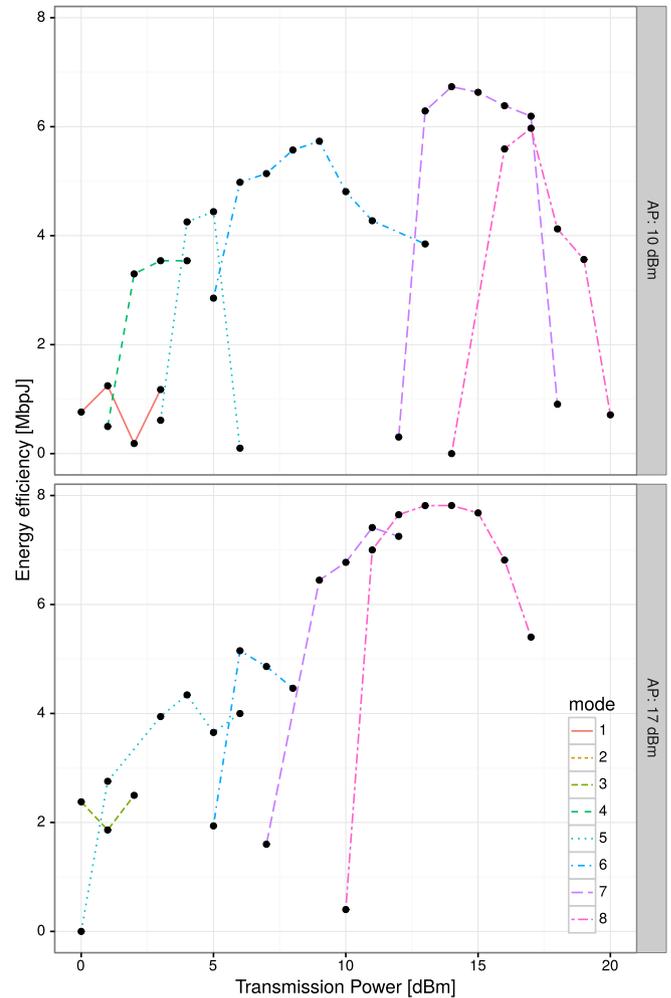
---

[2] The model explores a range between 0 and 30 dBm to get the big picture, but this particular wireless card only allows us to sweep from 0 to 20 dBm.

Fig. 8. Experimental study of Fig. 7 for two AP configurations.

of 1-dBm steps and the random variability of the channel, the experimental results validate the analytical ones, as the qualitative behaviour of both graphs follows the one illustrated in Fig. 7. In particular, the performance "drops" of each dominant mode can be clearly observed (especially for the 36, 48 and 54 Mbps MCSs) despite the variability in the results.

5. On the performance of RA-TPC algorithms

So far, we have demonstrated through numerical analysis, and validated experimentally, the existence of a trade-off between two competing performance figures, namely, goodput and energy efficiency. This issue arises even for a single spatial stream in absence of interference. Furthermore, we have discussed in Section 3.6 some ideas about the kind of mechanisms that energy-aware RA-TPC algorithms may incorporate, to leverage the behaviour that we have identified in our analysis in these so-called mode transitions. *In nuce*, the algorithms should be *conservative* during these transitions.

During that discussion, we neglected the challenge of estimating channel conditions. In practice, any RA-TPC algorithm has imperfect channel knowledge, and therefore will adapt to changing conditions in a suboptimal way. In this section, we will analyse and compare the performance of several representative existing RA algorithms, which also incorporate TPC, to confirm if the *conservativeness* in such decisions may have a positive impact on the achieved performance.

