

# Design and performance evaluation of throughput-aware rate adaptation protocols for IEEE 802.11 wireless networks

Emilio Ancillotti, Raffaele Bruno\*, Marco Conti

Institute for Informatics and Telematics, Italian National Research Council, Via G. Moruzzi 1, 56124 Pisa, Italy

## ARTICLE INFO

### Article history:

Received 9 December 2008  
Received in revised form 7 July 2009  
Accepted 6 August 2009  
Available online 31 August 2009

### Keywords:

802.11 wireless networks  
Rate adaptation algorithms  
Experimental evaluation

## ABSTRACT

Experimental studies have shown that traditional rate adaptation schemes for 802.11 wireless networks suffer from significant throughput degradation in highly congested networks. To address this problem, this paper makes the following two main contributions. First, we design a method to accurately estimate the per-packet transmission times, and we use these measurements to provide a broad classification of the network state and to identify network congestion. Second, we design a new Throughput-Aware Rate Adaptation (TARA) scheme, which uses the congestion estimates to mitigate the negative impact of link-layer collisions on the operations of rate adaptation, without requiring changes in the 802.11 MAC specification. Another key feature of our solution is to minimize probing overheads and limit unnecessary rate decreases by predicting the throughput gain that could be brought about by a change in the transmission rate. Through experiments conducted across a variety of network scenarios and traffic patterns, we show that TARA achieves significantly higher throughput than the other rate adaptation algorithms implemented in the legacy Madwifi driver.

© 2009 Elsevier B.V. All rights reserved.

## 1. Introduction

Over the last decade we have witnessed the proliferation of IEEE 802.11 (WiFi) networks. The IEEE 802.11 technology [1] was originally designed for providing wireless network connectivity to nomadic users, i.e., users carrying portable devices, such as laptops, who connect to the Internet from fixed locations. Therefore, the large majority of existing 802.11 wireless networks are small-scale, infrastructure-based networks, commonly called wireless LANs (WLANs), deployed in home or office environments. Nevertheless, this is now rapidly changing. The improvements of physical layer technologies, such as the adoption of OFDM modulations, have enabled transmission rates of up to 54 Mbps in IEEE 802.11a/g mode, and even higher rates are envisioned in future extensions, e.g., adding MIMO capabilities to IEEE 802.11n [2]. In addition, the deployment of infrastructure-based wireless networks in public places, commonly known as WiFi “hot-spot” networks, where people can benefit from a seamless public access to the Internet, has undergone a substantial growth. Finally, significant efforts are also underway to extend the use of IEEE 802.11 technologies to metro-scale wireless mesh networks, by taking advantage of multi-hop communications to form a mesh backbone among APs [3,4].

A key aspect of the IEEE 802.11 technology is the *rate adaptation* scheme, i.e., the mechanism used by each station to select the transmission rate that provides the *best* link performance (usually measured in terms of maximum link capacity). Indeed, the inaccurate selection of the optimal transmission rate for the current channel state, as well as the inability to quickly adapt the data rate to the fluctuations of wireless channel conditions, may lead to significant performance degradations. Therefore, several rate adaptation algorithms have been proposed to address these problems. These schemes primarily differ in the

\* Corresponding author.

E-mail addresses: [e.ancillotti@iit.cnr.it](mailto:e.ancillotti@iit.cnr.it) (E. Ancillotti), [r.bruno@iit.cnr.it](mailto:r.bruno@iit.cnr.it) (R. Bruno), [m.conti@iit.cnr.it](mailto:m.conti@iit.cnr.it) (M. Conti).

metrics used to estimate the channel quality, and in the heuristics adopted to select the best transmission rate. Most of the practical rate adaptation protocols implemented in legacy cards select the transmission rate based on specific patterns of frame losses and successes [5–7], or on congestion-related statistics derived from observed retransmission probabilities [8–12]. Recently, exhaustive experimental studies have been conducted to analyze and compare the performance of these schemes [13–16]. These measurements studies have clearly shown that existing rate adaptation schemes relying only on estimates of link qualities, and frame loss characteristics suffer from significant throughput degradation when transmission failures are caused by collisions. To mitigate the negative impact of collisions on the rate selection, a number of rate adaptation protocols employ loss-differentiation techniques based on the selective use of the RTS option [9,17], explicit failure notifications [18–20], or complex theoretical models [12]. However, RTS/CTS handshake introduces a high overhead and it is usually not activated in currently deployed 802.11 WLANs. In addition, many of these solutions require changes to the IEEE 802.11 standard, and are not compatible with existing cards.

In this paper we address the problem of designing more efficient rate adaptation algorithms that ensure high system performance in both congested and/or highly interfered networks. To this end, this work makes the following two main contributions. First, we design a method to accurately estimate the per-packet transmission times, and we use these measurements to provide a broad classification of the network state and to identify network congestion. We evaluate the efficiency of this technique through experiments in both a WLAN and an outdoor mesh network. The second contribution is to employ this monitoring tool of the network state in the design and implementation of a novel rate adaptation algorithm for 802.11 wireless networks, called the *Throughput-Aware Rate Adaptation (TARA)* protocol. The key feature of the TARA scheme is to mitigate the negative impact of collisions on the operations of rate adaptation, without relying on costly RTS/CTS probing for loss differentiation, or changes in the legacy MAC protocol. The basic idea behind the TARA design is to exploit the estimates on packet transmission times and network activity to select the transmission rate that will provide the highest throughput performance. TARA uses probing to update statistics at different data rates. However, a key feature of our solution is to minimize the probing overheads and limit unnecessary rate decreases by predicting the throughput gain that could be brought about by a change in the transmission rate. We have implemented the TARA scheme in the open-source Madwi driver [21], and we have experimentally evaluated its performance across a variety of network scenarios and traffic patterns. Our experimental results show that TARA significantly outperforms the other rate adaptation algorithms currently implemented in the Linux Madwifi driver, especially in environments where collisions induce non-negligible frame loss rates. In addition, in most of the investigated scenarios, TARA is able to closely approach the maximum throughput obtained with a fixed transmission-rate strategy.

The remainder of this paper is organized as follows. In Section 2, we review the related work. Section 3 describes the key features of the TARA scheme, and it illustrates the design and implementation of its main components. Section 4 compares the TARA scheme to the alternative rate adaptation algorithms implemented in the Madwifi driver. Finally, Section 5 concludes this paper with a discussion of the main results and future activities.

## 2. Background

The aim of this section is to discuss the strengths and weaknesses of representative approaches adopted for rate adaptation in 802.11-based wireless networks. More details will be given on the rate adaptation schemes AMRR [6], ONOE [7] and SampleRate [8] that we have selected for the experimental evaluation and comparison.

In general, rate adaptation involves three main tasks: (i) the assessment of the channel conditions, (ii) the selection of the best transmission rate, and (iii) the corresponding rate change. Various classifications of the existing rate adaptation schemes can be proposed, based on the methods used to implement these functions. In this work we suggest classifying the rate adaptation schemes according to the type of metrics used to assess the channel quality, and we identify two broad categories: solutions based on *physical layer measurements*, and solutions based on *MAC layer measurements*.

### 2.1. Rate adaptation schemes based on physical layer measurements

This class of rate adaptation algorithms relies on measurements of the wireless signal characteristics, such as the signal-to-noise ratio (SNR) or the received signal strength indicator (RSSI), to infer the transmission rate that would provide the maximum link-layer throughput [22]. In general, SNR-based protocols estimate the channel state at the receiver based upon the signal strength of control messages. Then, the receiver determines the best modulation rate from these measurements, and informs the transmitter through a four-way handshake. For instance, both RBAR (Receiver-Based AutoRate) [23] and OAR (Opportunistic Auto-Rate) [24] employ RTS frames to estimate the SNR at the receiver, and piggyback this information to the sender on the subsequent CTS frames, so that the sender can adjust the rate accordingly. In [23] the sender transmits a single data packet at the new rate and then it repeats the rate selection, while in [24] time-share fairness is achieved by sending a burst of data packets in proportion to the ratio between the currently used transmission rate and the base data rate. A simpler approach is adopted in [25], which relies on RTS/CTS probing. Specifically, assuming channel reciprocity, in [25] each station adapts the transmission rate depending on the RSS of the frames received from the AP. Channel reciprocity is also leveraged in [26], where RSSI measurements are used to predict the future channel conditions, while historical information on the success or failure of packets is used to estimate the instantaneous path loss to the destination.

In general, it might be difficult to obtain accurate SNR or RSSI measurements from legacy cards [16,26]. Moreover, SNR-based rate adaptation schemes need an accurate channel model to establish a correct mapping between the signal strength measurements and the corresponding link performance, which is difficult due to the complexities and variability of radio propagation phenomena, especially in outdoor environments. In this case, the only practical solution might be *in situ* profiling to predetermine the relations between the delivery ratio and the SNR for that specific environment. However, the generation of accurate profiles usually requires extensive and time-consuming measurement campaigns. Furthermore, many experimental studies conducted in typical urban environments have shown that, in general, there is a loose correlation between the delivery ratio and the SNR because link performances are dominated by multi-path effects [27,28].

## 2.2. Rate adaptation schemes based on MAC layer measurements

The rate adaptation schemes that collect statistics on transmission events (e.g., number of retries, patterns of successful or failed transmissions, etc.) to guide the rate selection process fall within the category of solutions based on MAC layer measurements. Generally, the advantage of these schemes is simplicity, because commercial cards usually provide a good support to access information on packet receptions at the driver level. On the negative side, collecting meaningful statistics on frame loss patterns can be too slow to follow rapid channel fluctuations. In addition, as we will explain in more detail the following, it is quite complicated to distinguish the causes of observed transmission failures, and reducing the transmission rate in response to a frame collision significantly degrades the system throughput.

