On the feasibility of power control in current IEEE 802.11 devices

Fehmi Ben Abdesslem, Luigi Iannone, Marcelo Dias de Amorim,
Konstantin Kabassanov, and Serge Fdida
LIP6/CNRS – University of Paris VI
8, rue du Capitaine Scott – 75015 – Paris – France
{fehmi,iannone,amorim,kabassan,sf}@rp.lip6.fr

Abstract

Recent research in wireless communications has achieved important results by exploring more and more sophisticated solutions involving power control. Cross-layer design and topology control are the main examples. Although much has been done in the theoretical domain, there is still a large gap between theory and practice. In this paper, we investigate whether current IEEE 802.11 devices are able to comply with cross-layer and topology control requirements. Our study and associated measurement results reveal that many novel power control solutions cannot be efficiently implemented over existing IEEE 802.11 cards.

1. Introduction

Wireless communications are changing the way people get connected, in both space and time. One particular technology has been responsible for the explosion of the wireless popularity, namely IEEE 802.11 [2]. This standard, widely deployed in both experimental and commercial products, has become the *de facto* wireless technology for local area networks. Extent research has been (and is still being) done considering the IEEE 802.11 as the underlying communication technology.

A particular technique that is gaining importance in the design of communication protocols for self-organizing networks is *cross-layer* design. When such an approach involves the lower layers of the protocol stack, one of the key components is *power control* [6, 3, 7]. Furthermore, power control is inherently related to topology control mechanisms and has been used in different network families, from wireless sensors, to ad hoc and wireless mesh networks [8, 9].

Although power control has been extensively studied in theory, the implementation of enhanced algorithms is still uncommon. The main reason is that, in general, such improvements impose deep changes in current standards, which yet if not perfect, as pointed out by several works, are consolidated and widely used. We believe that, for a solution to succeed, it is fundamental that it might be easily implementated over existing hardware and software systems. This paper addresses exactly this topic, *i.e.*, the investigation of whether new power control approaches (*e.g.*, algorithms, protocols, techniques) can be implemented using today's sytems and off-the-shelf wireless cards.

We focus on a number of issues related to the implementation of power control in IEEE 802.11. We evaluate a number of commercial IEEE 802.11 cards and their associated drivers. The idea is to experiment existing solutions with the goal of identifying the real usability of the wireless equipments in self-organizing networks. We show how some of the difficulties come from cross-layer solutions breaking the traditional layering of the protocol stack, but also how sometimes they come from the driver or the operating system on which the proposal relies.

The remainder of this paper is organized as follows. In Section 2, we present some examples of algorithms that need wireless devices with enhanced power control features. In Section 3, we address some practical issues in order to perform correct tests. Then, in Section 4, we present our measurements as well as our observations and analysis.

2. Topology control and cross-layer design

At the lower layers of the protocol stack, new forms of communication in the wireless medium have to be able to perform efficient dynamic power control. Indeed, in order to improve global performance in terms of energy consumption, throughput, latency, each node has to adapt its behavior to the channel conditions as well as improving the cooperation with other nodes.

Topology control systems aim at finding the optimal set of neighbors by shrinking the transmission power. Proposed solutions range from quasi-static common power levels to more dynamic and node-dependent approaches. For instance, Ramanathan and Rosales-Hain propose a static power control algorithm [8] where, once the optimal coverage has been computed, the power level is fixed until a physical topology change occurs. Such a kind of approach, where power level varies at a time scale larger than the one of routing updates, can be implemented using wireless cards supporting power control in its basic form, *i.e.*, the same power level for all transmitted packets.

Solutions based on such steady power control can also be implemented on real platforms. Kawadia and Kumar analyze the effects of power control in wireless networks and propose a set of protocols based on the COMmon POWer (COMPOW) approach [4]. COMPOW aims at finding the common power, for all nodes, that improves the aggregate network throughput. COMPOW has been implemented using Cisco Aironet 350 cards. Nevertheless, the authors complain about the high latency of power change due to the need of hardware reset. In order to reduce the delay introduced by power switching operation, they propose a quite complex architecture based on several routing daemons, one for each possible power level. They propose as well a modified scheduling policy in the kernel and some hacks on the card driver.