5.1. Considered RA-TPC algorithms

If we take a look at the actual operation of WiFi networks, the Minstrel algorithm [12], which was integrated into the Linux kernel, has become the *de facto* standard due to its relatively good performance and robustness. However, Minstrel does not consider TPC and, in consequence, there is no TPC in today's WiFi deployments. Moreover, despite some promising proposals have been presented in the literature, there are very few of them implemented, although there are some ongoing efforts such as the work by the authors of *Minstrel-Piano* [13], who are pushing to release an enhanced version of the latter for the Linux kernel with the goal of promoting it upstream.[3]

As stated before, RA is a very prolific research line in the literature, but the main corpus is dedicated to the MCS adjustment without taking into account the TXP [4]. There is some work considering TPC, but the motivation is typically the performance degradation due to network densification, and the aim is interference mitigation [14] and not energy efficiency. Given that we are interested in assessing RA implementations with TPC support, we consider only open-source algorithms that can be tested using the NS3 Network Simulator. After a thorough analysis of the literature, we consider the following set of algorithms:

- *Power-controlled Auto Rate Fallback* (PARF) [15], which is based on *Auto Rate Fallback* (ARF) [16], one of the earliest RA schemes for 802.11. ARF rate adaptation is based on the frame loss ratio. It tunes the MCS in a very straightforward and intuitive way. The procedure starts with the lowest possible MCS. Then, if either a timer expires or the number of consecutive successful transmissions reach a threshold, the MCS is increased and the timer is reset. The MCS is decreased if either the first transmission at a new rate fails or two consecutive transmissions fail. PARF builds on ARF and tries to reduce the TXP if there is no loss until a minimum threshold is reached or until transmissions start to fail. If transmission fails persist, the TXP is increased.

- *Minstrel-Piano* (MP) [13] is based on Minstrel [12]. Minstrel performs per-frame rate adaptation based on throughput. It randomly probes the MCS space and computes an exponential weighted moving average (EWMA) on the transmission probability for each rate, in order to keep a long-term history of the channel state. As the previous algorithm, MP adds TPC without interfering with the normal operation of Minstrel. It incorporates to the TPC the same concepts and techniques than Minstrel uses for the MCS adjustment, i.e., it tries to learn the impact of the TXP on the achieved throughput.

- *Robust Rate and Power Adaptation Algorithm* (RRPAA) and *Power, Rate and Carrier-Sense Control* (PRCS) [14], which are based on *Robust Rate Adaptation Algorithm* (RRAA) [17]. RRAA consists of two functional blocks, namely, rate adaptation and collisions elimination. It performs rate adaptation based on loss ratio estimation over short windows, and reduces collisions with a RTS-based strategy. The procedure starts at the maximum MCS. The loss ratio for each window of transmissions is available for rate adjustment in the next window. There are two thresholds involved in this adjustment: if the loss ratio is below both of them, the MCS is increased; if it is above, the MCS is decreased; and if it is in between, the MCS remains unchanged. RRPAA and PRCS build on this and try to use the lowest possible TXP without degrading the throughput. For this purpose, they firstly find the best MCS at the maximum TXP and, from there, they jointly adjust the MCS and TXP for each window based on a similar
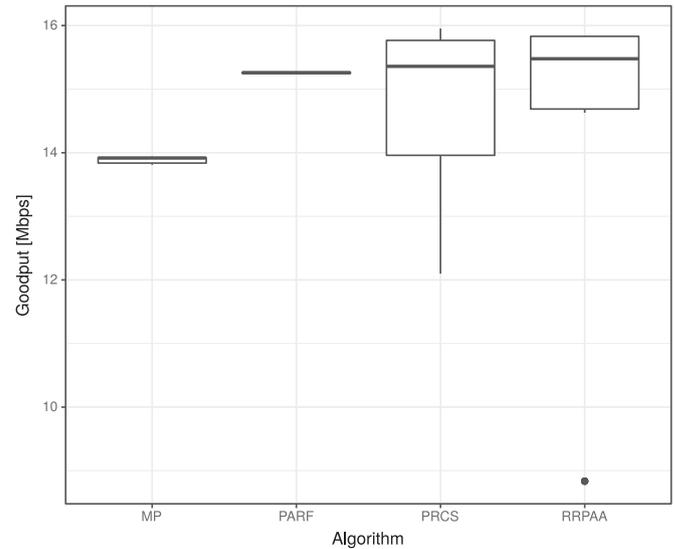

Fig. 9. Simulation scenario.