The earliest example of this type of rate adaptation algorithm is ARF (Automatic Rate Fallback) [5]. With ARF, a sender starts transmitting at the base data rate, and triggers a timer. After  $n$  consecutive successful transmissions (usually  $n = 10$ ), or a timer expiration, the sender transmits a *probe packet* at a higher data rate, and resets the timer. If the transmission of the probe packets fails, the sender immediately falls back to the previous rate; otherwise, it keeps using the new transmission rate. The transmission rate is also decreased upon two consecutive transmission failures. The simple heuristic employed by the ARF scheme has inspired several variants, such as AARF (Adaptive ARF) [6], AMRR (Adaptive Multi Rate Retry) [6] and ONOE [7], which try to reduce the probing overhead, and to make the rate adaptation process less vulnerable to short-term channel fluctuations. For instance, in AARF the probing threshold  $n$  used in ARF is variable: if the transmission of the probing packet fails, not only does the sender immediately fall back to the previous rate, but it also doubles the probing threshold. The approach followed in AMRR [6] is more sophisticated. This scheme assumes that four data rates ( $r_0, r_1, r_2, r_3$ ) and transmission counts ( $c_0, c_1, c_2, c_3$ ) are associated to each frame. More precisely, the modulation rate  $r_0$  is used for the first frame transmission attempt, and for the next  $c_0 - 1$  retries. If the transmission keeps on failing, the transmitter tries rate  $r_1$  for  $c_1$  times, then rate  $r_2$  for  $c_2$  times and, finally, rate  $r_3$  for  $c_3$  times before discarding the packet. In [6] it is suggested to set  $c_0 = c_1 = c_2 = c_3 = 1$ , i.e., each rate is tried just once. Concerning the data rates,  $r_3$  is the lowest modulation rate, while  $r_1$  is the rate immediately lower than  $r_0$ , and  $r_2$  is the rate immediately lower than  $r_1$ . To select  $r_0$ , the AMRR algorithm employs the following heuristic: if less than 10% of the packet transmissions failed during the last observation period (and total number of frame transmissions is at least ten), then increase the data rate; otherwise, if more than 33% of the packet transmissions failed during the last period (and the total number of frame transmissions is at least ten), then decrease the data rate. By default, the observation period is equal to one second.

The ONOE algorithm [7] is a variant of the AMRR scheme. Specifically, ONOE uses larger retransmission counts than AMRR (namely,  $c_0 = 4$  and  $c_1 = c_2 = c_3 = 2$ ), while it sets the  $r_1, r_2, r_3$  data rates in the same way as AMRR. The major difference between the two schemes is that the ONOE algorithm associates a number of *credits* to the current rate  $r_0$ . More precisely, if less than 10% of the frame transmissions failed during the last period, then the credits are incremented by one; otherwise, the credits are reduced by one. The data rate  $r_0$  is increased only if the total credits accumulated by the sender are above a threshold (the default is ten). If more than 50% of the frame transmissions failed during the last period, then the transmission rate is immediately decreased. Whenever the rate is changed, the credit counter and the rate statistics are reset. Note that ONOE is less sensitive to single packet failures than AMRR. However, it is also very conservative, and it takes several seconds to increase the transmission rate. An alternative approach has been recently proposed in [12], which extends the ARF scheme by introducing variable rate-increasing and rate-decreasing thresholds. The basic idea is to compare the pattern of frame retries to a theoretical ARF model in order to infer the channel conditions and to dynamically find new collision-robust thresholds.

The SampleRate scheme designed in [8] is based on an approximate estimate of the expected packet transmission time. Normally, the SampleRate algorithm transmits each frame at the data rate with the shortest expected packet transmission time, with the objective of ensuring the best throughput performance in the long term. In addition, SampleRate sends probe packets every ten frames to collect statistics at different transmission rates. However, SampleRate tries to probe only rates that may perform better than the current one. To this end, SampleRate randomly selects one transmission rate from the set of rates whose *lossless* (i.e., without any retries) packet transmission time is lower than the average transmission time of the current bit-rate. Similar to SampleRate is the BEWARE algorithm [11], which uses a mathematical model to characterize the expected transmission time from the frame error probability and the probability that nearby stations transmit a packet. Finally, a further variant of the SampleRate scheme is the Wireless cOngestion Optimized Fallback (WOOF) protocol [10], which uses the channel utilization, i.e., the fraction of time for which the medium is utilized in some time interval, to correct the SampleRate estimates of expected packet transmission times in congested environments.

Experimental studies [8,15] have pointed out that, as the number of contending stations increases, SampleRate has generally better performance than other probe-based approaches using fixed loss thresholds for increasing/decreasing the

transmission rate. However, in collision-dominated environments, significant throughput degradations are still observed due to unnecessary rate decreases when the frame losses are due to collisions [15,16]. To improve the performance and efficiency of rate adaptation schemes when transmission failures are caused by collisions, a number of schemes have been proposed, such as CARA (Collision-Aware Rate Adaptation) [17] and RRAA (Robust Rate Adaptation Algorithm) [9], which selectively activate RTS/CTS exchange on a per-frame basis. The key assumption is that the transmission failure of an RTS frame is the result of a collision, while after a successful RTS/CTS exchange the successive data frame transmission can only fail due to channel errors. Therefore, the rate decision process reacts only to wireless losses that are not involving RTS frames. In other words, RTS transmissions are used to filter out collisions. CARA is basically an extension of ARF, which selectively turns on the RTS option after the loss of a single packet, and then off for the following packet. On the other hand, RRAA calculates the frame error rate as the ratio between the number of ACK frames received and the number of transmitted frames. Then, RRAA decreases or increases the data rate if this ratio exceeds predetermined thresholds. Moreover, RRAA uses an adaptive RTS filter to suppress collision losses when it estimates this loss ratio. However, per-frame selective RTS is a costly solution in lossy environments, and it is not effective when differences in average link qualities exist between competing transmitters [16]. In addition, filtering out collision losses may result in *biased* rate decisions when the throughput decrease is caused jointly by channel errors and collision events [11]. Finally, alternative approaches are proposed in [18–20], which add extra frames and fields to explicitly notify the sending station whether the transmission failure is due to collision or noise. However, these solutions require changes to the IEEE 802.11 standard, and they are not compatible with existing cards.

### 3. TARA: Throughput-aware rate adaptation

In this section, we describe the proposed *Throughput-Aware Rate Adaptation* (TARA) scheme. Our solution is based on two main functionalities: (i) a method to measure the duration of transmission events and to infer the network state, and (ii) a probe-based rate adaptation algorithm that maximizes the link-layer throughputs while reducing the probing overheads and limiting unnecessary rate changes. In the following we illustrate these two components separately.

#### 3.1. Monitoring channel state

In wireless networks, throughput measurements are affected by a multiplicity of factors, including packet collisions caused by congestion or external interference, lossy channels, and the chosen transmission bit-rate. For these reasons, it is generally recognized that the measurement of the durations of specific events are more indicative of the network state than the observed throughput. For instance, in [29] it is proposed to use the number of consecutive idle slots between two transmission attempts as a measure of the network state. In contrast, several studies have proposed to use the medium utilization, defined as the fraction of time for which the wireless channel is busy within a given interval, as a measure of congestion in the wireless medium. In [30] the medium utilization is measured by adding the transmission duration of all data, management, and control frames recorded by a “sniffer”. However, this approach involves significant processing overhead for each received packet, making it unsuitable for congestion identification in real time. A more practical and lightweight approach is proposed in [10], which approximates the medium utilization using the number of clock ticks for which the medium is sensed to be busy. In [10] it is shown that in some situations the channel busy time is a good indicator of the packet losses caused by network congestion. However, this finding cannot be generalized because this measure is not able to distinguish between different types of channel occupations, i.e., transmissions or external interferences. Moreover, the measurement of channel clocks requires access to the network card firmware, which is not generally provided in legacy drivers.

In the following we describe an alternative method that can be readily implemented in legacy Madwifi drivers, aiming to identify network contention through measurements of how long it takes to successfully complete a frame transmission. Moreover, in the following we also show how to exploit these measurements to broadly classify the network state as *collision dominated* or *noise dominated*. For ease of explanation, in Fig. 1 we illustrate the events observed on the wireless channel by a monitoring station before completing the successful transmission of its  $k$ th packet. Without loss of generality, in the following development we assume that all the packets have a fixed size equal to  $l$  bytes.

Let  $t_{HOL}^k$  be the time instant when the  $k$ th packet of the monitoring station is ready for being transmitted, i.e., has reached the head-of-line of the station’s transmission buffer. Let us denote with  $t_{succ}^k$  the time instant when the monitoring station has successfully completed the transmission of the  $k$ th packet. In saturated conditions, it holds that  $t_{HOL}^k = t_{succ}^{k-1}$ . However, if we consider stations generating bursty, or sporadic traffic, the transmission buffers can be empty. In this case, the  $t_{HOL}^k$  value corresponds to the time instant when the  $k$ th packet is generated. Intuitively, the total time needed to successfully transmit the  $k$ th packet using modulation rate  $r$ , say  $T_{MAC}^k(r, l)$ , can be simply expressed as

$$T_{MAC}^k(r, l) = t_{succ}^k - t_{HOL}^k. \quad (1)$$

This metric can be readily used to compute the instantaneous link-layer throughput  $\gamma^k(r, l)$  as follows:

$$\gamma^k(r, l) = \frac{l}{T_{MAC}^k(r, l)}. \quad (2)$$

In the following we detail how to practically estimate the  $T_{MAC}^k(r, l)$  value using information available at the driver level.

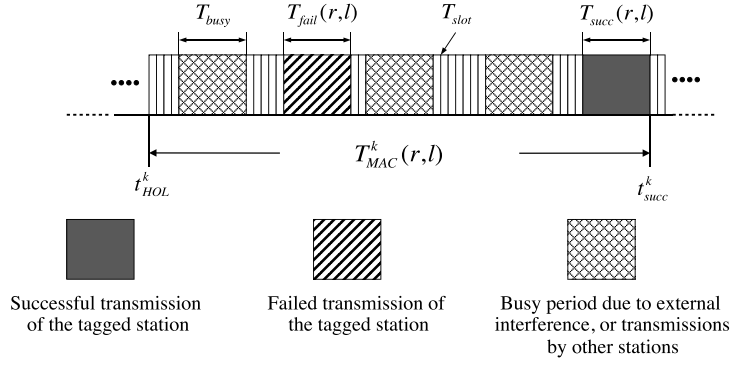


Fig. 1. Channel events observed by a monitoring station before the successful transmission of the  $k$ th packet.