Some routing control algorithms, like the one proposed by Subbarao [9], are difficult to implement since they require a change in the transmitting power level on a per-packet basis, based on some feedback. Because of this, two back-to-back packets sent to the same next-hop may be transmitted with different power levels. Similarly, if two consecutive packets are addressed to different neighbors, it is likely that the transmitting power changes, since there is a non negligible probability that different nodes respond with different feedbacks. Such a dynamic approach is not supported by all current cards.

The same issues concerning dynamic power control have been tackled by Iannone and Fdida [1]. The authors, using a simple cross-layer design, integrate several link quality parameters and dynamic power control in the routing framework. The work is based on a simple heuristic aiming at optimizing, the communications between each pair of neighbors in terms of rate, interference, and packet error rate (PER). Once again, although the simulation analysis presented in the paper leads to interesting results, in today's system their solution can be hardly implemented and cannot be supported by all wireless cards.

In order to simplify the implementation of power control based approaches, its management needs to be improved, starting from the physical (card) level, up to the user space of the operating system.

3. Practical issues

The wireless adapters market comprises a wide choice of cards that can be classified according to their *chipsets*. Almost every manufacturer chooses to embed in their cards one of the available chipsets, while a minority, like Cisco and Intel, opted for proprietary solutions in some of their products. Even if all network adapters are created for the same purposes, they present important heterogeneity in terms of features, flexibility, and compatibility due to their different chipsets. Among the large number of chipsets on the market (*e.g.*, Atheros, Prism, Realtek, Ralink, Orinoco, ADMtek, and Broadcom), only few of them provide enough functionalities to correctly accommodate enhanced algorithms, and in particular power control.

Our tests reveal that manufacturers' drivers often do not offer all features provided by the embedded chipset. To evaluate all these features, regardless of the driver provided by the manufacturer, the wireless card must be supported by a free operating system (*e.g.*, Linux, FreeBSD, NetBSD) and drivers. In our tests we used a Linux platform, composed of Fedora 2 (kernel **2.4.20**) distribution on a laptop, and Redhat 9 (kernel 2.4.20) distribution on a desktop. Wireless drivers for Linux are implemented for almost every chipset, under the General Public License (GPL), by several contributors.

Nonetheless, even if a wireless card is supported by Linux, the development level of the driver is also an important issue to be considered. Depending on the card's popularity, a driver is more or less prone to development efforts. Currently, most of the available wireless drivers for Linux do not take full advantage of hardware capabilities because of poor development. This in turn limits the number of chipsets that can be effectively used to implement enhanced power control algorithms.

Among the large choice of chipsets we studied, Atheros chipset fulfills correctly these requirements, thus most of our tests were realized using cards based on this chipset. The Linux driver for Atheros chipsets, namely Madwifi (Multiband Atheros Driver for WiFi), allows to change the transmit power. It is still under development and continuously updated; in our tests we used version 1.3.

4. Experimental tests

The wireless cards that we tested during our measurements (listed in Table 1) belong to four different manifacturers and are mainly based on Atheros chipset. The only non-Atheros card we tested, whose driver supports correctly power control, is the Cisco AIR-PCM352. Hereafter, before presenting the results of our measurements, we describe the tools we used to perform the tests.

Table 1. Some wireless cards and related characteristics.

Manufacturer	Model	Bus type	Chipset	Linux Driver	IEEE 802.11	Power Output (dBm)
D-Link	DWL-AG520	PCI	Atheros	MadWifi	a/b/g	15 ± 2
D-Link	DWL-G650	CardBus	Atheros	MadWifi	b/g	15 ± 2
Netgear	WAG311	PCI	Atheros	MadWifi	a/b/g	15 ± 2
Netgear	WAG511	CardBus	Atheros	MadWifi	a/b/g	15 ± 2
Cisco	AIR-PI21AG-E-K9	PCI	Atheros	MadWifi	a/b/g	10,13,15,17,18,20 (b/g) 10,11,13,14,16 (a)
Cisco	AIR-CB21AG-E-K9	CardBus	Atheros	MadWifi	a/b/g	10,13,15,17,18,20 (b/g) 10,11,13,14,16 (a)
Cisco	AIR-PCM352	PCMCIA	Cisco	Airo-Linux	b	0, 7, 13, 15, 17, 20
3Com	3CRPAG175	CardBus	Atheros	MadWifi	a/b/g	17 (b/g) 16 (a)

4.1. Measurements tools

In order to configure and access information about wireless LAN cards in a standard and uniform way we used the Wireless Extension [10]. This extension is included within current Linux distributions and supported by many chipset drivers. Among the tools included in the package, we used `iwconfig` to change the transmit power. In order to retrieve the Received Signal Strength Indicator (RSSI) for each neighbor in ad hoc mode we used `iwlist`. This tool can display the list of all neighbors with their respective RSSIs.