Fig. 10. Goodput achieved per simulated algorithm.

thresholding system. RRPAA and PRCS are very similar and only differ in implementation details.

Based on their behaviour, these algorithms can be classified into three distinct classes. First of all, MP is the most aggressive technique, given that it constantly samples the whole MCS/TXP space searching for the best possible configuration. On the opposite end, RRPAA and PRCS do not sample the whole MCS/TXP space. Instead, they are based on a windowed estimation of the loss ratio, which makes the MCS/TXP transitions much lazier. Finally, PARF falls in between, as it changes the MCS/TXP to the next available proactively if a number of transmissions are successful, but it falls back to the previous one if the new one fails. In practice, this may result in some instability during transitions.

5.2. Scenario

This evaluation is publicly available,[4] and builds upon the code provided by Richart et al. in [14].[5] We assessed the proposed algorithms in the toy scenario depicted in Fig. 9. It consists of a single access point (AP) and a single mobile node connected to this AP configured with the 802.11a PHY. The mobile node at the farthest distance at which is able to communicate at the lowest possible rate (6 Mbps) and highest TXP (17 dBm), and then it moves at constant speed towards the AP. The simulation stops when the node is directly in front of the AP and it is able to communicate at the highest possible rate (54 Mbps) and lowest TXP (0 dBm). This way, we sweep through all mode transitions available.

---

[3] https://github.com/thuehn/Minstrel-Blues.

[4] https://github.com/Enchufa2/ns-3-dev-git.
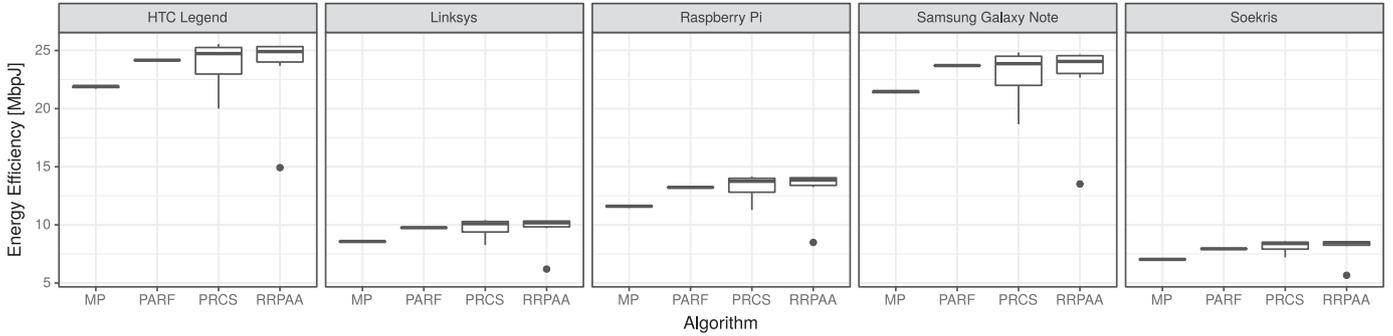[5] https://github.com/mrichart/ns-3-dev-git.

Fig. 11. Energy efficiency achieved per simulated algorithm and device.

For the whole simulation, the AP tries to constantly saturate the channel by sending full-size UDP packets to the node. Every transmission attempt is monitored, as well as every successful transmission. The first part allows us to compute the transmission time, while the latter allows us to compute the reception time (of the ACKs) and the goodput achieved.

The simulation model assembles the power model (5) with the parametrisation previously made (see Table 2) for all the devices considered in Section 3: HTC Legend, Linksys WRT54G, Raspberry Pi, Samsung Galaxy Note 10.1 and Soekris net4826-48. Thus, the total energy consumed is computed for all the devices and each run using the computed transmission time, reception time and idle time. The beacons are ignored and considered as idle time.