Let  $n_{\text{retry}}^k$  be the number of frame retransmissions performed by the monitoring station before successfully transmitting the  $k$ th packet. Let us denote with  $CW(i)$  the length of the congestion window used by each station to randomly select the backoff timer preceding the  $i$ th retransmission of the same frame. Based on the 802.11 MAC specification, the total time, say  $T_{\text{INT}}^k(r, l)$ , spent by the monitoring station backing off or transmitting during the  $T_{\text{MAC}}^k(r, l)$  period can be approximated as

$$T_{\text{INT}}^k(r, l) \cong \sum_{i=0}^{n_{\text{retry}}^k} \frac{CW(i) - 1}{2} \cdot T_{\text{slot}} + n_{\text{retry}}^k \cdot T_{\text{fail}}(r, l) + T_{\text{succ}}(r, l), \quad (3)$$

where  $T_{\text{fail}}(r, l)$  is the duration of a failed frame transmission using modulation rate  $r$ , and  $T_{\text{succ}}(r, l)$  is the duration of a successful frame transmission using modulation rate  $r$ , both including the MAC protocol overheads (e.g., ACK timeout and IFS intervals). Since the Madwifi driver does not expose the backoff timer used for each transmission attempt, we approximate this parameter using its average value, i.e.,  $(CW(i) - 1)/2$ . Note that  $T_{\text{INT}}^k(r, l)$  corresponds to the transmission time computed in SampleRate [8]. Now it is straightforward to compute the total time, say  $T_{\text{EXT}}^k(r, l)$ , the channel is busy due to events not involving the monitoring station's transmissions and backoffs as follows:

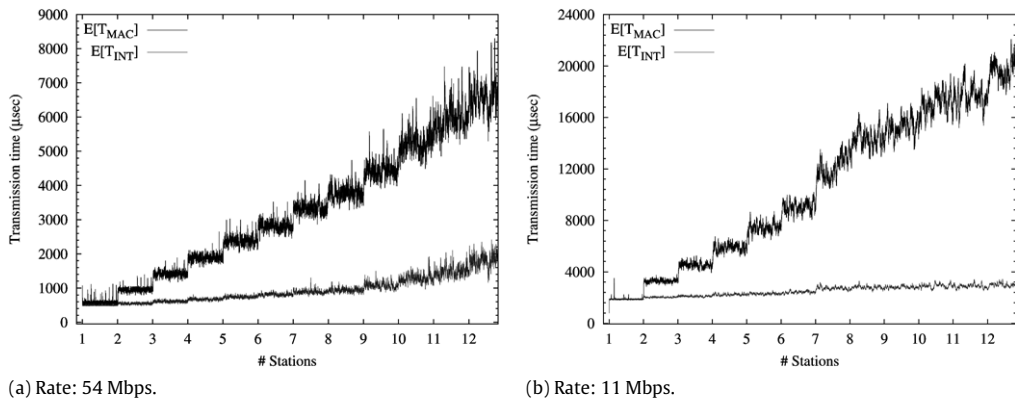
$$T_{\text{EXT}}^k(r, l) = T_{\text{MAC}}^k(r, l) - T_{\text{INT}}^k(r, l). \quad (4)$$

In general, individual measurements of  $T_{\text{INT}}^k(r, l)$  and  $T_{\text{EXT}}^k(r, l)$  cannot be used to reliably identify and characterize the network congestion levels, unless we have theoretically derived values that establish a relationship between measured channel occupations and network congestion levels. However, such analytical values would require accurate models that are able to capture the impact of several factors, including coordination issues for shared access methods, propagation impairments and external interference, which is difficult to obtain. In contrast, our claim, which will be validated through experimental results, is that the ratio between the parameters  $T_{\text{INT}}^k(r, l)$  and  $T_{\text{EXT}}^k(r, l)$  can be a good metric to broadly classify the network state, and to distinguish, at least, between collision-dominated or noise-dominated environments.

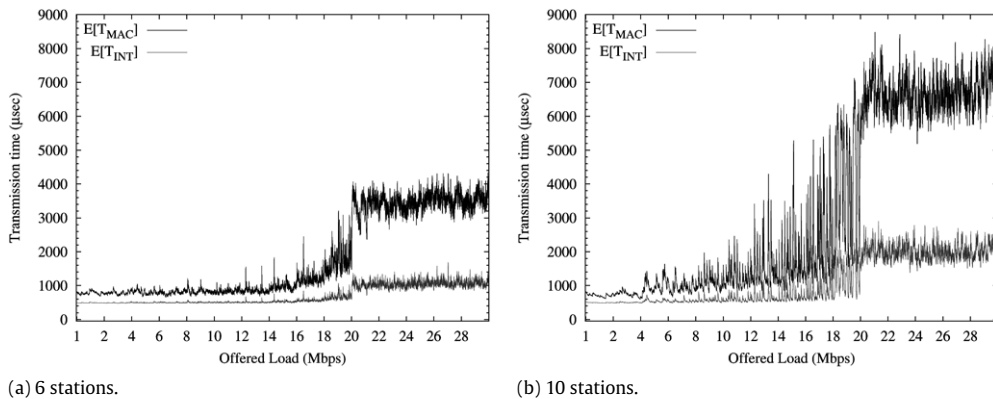
### 3.1.1. Implementation and validation of time-based metrics

To compute the metrics defined in formulas (1), (3) and (4) we patched the Linux Madwifi driver in order to collect the following measurements: (1) the time instants when MAC ACK frames are received, and (2) the time instants when the transmission buffer changes its status from empty to not empty and vice versa. Note that the legacy drivers already provide monitoring tools to measure the per-packet number of retransmissions, which is needed to estimate the  $T_{\text{INT}}^k$  parameter given in (3). To reduce the fluctuations of the measurements, and to provide run-time estimations reflecting average behaviors, we use EWMA (Exponentially Weighted Moving Average) filters. A first set of experiments was performed in a homogeneous WLAN scenario, consisting of an AP and up to 12 static clients. These clients are IBM Thinkpad model R50E laptops. Each of these machines is equipped with a NetGear WPN511 card. Note that we do not use a commercial AP but a computer instrumented as the access point in order to have full access to all implementation details. This experimental WLAN is deployed on one floor of a typical office building, where other uncoordinated APs, generally operating on non-orthogonal channels, are also deployed. However, clients are positioned close to the AP they are associated to ensure similar channel conditions.

A first group of experiments was conducted in saturated conditions, i.e., each station sends as many UDP packets as possible to the AP. Every 60 s a new client is activated. If not otherwise stated, the packet size is 1500 bytes. In Fig. 2 we report the  $E[T_{\text{MAC}}(r, l)]$  and  $E[T_{\text{INT}}(r, l)]$  estimates versus the number of active stations for two representative transmission rates (the  $E[\cdot]$  notation indicates the average operator). From the experimental results it is straightforward to observe that the  $E[T_{\text{MAC}}(r, l)]$  value rapidly increases as the number of contending stations increases, and that the  $E[T_{\text{EXT}}(r, l)]$  contribution becomes dominant over  $E[T_{\text{INT}}(r, l)]$ . More precisely, the experimental results indicate that the ratio between  $E[T_{\text{EXT}}(r, l)]$  and  $E[T_{\text{INT}}(r, l)]$  is greater than 1 for  $n > 2$ , independently of the  $r$  value. This initial finding supports our argument that  $E[T_{\text{MAC}}(r, l)]/E[T_{\text{INT}}(r, l)] > 1$  might be a good indicator of collision-dominated environments, where external factors (i.e., transmissions from other stations or interference from external radio sources) dominate the overall network performance.



**Fig. 2.**  $E[T_{MAC}]$  and  $E[T_{INT}]$  estimates in a saturated WLAN versus the number of active stations.



**Fig. 3.**  $E[T_{MAC}]$  and  $E[T_{INT}]$  estimates in an unsaturated WLAN versus the total generated load.

To further support the above observation, we have performed a second group of experiments in unsaturated conditions. In these tests each station activates a constant bit-rate (CNR) flow towards the AP using UDP as the transport layer. All the flows generate the same load, and every 60 s the total offered load is incremented by 1 Mbps until the network becomes saturated. In Fig. 3 we report the  $E[T_{MAC}(r, l)]$  and  $E[T_{INT}(r, l)]$  estimates versus the total generated load for two representative network sizes. From the experimental results it is straightforward to observe that in unsaturated conditions the  $E[T_{MAC}(r, l)]$  value is not dominated by the  $E[T_{EXT}(r, l)]$  value, although the measurements are affected by a high variability when many flows are active. This variability, as well as the one observed in Fig. 2, can be explained by noting that the 802.11 exponential backoff algorithm can easily induce short-term unfairness [31,32]. Indeed, the variability of the backoff timer is the main cause of the large deviation of individual samples of packet transmission times. This may also have a significant impact on the design of the rate adaptation algorithm because following the transient fluctuations of link-layer throughput may lead to unnecessary rate changes.

A second set of experiments was conducted in an experimental outdoor mesh network deployed in the CNR's campus area. This mesh test-bed consists of five mesh routers deployed on the rooftops of various building, which are equipped with both directional and omni-directional antennas. The hardware setup of this network are described in more details in our previous paper [15]. Fig. 4 illustrates the network topology and connectivity graph of our test-bed. Solid lines indicate links between directional antennas, while dashed lines are used to represent links between omni-directional links. As shown in the diagram, all mesh routers except node A are equipped with omni-directional antennas of various gains (8 dBi for nodes B and D, and 15 dBi for nodes C and E). Node A is equipped with one 15 dBi Yagi directional antenna pointing to node C, and one 19 dBi Grid directional antenna pointing to node B, while nodes B and C are equipped with one 19 dBi Grid directional antenna and one 15 dBi Yagi directional antenna, respectively, both pointing to node A. The shortest link in our network is from D to E, which is 80 m long, while the longest link is from A to B, which extends over 280 m. The differences in link distances and antenna characteristics ensure a reasonable variability of link qualities.