Whereas Atheros driver returns an exhaustive list of neighbors in range with their RSSIs, all the other chipsets tested only return the list of cells in range and a per-cell RSSI. A cell is defined as the virtual wireless network, identified by a given Service Set Identifier (SSID), formed by nodes using the same frequency. The RSSI of a cell is the average of the values of all nodes that belong to the cell. As a consequence, we present here only results of Atheros-based cards, which can measure the RSSI of a specific node at the receiver. The other cards are not adapted at all to support enhanced power control algorithms (*e.g.* [1, 5]), since even if they can change transmit power, they cannot give the required feedback to the algortihms.

4.2. Measures accuracy

A distinctive feature of the Atheros driver (MadWifi) is the RSSI smoothing, *i.e.* the value returned by the driver is an average of the last few received packets. In order to have accurate measurements, we rather need an exact value of the last received packet, thus we modified the Atheros driver, removing this smoothing behavior.

We opted for a simple measurement set up. Power output is increased from 1 to 60 mW in steps of 1 mW every 20 seconds. For Cisco AIR-PCM352 (PCMCIA, 802.11b), the driver only allows a fixed set of values for the output power, namely 1, 5, 20, 30, 50, and 100 mW.

To retrieve the RSSI regularly, the receiver must get some packets from the transmitter. Thus, for our tests, the latter sends each second a ping packet to keep the RSSI value up to date at the receiver. Changing the time interval between two packets would not change the general behavior of the obtained curves.

Since all measurements were realized indoor, possible interference phenomena could have slightly biased the results, but we repeated each test several times, always observing the same behavior, thus conclusions remain valid in the general case. The purpose of our experiments is not to report only the hardware features of retail wireless cards, but rather to show the general capabilities of the hardware, the driver, and the tools available all together. There is no doubt that with an operating system different from Linux, or different drivers, or other tools, the cards could give different values, though the general behavior certainly remains unchanged. Bridging the gap between theoretical and practical power control solutions does not necessarily require only hardware improvements, but also software ameliorations (*i.e.*, drivers and tools).

4.3. Measurements Results

We first wondered whether retail 802.11 adapters are able to change their transmit power with accuracy. We tested the wireless network adapters listed in the previous part, using the Cisco AIR-PI21AG-E-K9 (PCI bus) as receiver because of its external antenna, placed one meter away from the tested transmitter card.[1]

None of the wireless adapters tested are able to change with accuracy their power output. Nevertheless, for some cards we can roughly determine different levels. For Cisco AIR-PCM352 (PCMCIA bus), the RSSI values reported by `iwlist` are coherent but, as shown in Fig. 1, each power level needs an average calculation to clearly appear.

The first set of cards we tested is composed of the Cisco AIR-PI21AG-E-K9 (PCI bus) and AIR-CB21AG-E-

---

[1] Due to space contraints we cannot show the plots of all tested cards. We show the most representative plots and describe the behaviour of the other cards compared to them.

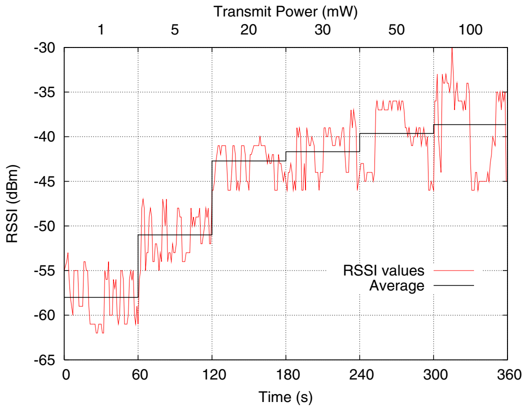

Figure 1. Received signal strength with different output powers for Cisco Aironet 352 (AIR-PCM352).