We set up one simulation for each algorithm (PARF, MP, PRCS, RRPAA) with a fixed seed, and perform 10 independent runs for each simulation. We use boxplots for the results unless otherwise mentioned.

5.3. Results

We first analyse the goodput achieved per each algorithm, which are depicted in Fig. 10. The median of the average goodput across several runs for RRPAA is the highest, followed by PRCS, PARF and MP. PRCS and RRPAA, which are very similar mechanisms, show a higher variability across replications compared to PARF and MP, which have little dispersion.

Fig. 11 shows the energy efficiency achieved per algorithm, computed for all the devices presented in Section 3. As expected, the numerical values of the energy efficiency achieved are different across devices, but the relative performance is essentially the same, as in the previous case. Indeed, the efficiency follows the pattern seen in Fig. 10: RRPAA results the most energy efficient in our scenario, followed by PRCS, PARF and MP. PRCS and RRPAA exhibit the same variability across replications as in the case of goodput, which is particularly notable for the most efficient devices, i.e., the HTC Legend and the Samsung Galaxy Note.

5.4. Discussion

In order to shed some light into the reasons behind the differences in performance, Fig. 12(a) and (b) show the behaviour of each algorithm throughout the simulation time for one run, showing the evolution of the MCS and TXP chosen by each algorithm, respectively. Here, we can clearly differentiate that there are two kinds of behaviour: while MP and PARF are constantly sampling other MCSs and TXPs, PRCS and RRPAA are much more conservative in that sense, and tend to keep the same configuration for longer periods of time.

MP randomly explores the whole MCS/TXP space above a basic *guaranteed* value, and this is the explanation for the apparently uniformly greyed zone. Also, this aggressive approach is clearly a

disadvantage in the considered toy scenario (deterministic walk, one-to-one, no obstacles), and this is why the achieved goodput in Fig. 10 is slightly smaller than the one achieved by the others. PARF, on its part, only explores the immediately higher MCS/TXP, which leads to a higher goodput and efficiency.

On the other hand, PRCS and RRPAA sampling is much more sparse in time. As a consequence, Fig. 12(a) and (b) are much more different across replications, leading to the high variability shown in Fig. 10 compared to MP and PARF.

In terms of TXP, all the algorithms exhibit a similar *aggressiveness*, in the sense that they use a high TXP value in general. Indeed, as Fig. 12(b) shows, the TXP is the highest possible until the very end of the simulation, when the STA is very close to the AP. This is the cause for the high correlation between Figs. 10 and 11.

A noteworthy characteristic of PRCS and RRPAA is that, in general, they *delay* the MCS change decision, as depicted in Fig. 12(a). Most of the times, they do not even use the whole space of MCS available, unlike MP and PARF. Because of this, they tend to achieve the best goodput and energy efficiency.

5.5. Conservativeness at mode transitions

Building on the concept of *conservativeness* developed in Section 3.6 (i.e., the tendency to select a lower MCS/TXP in the transition regions), we explore whether there is any correlation of with the energy efficiency achieved by a certain algorithm and this tendency. For that purpose, we first define a proper metric.

In the first place, we define the *normalised average MCS* as the area under the curve in Fig. 12(a) normalised by the total simulation time and the maximum MCS:

$$\widehat{\text{MCS}} = \frac{1}{\max(\text{MCS}) \cdot t_{\text{sim}}} \int_0^{t_{\text{sim}}} \text{MCS}(t) dt \qquad (13)$$

where t_{sim} is the simulation time and $\max(\text{MCS})$ is 54 Mbps in our case. The same concept can be applied to the TXP:

$$\widehat{\text{TXP}} = \frac{1}{\max(\text{TXP}) \cdot t_{\text{sim}}} \int_0^{t_{\text{sim}}} \text{TXP}(t) dt \qquad (14)$$

where $\max(\text{TXP})$ is 17 dBm in our case. Both $\widehat{\text{MCS}}$ and $\widehat{\text{TXP}}$ are unitless scores between 0 and 1, and lower values mean a more conservative algorithm. Therefore, we can define a *Conservativeness Index* (CI) as the inverse of the product of both scores:

$$\text{CI} = \frac{1}{\widehat{\text{MCS}} \cdot \widehat{\text{TXP}}} \qquad (15)$$

where CI > 1.[6]