In these tests, each mesh router in turn sends UDP packets to one of its neighbours as fast as it can. Each sender transmits for 60 s at each of the 802.11 data rates. Since one link is active at a time, we expect that frame retransmissions are primarily caused by channel errors, or interference originated from nearby APs. In Fig. 5 we report the measured  $E[T_{MAC}(r, l)]$  and  $E[T_{INT}(r, l)]$  values versus time for two representative links. More precisely, we show the results related to link  $A \rightarrow B$ , which obtains the maximum link throughput using the 54 Mbps data rate, and link  $B \rightarrow E$ , which obtains the maximum



Fig. 4. Connectivity graph of our experimental outdoor mesh network. Solid lines are directional links, while dashed lines are omni-directional links.

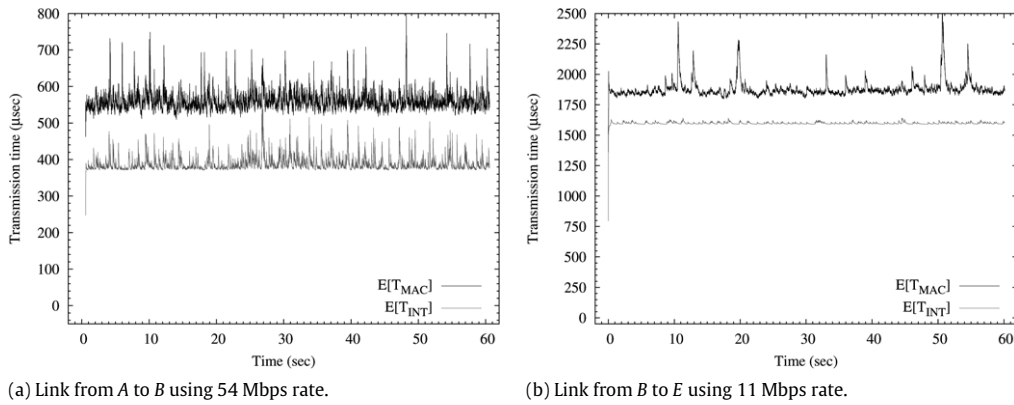


Fig. 5.  $E[T_{MAC}]$  and  $E[T_{INT}]$  estimates for two representative links of our outdoor mesh network.

link throughput using the 11 Mbps data rate. The plotted curves indicate that now the  $E[T_{INT}(r, l)]$  time is the dominant contribution in the overall  $E[T_{MAC}(r, l)]$  time. This confirms our initial intuition that, with low contention levels, the impact on the average packet transmission time of busy periods due to external interference sources, or transmissions from other stations, is negligible. In this case we have that  $E[T_{EXT}(r, l)]/E[T_{INT}(r, l)] < 1$ , and this can be assumed as a sufficient condition to conclude that *internal noise* (i.e., channel errors) dominates the overall network performance. In Section 3.2 we will elaborate more on the importance of this property in the design of the TARA algorithm. Finally, our experimental results indicate that the time estimates are affected by sporadic spikes, which are probably due to the occurrence of transient deep fades on the channel. This also suggests that the packet transmission-time metric can be useful also to identify sudden changes of the channel conditions.

### 3.2. Rate selection and change

The key idea of the TARA algorithm is to use a *performance gain factor*  $G(r, \tilde{r}, l)$  to quantify, given a fixed frame size equal to  $l$  bytes, the expected increase or decrease of the link-layer throughput when changing the transmission rate from  $r$  to  $\tilde{r}$ . Intuitively, the performance gain factor  $G(r, \tilde{r}, l)$  can be defined as

$$G(r, \tilde{r}, l) = \frac{E[\tilde{\gamma}(\tilde{r}, l)]}{E[\gamma(r, l)]}, \quad (5)$$

where  $E[\tilde{\gamma}(\tilde{r}, l)]$  is the *predicted* average link-layer throughput for modulation rate  $\tilde{r}$ , whereas  $E[\gamma(r, l)]$  is the *measured* average link-layer throughput for modulation rate  $r$ , which is computed using formula (2). Then, the  $G(r, \tilde{r}, l)$  parameter can be used to decide which transmission rate should be probed.

The  $E[\tilde{\gamma}(\tilde{r}, l)]$  parameter can be estimated as  $l/E[\tilde{T}_{MAC}(\tilde{r}, l)]$ , where the denominator is the predicted average packet transmission time for modulation rate  $\tilde{r}$ . Similarly to Eq. (4),  $E[T_{MAC}(\tilde{r}, l)]$  is given by  $E[\tilde{T}_{EXT}(\tilde{r}, l)] + E[\tilde{T}_{INT}(\tilde{r}, l)]$ . To derive these two terms we make the following assumptions.

- A change in the transmission rate does not affect the behavior of other stations' transmissions. This implies that

$$E[\tilde{T}_{EXT}(\tilde{r}, l)] = E[T_{EXT}(r, l)]. \quad (6)$$

- If  $\tilde{r}$  is a lower data rate than  $r$ , then the channel become lossless for the tagged station. This is an optimistic assumption because decreasing the rate cannot reduce the collision probability. However, the lack of a precise knowledge of the network contention level makes it quite difficult to accurately anticipate the effect of a rate decrease on the overall retransmission probability. For these reasons, assuming that the channel become lossless after a rate decrease is commonly adopted in other rate adaptation schemes based on estimates of per-packet transmission times, including SampleRate [8] or BEWARE [11]. This condition can be formally expressed as follows:

$$E[\tilde{T}_{INT}(\tilde{r}, l)] = T_{succ}(\tilde{r}, l). \quad (7)$$

- If  $\tilde{r}$  is a data rate greater than or equal to data rate  $r$ , we assume that the channel characteristics do not change, i.e., the tagged station experiences the same number of retransmissions to successfully transmit a packet. This is again an optimistic assumption because higher transmission rates can be more vulnerable to channel noise and interference than lower transmission rates. On the other hand, if the retransmissions are primarily due to collisions, increasing the rate does not increase the collision probability. This condition can be formulated as follows:

$$E[\tilde{T}_{INT}(\tilde{r}, l)] = E[T_{INT}(r, l)] - \Delta(r, \tilde{r}, l), \quad (8)$$

where  $\Delta(r, \tilde{r}, l) = T_{succ}(r, l) - T_{succ}(\tilde{r}, l)$  represents the reduction of the time spent for a successful frame transmission when increasing the modulation rate.

Owing to the above considerations, and using expression (2), formula (5) can be rewritten as

$$G(r, \tilde{r}, l) = \frac{E[T_{EXT}(r, l)] + E[T_{INT}(r, l)]}{E[T_{EXT}(r, l)] + E[\tilde{T}_{INT}(\tilde{r}, l)]}. \quad (9)$$

Now we have all the elements to completely describe the operations of the TARA protocol. For ease of explanation, Algorithm 1 reports the pseudo-code of the TARA scheme, named TARA1. The TARA1 protocol operates on a *per-packet* basis, i.e., for each new packet to transmit the sender executes Algorithm 1 to determine the rate to be used. Normally, TARA1 selects the best transmission rate  $r_{new}$  for the new packet as the modulation rate that provides the maximum average link-layer throughput. To this end, the sender invokes the `FindBestRate1()` function (line 11), which goes through the available  $E[\gamma(r, l)]$  estimates to determine the  $r_{new} \in \mathcal{R}$  that satisfies the following condition:  $r_{new} = \operatorname{argmax}_{r \in \mathcal{R}} \{E[\gamma(r, l)]\}$ . For the sake of reliability, TARA1 does not consider data rates that discarded more than  $MAX\_SUCC\_FAIL$  packets consecutively. This last condition is introduced to avoid using data rates that experienced several recent failures, and it is leveraged from the SampleRate design [8]. In our implementation,  $MAX\_SUCC\_FAIL = 4$ , and this counter is automatically reset every 10 s. Finally, TARA1 periodically probes data rates that may perform better than the one currently in use. More precisely, every time the probing timer expires the station randomly selects a new data rate from the set of data rates that have a gain factor greater than 1. In the case when the probing set is empty, the probing phase is skipped and the normal rule is used to select the transmission rate.

**Algorithm 1.** Pseudo-code of TARA1 operations for each new data packet to transmit.

**Input:** Set  $\mathcal{R}$  of available transmission rates,  
 Measured  $E[T_{EXT}(r, l)]$  and  $E[T_{INT}(r, l)]$  for each transmission rate  $r \in \mathcal{R}$  and packet size  $l$ ,  
 Table of  $G(r, \tilde{r}, l)$  values for each pair of transmission rates  $r$  and  $\tilde{r}$  (with  $r, \tilde{r} \in \mathcal{R}$ ).  
**Result:** Modulation rate  $r_{new}$  for the new packet transmission.

```

1 begin
2    $r_{prev} \leftarrow$  rate used for the previous packet transmission;
3    $\mathcal{R} \leftarrow \emptyset$ ;
4   if (Probe_Timer is expired) then
5     reset Probe_Timer;
6     for all element  $r$  of set  $\mathcal{R}$  do
7       if  $G(r_{prev}, r, l) > 1$  then  $\tilde{\mathcal{R}} \leftarrow \tilde{\mathcal{R}} \cup r$ 
8     end
9   end
10  if ( $\tilde{\mathcal{R}} \neq \emptyset$ ) then  $r_{new} \leftarrow$  random element in set  $\tilde{\mathcal{R}}$ ;
11  else  $r_{new} \leftarrow$  FindBestRate1 ( $MAX\_SUCC\_FAIL$ );
12 end

```



The basic idea behind the TARA1 design is to use the gain factor to limit the probing overhead, and to exploit statistics on the link-layer throughput to select the best transmission rate. However, further improvements would be possible if the protocol design is extended with methods capable of classifying the network state in terms of its congestion level. To this end, we propose a variant of the TARA1 scheme, called *TARA2*, which uses the ratio between  $E[T_{EXT}(r, l)]$  and  $E[T_{INT}(r, l)]$  to broadly classify the network state as *collision dominated* or *noise dominated*. It is important to note that the focus of this classification is not to differentiate between individual frame losses, but to provide an indication about dominant factors, which can help to guide the long-term rate adaption process. For instance, in collision-dominated environments, probing should be performed less frequently, because a changing rate may have negligible impact on the link-layer performance.