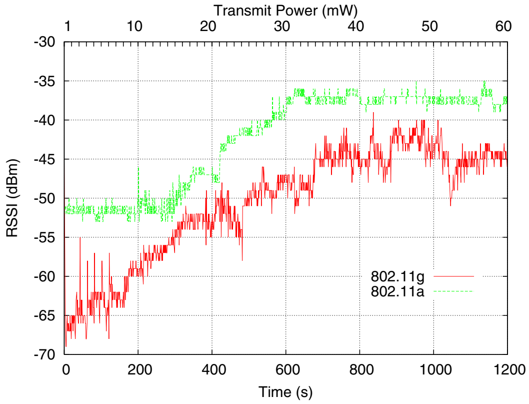

Figure 2. Received signal strength with different output powers for Cisco 802.11a/b/g (AIR-PI21AG-E-K9).

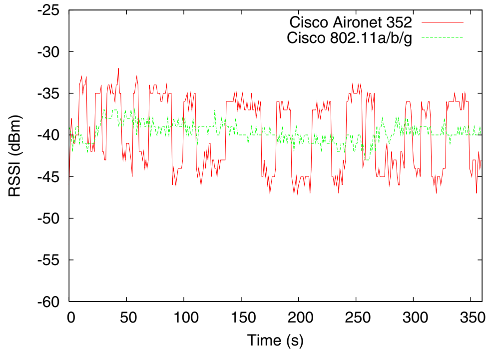

Figure 3. Signal strength received from Cisco cards, transmitting at 50 mW using 802.11b.

K9 (Cardbus) cards. We obtained a growing RSSI curve while increasing the transmit power, but no change can be observed above 35 mW. Figure 2 shows that Cisco wireless adapters seem to keep roughly the same behavior at 35 mW as well as at 60 mW. Cisco wireless network adapters did not present a different behavior when transmitting with 802.11b, 11g, or 11a. Since the above two cards are the most representative, in the remaining of the section we refer to them simply as Cisco a/b/g cards.

The second set of adapters is represented by two Netgear cards. When performing measurements using a/b/g, Netgear WAG311 (CardBus) presents roughly the same results than Cisco a/b/g cards, while Netgear WAG511 (PCI) in b/g seems not to change its transmit power; the curve obtained from RSSI values at the receiver remains almost flat from 1 mW to 60 mW. Nonetheless, when performing the same experiment using 802.11a technology with Netgear WAG511 we obtained a growing curve, similar to the one obtained for the Cisco a/b/g cards.

The third set of adapters is represented by two D-Link cards. D-Link DWL-AG520 (PCI) card presents only two power output levels; one under 30 mW and the other level above this value. Yet, the D-Link DWL-G650 (CardBus) tested gave the same results than Cisco a/b/g cards. RSSI values show a variation in the transmit power below 35 mW, but the RSSI curves remain almost flat above this value.

The last Atheros-based card tested is a 3Com 3CR-PAG175 card. Whereas we can hardly distinguish two power levels below and above 35 mW using b/g, a much better curve is once again obtained when transmitting with 802.11a.

For all cards, except Cisco AIR-PCM352, the power measured when the adapter is set at 35 mW seems to be the maximum value. From 35 mW to 60 mW, the RSSI values remain nearly unchanged. This result does not fit exactly the manufacturers' specifications. The current Atheros driver is only able to set the power up to 60 mW, whereas some wireless adapters are supposed to transmit up to 100 mW. Other adapters, like D-Link, Netgear and 3Com, surprisingly presented some power adaptation capabilities whereas, according to their manufacturer, they were not supposed to do it.