---

[6] It must be taken into account that the CI is not suitable for comparing *any* algorithm. For instance, in an extreme case, an "algorithm" could select 6 Mbps and 0 dBm always, resulting in a very low CI, but a very bad performance at the same time. The CI should only be used for comparing similarly performant algorithms, as it is the case in our study given the results shown in Figs. 10 and 11.
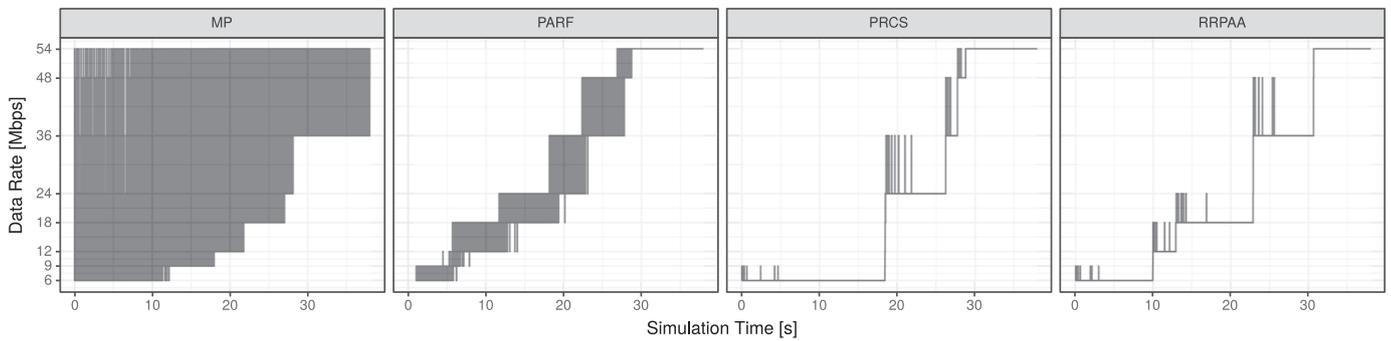
(a) MCS vs. simulation time.
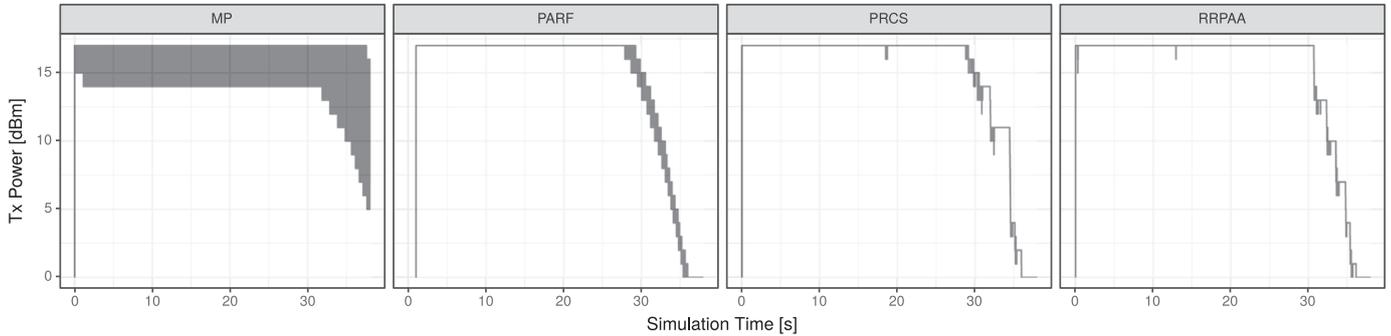
(b) TXP vs. simulation time.