A second enhancement implemented in TARA2 is the suppression of transient drops in the link-layer throughput during the rate adaptation process. More precisely, our experimental results indicate that in collision-dominated environments the link-layer throughput may be subject to high variability due to the exponential behavior of the 802.11 backoff algorithm (see Figs. 2 and 3). Thus, the selection of the probing rate should take into account both the average and the standard deviation of the link-layer throughput estimates.

**Algorithm 2.** Pseudo-code of TARA2 operations for each new data packet to transmit.

**Input:** Set  $\mathcal{R}$  of available transmission rates,  
 Measured  $E[T_{EXT}(r, l)]$  and  $E[T_{INT}(r, l)]$  for each transmission rate  $r \in \mathcal{R}$  and packet size  $l$ ,  
 Table of  $G(r, \tilde{r}, l)$  values for each pair of transmission rates  $r$  and  $\tilde{r}$  (with  $r, \tilde{r} \in \mathcal{R}$ ).  
**Result:** Modulation rate  $r_{new}$  for the new packet transmission.

```

1 begin
2    $r_{prev} \leftarrow$  rate used for the previous packet transmission;
3    $\mathcal{R} \leftarrow \emptyset$ ;
4   if  $\left(\frac{E[T_{EXT}(r_{prev}, l)]}{E[T_{INT}(r_{prev}, l)]} < 1\right)$  then Noise_Dominated  $\leftarrow true$ ;
5   else Noise_Dominated  $\leftarrow false$ 
6   if (Probe_Timer is expired) then
7     reset Probe_Timer;
8     for all element  $r$  of set  $\mathcal{R}$  do
9       if (Noise_Dominated= $true$ ) and  $G(r_{prev}, r, l) > 1$  then  $\tilde{\mathcal{R}} \leftarrow \tilde{\mathcal{R}} \cup r$ ;
10      else If (Noise_Dominated= $false$ ) and  $G(r_{prev}, r, l) > (1 + cov[\gamma(r_{prev}, l)])$  then
11         $\tilde{\mathcal{R}} \leftarrow \tilde{\mathcal{R}} \cup r$ 
12      end
13    end
14  if ( $\tilde{\mathcal{R}}$ ) then  $r_{new} \leftarrow$  random element in set  $\tilde{\mathcal{R}}$ ;
15  else  $r_{new} \leftarrow$  FindBestRate2 (MAX_SUCC_FAIL );
16 end
  
```

For ease of explanation, the pseudo-code of the TARA2 scheme is illustrated in Algorithm 2. If the network is in the noise-dominated state (i.e.,  $E[T_{EXT}(r, l)]/E[T_{INT}(r, l)] < 1$ ), TARA2 selects the set of probing rates as in TARA1 (line 9). Otherwise, a rate is eligible to be probed only if its gain factor compensates the coefficient of variation of link-layer throughput estimates, say  $cov[\gamma(r_{prev}, l)]$  (line 10), which is a measure of the *relative dispersion* of the throughput measurements. We recall that the coefficient of variation of a random variable is a dimensionless parameter computed as the ratio between the standard deviation and the mean of the random variable. Therefore, condition  $G(r_{prev}, r, l) > (1 + cov[\gamma(r_{prev}, l)])$  means that rate  $r$  ensures a potential gain factor sufficiently high to counterbalance the short-term throughput variability caused by the exponential 802.11 backoff algorithm. Finally, the TARA2 scheme uses a modified FindBestRate2() function. More precisely, if Noise\_Dominated = *true*, FindBestRate2() and FindBestRate1() perform identically. In contrast, if Noise\_Dominated = *false*, then the best transmission rate  $r^*$  is chosen so that  $r^* = \operatorname{argmax}_{r \in \mathcal{R}} [\gamma(r, l) + \sigma(\gamma(r, l))]$ , where  $\sigma(x)$  is the standard deviation of the r.v.  $x$ . In other words, FindBestRate2() can prefer rate  $r_1$  to rate  $r_2$  even if the first one gives a lower average link-layer throughput, if the throughput standard deviation compensates this difference.

#### 4. Experimental evaluation

To demonstrate the effectiveness of the TARA schemes we have conducted experiments across a variety of network conditions and traffic patterns, and we have compared the performance of TARA to that of three existing rate selection algorithms provided with the Madwi driver: AMRR [6], ONOE [7], and SampleRate [8]. It is important to note that the Madwifi driver is the most commonly used open-source driver for Atheros-based cards.

Regarding the experimental methodology, we repeated each test ten times and we computed the 95% confidence intervals in addition to the average values to verify the statistical validity of our measurements. When more controlled experiments were needed (e.g., to investigate the influence of medium contention levels on the performance of rate adaptation algorithms) we conducted our measurement studies during night hours, to minimize the impact of external factors.

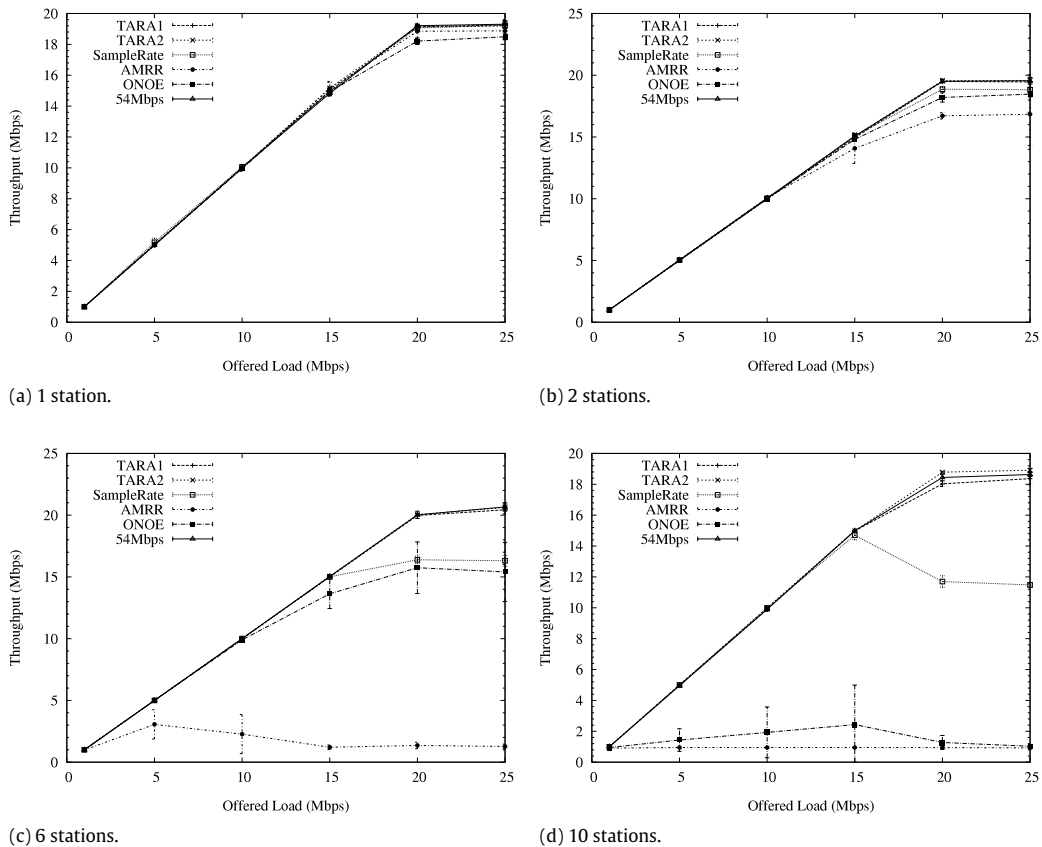


Fig. 6. Aggregate throughput versus the total offered load for the homogeneous WLAN case.

#### 4.1. Collision-dominated environment

The goal of this set of experiments is to compare the performance of the rate adaptation algorithms with increasing network contention. To this end, we use the same WLAN described in Section 3.1.1, where the stations are located in close proximity of the AP to ensure that they have similar channel conditions. In our test, each station activates a constant bit-rate (CNR) flow towards the AP using UDP as the transport layer. All the flows generate the same load, and every 60 s the total offered load is incremented by 1 Mbps until the network becomes saturated. For each traffic pattern we have tested, in turn, TARA1, TARA2, SampleRate, AMRR, and ONOE, and the 12 fixed transmission rates supported by the 802.11g physical layer (from the highest to the lowest). Each experiment run lasts three minutes, thus the entire trial requires more than five hours to be completed. The same trial is repeated ten times and for different network sizes (i.e., number of active stations).

Fig. 6 reports the aggregated UDP throughput versus the total offered load for four representative network sizes. We tested all the 12 fixed transmission rates supported by the 802.11g physical layer, and the maximum throughput was obtained using the 54 Mbps data rate in all the considered network scenarios. For this reason, in Fig. 6 we only show results related to the 54 Mbps data rate and not to the other fixed transmission rates. First, let us consider the case of a high offered load equal to 25 Mbps, which is sufficient to fully utilize the channel capacity. Fig. 6(a) reports the experimental results related to a WLAN with a single active station. In this scenario, the TARA1, TARA2, and SampleRate algorithms approach very closely the maximum throughput, while AMRR introduces a throughput degradation of 3%, and the ONOE scheme is the worst one, with about 6% reduction in link performance. To explain these results, we can observe in Fig. 7(a) that about 99% of frames are successfully transmitted at the first attempt using the 54 Mbps data rate. When we reduce the transmission rate the aggregate throughput decreases because the efficiency of the 802.11 MAC protocol decreases and, of course, it is not compensated by a reduction of frame loss rates.