We performed also some tests concerning signal stability. Figure 3 shows how Cisco AIR-PCM352 was unable to keep stable its output power during these tests. Approaches like [5], where the signal strength measures are used to estimate the link quality for each neighbor, hardly can work with such kind of adapters. Other cards, transmitting on the same frequency, using 802.11b, and with exactly the same conditions, seem to keep their output power much more stable. As example in Figure 3 we plotted the stability behavior of two Cisco cards. To be sure that these fluctuations were not due to external phenomena (*e.g.*, interferences),

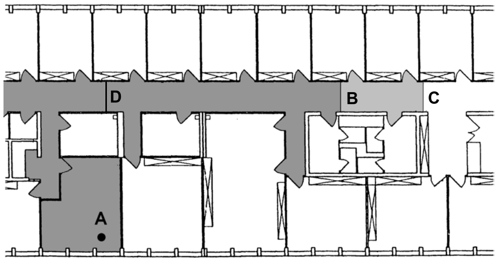

Figure 4. Indoor signal propagation when transmitting from `A`**.**

we performed several tests for each card, and each one led to the same conclusion. Obviously, a totally flat signal is not possible because of well-known radio properties.

The above measurements show clearly that there is often a large difference between the nominal transmit power, set by the user, and the power effectively used by the adapters. If the effective transmit power varies in a small interval, as many adapters do, then also the transmission range has limited variations. When the signal strength magnitude is not sufficient, all signals transmitted with different levels will be lost together within a short range of a few meters. Many power control algorithms rely on adapting the signal strength to the intended destination distance. If the target node is close, then lowering the output power will avoid disturbing unconcerned neighbors (farther away than the destination). To implement such a solution, a fine control on the transmission range is required. Nevertheless, the RSSI variation between the weakest nominal output power (5 mW) of D-Link DWL-AG520 and its highest nominal output power (60 mW) is only of 5 dBm, in our indoor environment. Thus, cards like D-Link DWL-AG520 are not able to significatively vary their transmission range. Figure 4 shows an example of this phenomena, illustrating the radio propagation measured indoor when transmitting with the D-Link DWL-AG520 at point `A`. From point `A` to point `B`, the signal is always received regardless of the transmit power used. Between `B` and `C` the signal is not always received, depending on the transmit power level of the card. Over the point `C`, no signal at all is received even with the maximum power level. Thus, power variation is effective only within the few meters between `B` and `C`. For Cisco a/b/g cards, instead, the variation of the received signal is around 20 dBm, which enlarges the area where power variation is effective, between points `D` and `C` in Figure 4.

5. Conclusion

Despite the amount of solutions proposed using power control approaches to improve wireless networks performance, real implementations are still to come. There is today a non negligible gap between theoretical research and the solutions available in the market. In this paper, we addressed this issue and have shown that existing wireless devices do not fulfill the requirements of power control algorithms.

What comes out from the results we provided above is that applying power control algorithms on real ad-hoc networks rises unexpected issues that simulations cannot reveal. Furthermore, retail IEEE 802.11 cards are heterogeneous, in terms of behavior and performance, as well as in terms of power control support.

In order to implement sophisticated cross-layer or topology control algorithms, there are still improvements to be done. Starting from the chipset, up to the protocol stack of the operating system, passing through the driver, a uniform support to power control should be designed. The development of a general framework for power control would be an effective mean to bridge the gap between theoretical and practical solutions.

References

[1] L. Iannone and S. Fdida. MRS: A simple cross-layer heuristic to improve throughput capacity in wireless mesh networks. In *Conext*, Oct. 2005.

[2] IEEE. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Standard 802.11*, June 1999.

[3] E. Jung and N. Vaidya. A power control mac protocol for ad-hoc networks. In *ACM Mobicom*, Sept. 2002.

[4] V. Kawadia and P. Kumar. Principles and protocols for power control in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks (Part I)*, Jan. 2005.

[5] F. Klemm, Z. Ye, S. Krishnamurthy, and S. Tripathi. Improving TCP performance in ad hoc networks using signal strength based link management. *Ad Hoc Networks Journal*, 2004.

[6] J. Monks and W. Hwu. A power controlled multiple access protocol for wireless packet networks. In *IEEE Infocom*, Apr. 2001.

[7] S. Park and R. Sivakumar. Load sensitive transmission power control in wireless ad-hoc networks. In *IEEE Globecom*, Nov. 2002.

[8] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *IEEE Infocom*, Mar. 2000.

[9] M. Subbarao. Dynamic power-conscious routing for manets: An initial approach. In *IEEE Vehicular Technology Conference*, May 1999.

[10] J. Tourrilhes. Wireless tools for linux. Available online at: http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html.