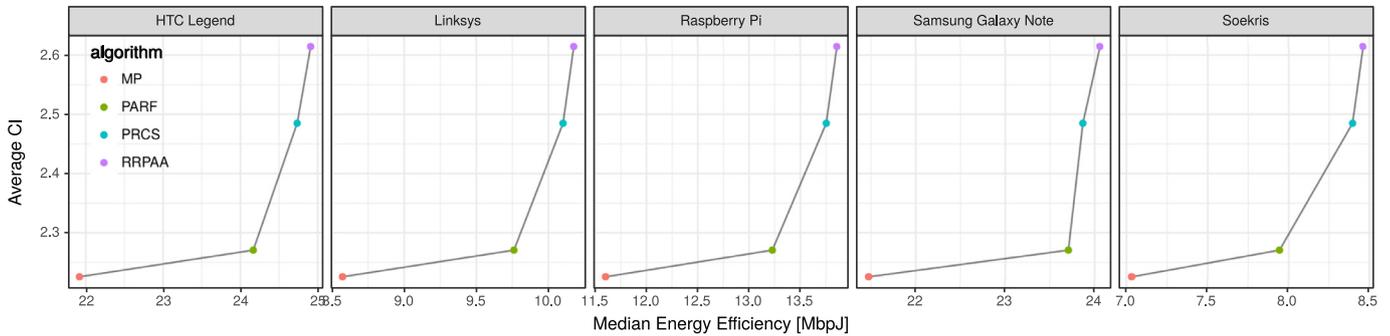**Fig. 12.** MCS and TXP evolution per algorithm for a selected run.

Fig. 13. Relationship between Conservativeness Index (tendency to select lower MCS and TXP) and energy effienciency per simulated device.

We computed the CI for each device and run, and the final results are depicted in Fig. 13 as the average CI across different runs vs. the median energy efficiency in Fig. 11 (note that the dots have been connected by straight lines to facilitate the visualisation).

The results in Fig. 13 show a positive non-linear relationship between the CI of an algorithm and the energy efficiency achieved for all the devices considered. MP is the algorithm with the lowest CI, which is in consonance with its aggressiveness (i.e., frequent jumps between MCS/TXP values, as shown in Fig. 12(a) and (b)), and the goodput achieved was also the lowest, as depicted in Fig. 10. On the other hand, PARF, PRCS and RRPAA achieved a similar performance in terms of goodput, but the ones with the most conservative behaviour (PRCS and RRPAA, as it can be seen in Fig. 12(a) and (b)) also achieve both the highest CI and energy efficiency.

This result evidences that the performance gaps uncovered by Fig. 4 under optimal conditions have also an impact in real-world RA-TPC algorithms. Therefore, we confirm that this issue must be taken into account in the design of more energy-efficient rate and transmission power control algorithms.

6. Conclusions

In this paper, we have revisited 802.11 rate adaptation and transmission power control by taking energy consumption into account. While some previous studies pointed out that MIMO rate adaptation is not energy efficient, we have demonstrated through numerical analysis that, even for single spatial streams without interfering traffic, energy consumption and throughput performance are different optimisation objectives. Furthermore, we have validated our results via experimentation.

Our findings show that this trade-off emerges at certain "mode transitions" when maximising the goodput, suggesting that small goodput degradations may lead to energy efficiency gains. For instance, a station at the edge of a mode transition may decide to reduce the transmission power a little in order to downgrade the modulation coding scheme. Or an opportunity to achieve a better goodput by increasing the transmission power and modulation coding scheme could be delayed if the expected gain is small.

We have assessed the performance of four state-of-the-art schemes through simulation, and we have demonstrated that cer-

tain conservativeness at mode transitions can make a difference for properly designed energy-aware rate adaptation with transmission power control algorithms.

Acknowledgements

References

[1] P. Serrano, A. de la Oliva, P. Patras, V. Mancuso, A. Banchs, Greening wireless communications: status and future directions, Comput. Commun. 35 (14) (2012) 1651–1661. http://dx.doi.org/10.1016/j.comcom.2012.06.011. Special issue: Wireless Green Communications and Networking.