As shown in Fig. 7(b), both TARA1 and TARA2 algorithms always select 54 Mbps as the best transmission rate, while SampleRate uses 48 Mbps for about 6% of total transmitted frames. The explanation of this behavior is that SampleRate estimates the expected packet transmission time without considering busy periods introduced by external interference sources, or transmissions from hidden terminals. As a consequence, the SampleRate algorithm underestimates the packet transmission times. For instance, in this configuration the lossless packet transmission time at 48 Mbps is assumed to be shorter than the estimated packet transmission times at 54 Mbps. For this reason, the sender periodically transmits probe

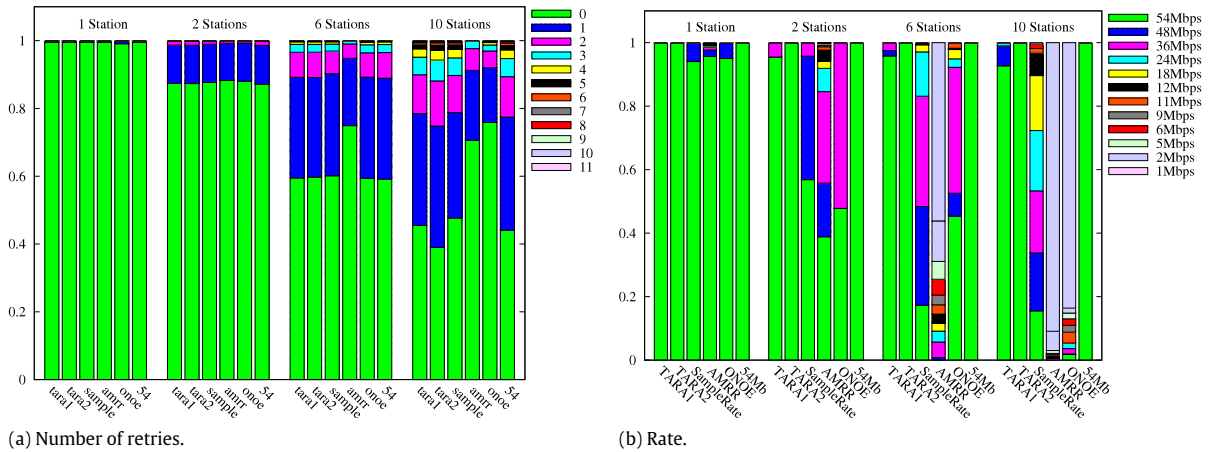


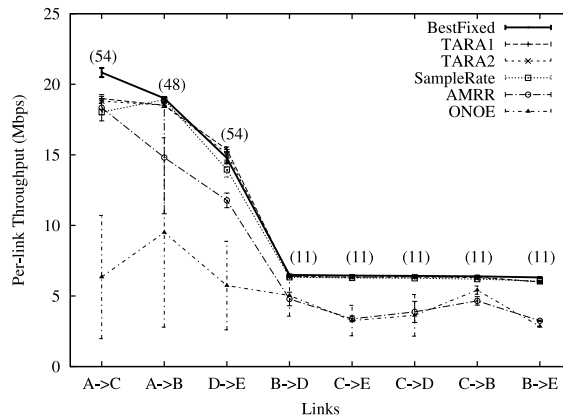
Fig. 7. Distributions (%) of number of retries and data rates in the homogeneous WLAN case for a total offered load equal to 25 Mbps.

frames at 48 Mbps. In contrast, by examining the rate distribution of the AMRR scheme we can observe that AMRR uses 54 Mbps for about 95% of the total transmitted frames. Indeed, AMRR reduces the transmission rate of the first transmission attempt only if the frame loss rate is higher than 33%. Thus, AMRR keeps using 54 Mbps as the most preferred transmission rate. Nevertheless, AMRR obtains a lower throughput than SampleRate. This is due to the fact that AMRR adopts a multiple rate retry strategy with small retry limits (all retry limits are set to 1). This implies that AMRR occasionally tries very low transmission rates. Finally, ONOE performs slightly worse than AMRR although it sends 97% of frames at 54 Mbps. This can be explained by observing that ONOE is less aggressive than AMRR in rate increasing, and its credit-based scheme can introduce large delays before again raising the transmission rate from lower rates.

Let us now consider a network scenario with ten stations sending packets concurrently, thus generating high contention in the network. During the performed tests, we observed that the maximum link-layer throughput is again obtained when transmitting at 54 Mbps. However, due to the high collision probability, only 44% of transmitted frames are received successfully at the first transmission attempt, and the frame loss rate is about 66% (see Fig. 7(a)). SampleRate evaluates the expected transmission time of a frame at a given rate using formula (3), which does not account for channel occupations due to transmissions from other stations present in the WLAN. As a consequence, SampleRate overestimates the maximum throughput achievable at the different transmission rates, leading to a conservative rate selection. Thus, SampleRate sends only 15% of transmitted frames at 54 Mbps. This inaccurate rate selection causes a throughput reduction of 49% over the best performance. In contrast, both TARA1 and TARA2 work very well, with the latter closely approaching the maximum throughput performance. More precisely, the TARA1 scheme sends 92% of transmitted frames at 54 Mbps, while TARA2 reaches a surprising 98%. The explanation of the slightly better performance of TARA2 over TARA1 is rooted in the ability of the former scheme to compensate the throughput variability induced by the 802.11 exponential backoff algorithm. Finally, both AMRR and ONOE perform remarkably worse than the other schemes, obtaining a throughput that is more than an order of magnitude lower than that measured using SampleRate. The explanation of this inefficiency is that the rate downgrade policies in both AMRR and ONOE operate to keep the frame loss rate below pre-determined and fixed thresholds (namely, 33% for AMRR and 50% for ONE). However, in the 802.11 MAC protocol collisions are needed to increase the backoff window size, thus achieving a time spreading of the transmission attempts that is adequate for the current congestion level. As proven in [33], an optimal collision rate exists, depending on the number of competing stations, that maximizes the channel capacity. In contrast, both AMRR and ONOE try to enforce arbitrary collision rates that can lead to operating conditions far from the optimal ones. Note that AMRR shows the worst performance because it permits only four attempts per frame.

In conditions of low–mid contention (namely, non-saturating offered load and/or low number of active stations) TARA1 and TARA2 are still the best rate adaptation schemes. By inspecting packet traces we observed that SampleRate starts to experience throughput degradations as soon as the probability of successful transmission at the first transmission attempt gets lower than 60%. It is interesting to note that AMRR shows throughput performance similar to ONOE with two active stations and ten active stations, while it is significantly worse with six active station. This steep degradation of the performance obtained with AMRR is caused by the fact that AMRR uses a lower loss ratio threshold to trigger the transmission rate downgrade than ONOE. However, ONOE's performance is affected by a significant variability, as shown by the extent of the confidence intervals, while the other schemes show a more stable behavior.

In conclusion, our experimental results show that SampleRate is a rate adaptation algorithm more robust to frame losses due to collisions than loss-ratio threshold-based schemes. However, our proposed TARA schemes significantly outperform all the other rate adaption protocols. This is achieved using direct and accurate measurements of the link-layer throughput for the rate selection. In this way the sender can avoid biased rate downgrades caused by the underestimation of the network contention level.



**Fig. 8.** Per-link throughput for the experimental mesh network under various rate adaptation schemes. The numbers in parentheses are the fixed transmission rates providing the best performance.

#### 4.2. Noise-dominated environment

The aim of this section is to investigate how the rate adaptation algorithms studied in Section 4.1 perform in a *noise-dominated* environment, with lossy but non-congested links. This is the case of an outdoor mesh test-bed, at least when only one link is active at a time, as explained in more detail in the following.

There are detailed performance studies of 802.11 links in community-based mesh networks [27,34]. These studies have shown that frame error rates on outdoor 802.11 links can be highly variable due to multi-path fading and external interference. It is evident that the performance of a rate adaptation algorithm is closely connected to the link-level loss characteristics affecting each transmission rate. Therefore, we have conducted a preliminary set of experiments to better understand the relation between the frame error rate due to channel errors and the various transmission rates in our mesh test-bed. The results of these tests have been reported in our previous paper [15], and here we only summarize the main findings. The first important observation is that the frame loss rates are not uniform over all transmission rates, but they can be grouped into three categories. Generally, there is a group of transmission rates not affected by channel errors, which experience negligible frame loss rates. Then, we may have a group of rates that do not work at all, because most of the transmitted frames are lost. Finally, we may have a group of transmission rates with intermediate frame loss rates, which are probably due to fading or interference rather than simple channel attenuation. Note that this classification of link behaviors is similar to the one originally proposed in [8].

A second important observation concerns that difference reliability of 802.11g technology compared to 802.11b. More precisely, the highest 802.11g rates are the most vulnerable to the channel conditions, although the extent of this vulnerability is highly variable and depends on the link characteristics. This finding is consistent with the results of other experimental studies conducted on outdoor mesh networks [27,34,35]. Finally, we have analyzed the correlations between frame losses at different time scales. Our results indicate that the transmission rates with negligible frame loss ratio are also quite stable, and observation periods of 1 s give reliable estimates of the long-term frame loss rate. This behavior is typical of links affected by almost independent channel errors. Note that these trends are consistent with the results reported in [27], where the authors noted that non-bursty links are predominant in urban 802.11-based mesh networks.

To compare TARA's performance against the other three algorithms (i.e., AMRR, ONOE and SampleRate) we have conducted a set of tests where each mesh router in turn sends UDP packets to one of its neighbours as fast as it can. For each link, we have tested, in turn, TARA1, TARA2, SampleRate, AMRR, and ONOE, and the 12 fixed transmission rates supported by the 802.11g physical layer (from the highest to the lowest). In Fig. 8 we report the average throughput and the 95% confidence intervals for eight representative links. In the figure, the curve labelled with "BestFixed" shows the maximum throughput obtained using a fixed transmission rate, and in the parentheses we specify the data rate used. From the results we can note that in our network the links with the highest capacity are the two directional links (i.e.,  $A \rightarrow B$  and  $A \rightarrow C$ ), and the shortest omni-directional link (i.e.,  $D \rightarrow E$ ). According to the link taxonomy proposed in [8], these links are gradual links because they guarantee a graceful throughput degradation when decreasing the bit-rate. Hence, the best transmission rates are the highest ones, either 54 Mbps or 48 Mbps. In contrast, the other omni-directional links reported in Fig. 8 show a drastic throughput drop for transmission rates higher than 11 Mbps, when they almost stop working altogether. However, the 11 Mbps data rate, which is obtained using 802.11b modulation schemes, is very robust against noise, and it is almost unaffected by channel errors.

As shown in Fig. 8, TARA1, TARA2, and SampleRate now have similar performance, and they closely approximate the best link performance. However, the TARA schemes slightly outperform SampleRate on all the shown links, except for link  $A \rightarrow B$ . As expected, there are no appreciable differences between the performance of TARA1 and TARA2. Regarding AMRR, it works reasonably well in both gradual and steep links, but it is worse than SampleRate and both TARA schemes. Two main reasons can be identified. Firstly, AMRR permits only four consecutive retries for each frame, and this induces a higher packet

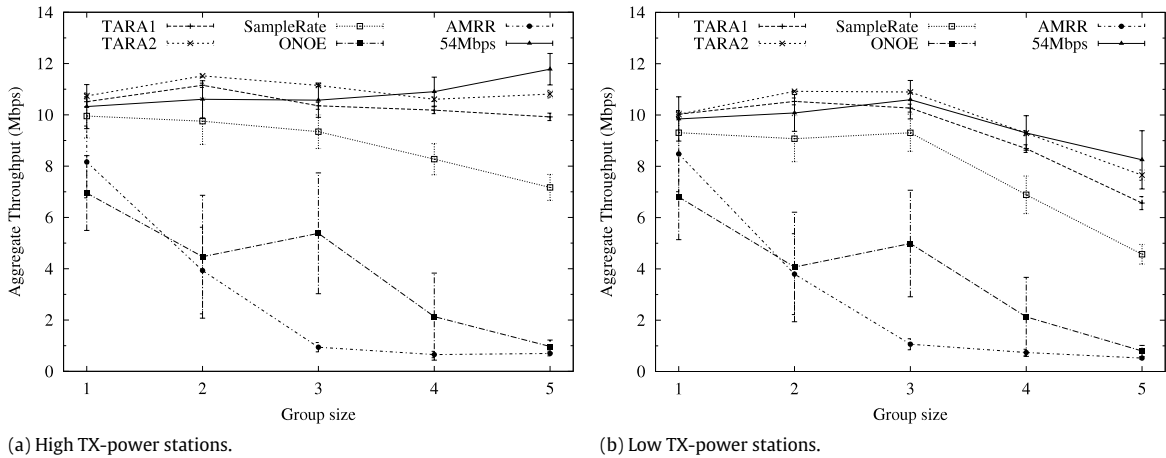


Fig. 9. Aggregate throughput for a heterogeneous WLAN versus the number of active stations per group.

loss. In addition, AMRR occasionally tries a very low transmission rate due to the specific parameter setting of its multiple rate retry strategy, that permits only one transmission attempt at the best transmission rate. On the other hand, ONOE works reasonably well for steep links (even better than AMRR), but is the worst for gradual links. This is due to the credit-based rate upgrade policy implemented by ONOE, which makes this algorithm very slow in increasing the transmission rate. Thus, when ONOE selects an intermediate transmission rate affected by moderate frame loss rates, which is more common for gradual links than steep links, it keeps using that bit-rate for several seconds, even if the channel conditions are improved.

Our experimental results somehow confirm the observations made in [9,26]. More precisely, the results reported in [9,26] suggested that SampleRate performs relatively better than ONOE and AMRR in noise-dominated environments, at least for settings with stable channel conditions. However, the conclusions in both [9] and [26] were based on results obtained from an indoor wireless network, while here we are considering outdoor mesh networks. In addition, our tests confirm that in environments where channel errors and external interference are presumably the dominant factors for frame loss, the two TARA variants perform similarly to the best of the other rate adaptation algorithms implemented in the Madwifi driver.

#### 4.3. Mixed environment

In the previous sections we have considered two contrasting scenarios. The former is a homogeneous WLAN scenario representing a collision-dominated environment, while the latter is an outdoor mesh network scenario representing a noise-dominated environment. It is important to recall that these are real-world environments, which are not fully controllable. On the one hand, this complicates the analysis and interpretation of the results. On the other hand, it makes it impossible to completely exclude either collision losses or channel errors. In this section, we want to investigate a more complex network scenario where both collisions and channel errors are equally important. To this end, we use an *heterogeneous* WLAN. In this network scenario the static WLAN clients are divided into two groups of equal size  $n$ . Then, stations in the first group (hereafter referred to as high TX-power stations) employ a full transmission power, while stations in the other group (hereafter referred to as low TX-power stations) use a reduced transmission power. Therefore, stations from different groups are connected to the AP with links of different quality. For instance, the percentage of packets transmitted at 54 Mbps that are successfully received by the AP ranges from 90% (for  $n = 1$ ) to 62% (for  $n = 5$ ) for high TX-power stations, while it ranges from 86% (for  $n = 1$ ) to 46% (for  $n = 5$ ) for low TX-power stations. These differences in retransmission probabilities are due to the fact that the low TX-power stations are more prone to channel errors than the high TX-power stations. In addition, this heterogeneous WLAN suffers from the same collision-induced frame losses as the homogeneous WLAN studied in Section 4.1.

In our tests each client sends as many UDP packets as possible to the AP. Every 60 s a new client is activated. Fig. 9 summarizes the throughput results. For completeness, we show the aggregate throughput obtained by each of the two groups. Note that it is important to differentiate the rate decisions of high TX-power stations and low TX-power stations because it is well known that *performance anomalies* can occur if the stations in a WLAN use different transmission rates [36]. More precisely, the frame transmissions of stations using low bit-rates last longer than the frame transmission of stations using high bit-rates. This implies that slow stations considerably limit the throughput of other stations roughly to the level of the lowest rate used. The experimental results confirm that in all the settings considered, both TARA1 and TARA2 significantly outperform the other rate adaptation algorithms, with TARA2 showing better performance than TARA1. This can be explained by observing that TARA2 consistently selects higher transmission rates than all the other rate adaptation algorithms. However, when  $n > 3$ , TARA2 also starts to select lower transmission rates, reducing the overall network

efficiency. For these reasons, when compared to a fixed rate strategy, a transmission rate of 54 Mbps provides slightly higher throughput than TARA1 and TARA2 for  $n > 3$ .

In contrast, most of the time both AMRR and ONOE select very low transmission rates, especially when the network contention level is high. In our tests, both AMRR and ONOE almost never selected the maximum rate. This is in line with the behaviors observed in the homogeneous WLAN scenario for increasing number of active stations. By inspecting the packet traces we observed that the use of low bit-rates contributes to reducing the retransmission probabilities, because fewer frames are lost due to channel errors. However, this small reduction in frame loss rates is not enough to compensate the degradation of throughput performance. This explains the extremely poor performance obtained by these two rate adaptation algorithms. Regarding SampleRate, it works reasonably well, and the lower throughput performance is due to its inability to correctly deal with the frame losses due to collisions.

## 5. Conclusions

In this study we have designed a novel technique to estimate the per-packet transmission time in 802.11-based wireless networks, which is based on accurate, but easy to obtain, statistics of transmission events. We have shown that this method can be used to accurately estimate the link-layer throughput, as well as to simply classify the network state in order to identify highly congested networks. Then, we have employed these estimates to design and implement a novel rate adaptation algorithm called TARA. The key feature of the TARA scheme is to mitigate the negative impact of collisions on the operations of rate adaptation, without relying on costly RTS/CTS probing for loss differentiation, or changes in legacy MAC protocol. Experimental studies conducted in diverse environments, both indoor and outdoor, and accounting for varying number of active users and interference levels, have shown that TARA significantly outperforms other rate adaptation algorithms currently implemented in commercial cards.

We believe that the results in this paper suggest several possible avenues for future research. For instance, in this study we have used the ratio between  $E[T_{EXT}]$  and  $E[T_{INT}]$  for classifying the network state. More sophisticated strategies might be used exploiting historical information to reveal trends or patterns in these two metrics. This can permit a more accurate categorization of network state. In addition, to extract average statistics from the collected measurements we used the classical EWMA filter. However, in practical network scenarios different stations may have significantly different packet arrival rates. For this reason, it is worth investigating alternative averaging techniques similar to those proposed in [26] to take into account the timing of the measurements, in order to give higher weights to recent samples. Another future direction we will pursue is the design of more efficient methods for collecting statistics at different data rates. More precisely, using a small number of probe packets may result in statistically inaccurate estimates, and it could be preferable to collect measurements over short time windows. In this case the trade-off would be between improved accuracy and increased overheads.

## Acknowledgments

This work was supported in part by the European Commission through FP7 project EU-MESH, ICT-215320, and by the Italian National Project PRIN-WORLD, under grant number 2007R989S.