[2] S. Eryigit, G. Gur, S. Bayhan, T. Tugcu, Energy efficiency is a subtle concept: fundamental trade-offs for cognitive radio networks, Commun. Mag., IEEE 52 (7) (2014) 30–36, doi:10.1109/MCOM.2014.6852080.

[3] A. Garcia-Saavedra, P. Serrano, A. Banchs, M. Hollick, Balancing energy efficiency and throughput fairness in IEEE 802.11 WLANs, Pervasive Mob. Comput. 8 (5) (2012) 631–645, doi:10.1016/j.pmcj.2012.03.006.

[4] S. Biaz, S. Wu, Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison, in: 2008 IEEE Symposium on Computers and Communications, 2008, pp. 130–136, doi:10.1109/ISCC.2008.4625680.

[5] K. Huang, K. Duffy, D. Malone, H-RCA: 802.11 collision-aware rate control, Networking, IEEE/ACM Trans. 21 (4) (2013) 1021–1034, doi:10.1109/TNET.2012.2216891.

[6] C.-Y. Li, C. Peng, S. Lu, X. Wang, Energy-based rate adaptation for 802.11n, in: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking. Mobicom '12, ACM, New York, NY, USA, 2012, pp. 341–352, doi:10.1145/2348543.2348585.

[7] M.O. Khan, V. Dave, Y.-C. Chen, O. Jensen, L. Qiu, A. Bhartia, S. Rallapalli, Model-driven energy-aware rate adaptation, in: Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing. MobiHoc '13, ACM, New York, NY, USA, 2013, pp. 217–226, doi:10.1145/2491288.2491300.

[8] D. Qiao, S. Choi, K. Shin, Goodput analysis and link adaptation for IEEE 802.11a wireless LANs, Mob. Comput., IEEE Trans. 1 (4) (2002) 278–292, doi:10.1109/TMC.2002.1175541.

[9] P. Serrano, A. Garcia-Saavedra, G. Bianchi, A. Banchs, A. Azcorra, Per-Frame energy consumption in 802.11 devices and its implication on modeling and design, IEEE/ACM Trans. Networking PP (99) (2014), doi:10.1109/TNET.2014.2322262. 1–1.

[10] I. Ucar, A. Azcorra, Deseeding energy consumption of network stacks, in: Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2015 IEEE 1st International Forum on, 2015, pp. 7–16, doi:10.1109/RTSI.2015.7325085.

[11] ITU-R, Recommendation P.1238, International Telecommunication Union, July, 2015.

[12] A. Mcgregor, D. Smithies, Rate adaptation for 802.11 wireless networks: minstrel, 2017.

[13] T. Huehn, C. Sengul, Practical power and rate control for WiFi, in: 2012 21st International Conference on Computer Communications and Networks (ICCCN), 2012, pp. 1–7, doi:10.1109/ICCCN.2012.6289295.

[14] M. Richart, J. Visca, J. Baliosian, Rate, power and carrier-sense threshold coordinated management for high-density IEEE 802.11 networks, in: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015, pp. 139–146, doi:10.1109/INM.2015.7140286.

[15] A. Akella, G. Judd, S. Seshan, P. Steenkiste, Self-management in chaotic wireless deployments, in: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking. MobiCom '05, ACM, New York, NY, USA, 2005, pp. 185–199, doi:10.1145/1080829.1080849.

[16] A. Kamerman, L. Monteban, Wavelan®-ii: a high-performance wireless LAN for the unlicensed band, Bell Labs Tech. J. 2 (3) (1997) 118–133.

[17] S.H.Y. Wong, H. Yang, S. Lu, V. Bharghavan, Robust rate adaptation for 802.11 wireless networks, in: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking. MobiCom '06, ACM, New York, NY, USA, 2006, pp. 146–157, doi:10.1145/1161089.1161107.