## References

- [1] IEEE WG 802.11, 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 (June 12 2007).
- [2] T. Paul, T. Ogunfunmi, Wireless LAN comes of age: Understanding the IEEE 802.11n Amendment, *IEEE Circuits and Systems Magazine* 8 (1) (2008) 28–54.
- [3] R. Bruno, M. Conti, E. Gregori, Mesh networks: Commodity multihop ad hoc networks, *IEEE Communications Magazine* 43 (3) (2005) 123–131.
- [4] I. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: A survey, *Computer Networks* 47 (4) (2005) 445–487.
- [5] A. Kamerman, L. Monteban, WaveLAN-II: A High-performance wireless LAN for the unlicensed band, *Bella Lab Technical Journal* (Summer) (1997) 118–133.
- [6] M. Lacage, M.H. Manshaei, T. Turletti, IEEE 802.11 rate adaptation: A practical approach, in: *Proc. of IEEE MSWiM '04*, Venice, Italy, 2004, pp. 126–134.
- [7] MadWifi driver documentation, Onoe Rate Control. <http://madwifi.org/wiki/UserDocs/RateControl>.
- [8] J. Bicket, Bit-rate selection in wireless networks, Massachusetts Institute of Technology, M.S. Thesis, February, 2005.
- [9] S. Wong, H. Yang, S. Lu, V. Bharghavan, Robust rate adaptation for 802.11 wireless networks, in: *Proc. of ACM MOBICOM'06*, Los Angeles, CA, USA, September 24–29, 2006, pp. 146–157.
- [10] P. Acharya, A. Sharma, E. Belding, K. Almeroth, K. Papagiannaki, A contention-aware routing metric for multi-rate multi-radio mesh networks, in: *Proc. of IEEE SECON'08*, San Francisco, CA, USA, 2008, pp. 1–9.
- [11] S. Wang, A. Helmy, BEWARE: Background traffic-aware rate adaptation for IEEE 802.11 MAC, in: *Proc. of IEEE WoWMoM'08*, New Port Beach, CA, USA, June 23–27, 2008, pp. 7–14.
- [12] J. Choi, J. Na, Y.-S. Lim, K. Park, C.-K. Kim, Collision-aware design of rate adaptation for multi-rate 802.11 WLANs, *IEEE Journal on Selected Areas in Communications* 26 (8) (2008) 1366–1375.
- [13] S. Pal, S. Kundu, K. Basu, S. Das, IEEE 802.11 rate control algorithms: Experimentation and performance evaluation in infrastructure Mode, in: *Proc. of PAM'06*, Adelaide, Australia, March 30–31, 2006.
- [14] K. Ramachandran, H. Kremling, M. Gruteser, P. Spasojevic, I. Seskar, Experimental scalability analysis of rate adaptation techniques in congested 802.11 networks, in: *Proc. of IEEE WoWMoM'07*, Helsinki, Finland, June 18–21, 2007, pp. 1–12.
- [15] E. Ancillotti, R. Bruno, M. Conti, Experimentation and performance evaluation of rate adaptation algorithms in wireless mesh networks, in: *Proc. of ACM PE-WASUN'08*, Vancouver, Canada, 2008, pp. 7–14.

- [16] J. Camp, E. Knightly, Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation, in: Proc. of ACM MOBICOM'08, San Francisco, CA, USA, 2008, pp. 315–326.
- [17] J. Kim, S. Kim, S. Choi, D. Qiao, CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs, in: Proc. of IEEE INFOCOM'06, Barcelona, Spain, April 23–29, 2006, pp. 1–11.
- [18] Q. Pang, V. Leung, S. Liew, A rate adaptation algorithm for IEEE 802.11 WLANs based on MAC-layer loss differentiation, in: Proc. of BROADNETS'05, Boston, Massachusetts, USA, October 3–5, 2005, pp. 709–717.
- [19] J. Yun, S. Seo, Novel collision detection scheme and its applications for IEEE 802.11 wireless LANs, *Computer Communications* 30 (6) (2007) 1350–1366.
- [20] S. Biaz, S. Wu, Loss differentiated rate adaptation for 802.11 wireless networks, in: Proc. of IEEE WCNC'08, Las Vegas, NV, USA, 2008, pp. 1639–1644.
- [21] MADWIFI Driver Specification, Multiband Atheros Driver For WIFI, <http://madwifi.org/>.
- [22] D. Qiao, S. Choi, K. Shin, Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs, *IEEE Transactions on Mobile Computing* 1 (4) (2002) 278–292.
- [23] G. Holland, N. Vaidya, P. Bahl, A rate-adaptive MAC protocol for multi-hop wireless networks, in: Proc. ACM MOBICOM'01, Rome, Italy, 2001, pp. 236–251.
- [24] B. Sadeghi, V. Kanodia, A. Sabharwal, E. Knightly, Opportunistic media access for multirate ad hoc networks, in: Proc. ACM MOBICOM'02, Atlanta, Georgia, USA, September 23–28, 2002, pp. 24–35.
- [25] J. Pavon, S. Choi, Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement, in: Proc. of IEEE ICC'03, vol. 2, Seattle, WA, USA, May 20–30, 2003, pp. 1108–1113.
- [26] G. Judd, X. Wang, P. Steenkiste, Efficient channel-aware rate adaptation in dynamic environments, in: Proc. of ACM MOBISYS'08, Breckenridge, CO, USA, 2008, pp. 118–131.
- [27] D. Aguayo, J. Bicket, S. Biswas, G. Judd, R. Morris, Link-level measurements from an 802.11b mesh network, *SIGCOMM Computer Communication Review* 34 (4) (2004) 121–132.
- [28] J. Bicket, D. Aguayo, S. Biswas, R. Morris, Architecture and evaluation of an unplanned 802.11b mesh network, in: Proc. of ACM MOBICOM'05, Cologne, Germany, 2005, pp. 31–42.
- [29] M. Heusse, F. Rousseau, R. Guillier, A. Duda, Idle sense: An optimal access method for high throughput and fairness in rate diverse Wireless LANs, in: Proc. of ACM SIGCOMM'05, Philadelphia, PA, USA, 2005, pp. 121–132.
- [30] A. Jardosh, K. Ramachandran, K. Almeroth, E. Belding, Understanding congestion in IEEE802.11b wireless networks, in: Proc. of IMC'05, Berkeley, CA, USA, 2005.
- [31] C. Koksali, H. Kassab, H. Balakrishnan, An analysis of short-term fairness in wireless media access protocols, *SIGMETRICS Performance Evaluation Review* 28 (1) (2000) 118–119.
- [32] Z. Li, S. Nandi, A. Gupta, Modeling the short-term unfairness of IEEE 802.11 in presence of hidden terminals, *Performance Evaluation* 63 (4) (2006) 441–462.
- [33] L. Bononi, M. Conti, E. Gregori, Runtime optimization of IEEE 802.11 Wireless LAN Performance, *IEEE Transactions on Parallel Distributed Systems* 15 (1) (2004) 66–80.
- [34] J. Camp, J. Robinson, C. Steger, E. Knightly, Measurement driven deployment of a two-tier urban mesh access network, in: Proc. of ACM MOBISYS'06, Uppsala, Sweden, 2006, pp. 96–109.
- [35] G. Bianchi, F. Formisano, D. Gustiniano, 802.11b/g link level measurements for an outdoor wireless campus network, in: Proc. of IEEE WOWMOM'06, New York, NY, USA, 2006, pp. 525–530.
- [36] M. Heusse, F. Rousseau, G. Berger-Sabbatel, A. Duda, Performance anomaly of 802.11b, in: Proc. IEEE INFOCOM 2003, vol. 2, San Francisco, USA, 2003, pp. 836–843.



**Emilio Ancillotti** received the Laurea degree in Computer Engineering, and the Ph.D. degree in Information Engineering, both from the University of Pisa, in 2003 and 2007, respectively. From 2003 to 2006 he worked as the teaching assistant at the Department of Information Engineering of the University of Pisa. Since 2007, he has been a researcher at IIT, an institute of the Italian National Research Council (CNR). His current research interests are in the area of wireless and mobile networks with an emphasis on pervasive computing, ad hoc networks, mesh networks and performance evaluation.



**Raffaele Bruno** is a researcher at IIT, an institute of the Italian National Research Council (CNR). He received a Ph.D. in computer engineering and the Laurea in telecommunication engineering, both from the University of Pisa, Italy, in 2003 and 1999, respectively. His main research interests include the design, modelling and performance evaluation of MAC, routing and transport protocols for wireless networks. He published more than 30 papers in international journals, conferences and workshops. He served and is currently serving on the TPC of several international conferences and workshops. He was Workshop Co-Chair of *IEEE PerSeNS 2006*, *IEEE MASS- GHS07* and *IEEE HotMESH'09*. He served as guest editor for a Special Issue of the *Pervasive and Mobile Computing (PMC) Journal on Homeland and Global Security*, and is currently serving as editor of a Fast Track on *Computer Communication Journal*.



**Marco Conti** is a research director at IIT, an institute of the Italian National Research Council (CNR). He published in journals and conference proceedings more than 200 research papers related to design, modeling, and performance evaluation of computer-network architectures and protocols. He co-authored the book "Metropolitan Area Networks" (1997) and is co-editor of the books "Mobile Ad Hoc Networking" (2004) and "Mobile Ad Hoc Networks: From Theory to Reality" (2007). He is the chair of the IFIP WG 6.3 "Performance of Communication Systems". He is the Editor-in-chief of the *Computer Communications* Journal and Associate Editor-in-chief of *Pervasive and Mobile Computing* Journal; he is on the editorial board of: *IEEE Transactions on Mobile Computing*, *Ad Hoc Networks*, *Journal of Communications Systems*, and *Wireless Ad Hoc and Sensor Networks: An International Journal*. He served as TPC chair of *IEEE PerCom 2006*, and of the IFIPTC6 Conferences *Networking 2002* and *PWC 2003*, and as TPC co-chair of *ACM WoWMoM 2002*, *WiOpt '04*, *IEEE WoWMoM 2005*, and *ACM MobiHoc 2006*. He served as general chair of *ACM REALMAN 2006* and *IEEE MASS 2007*, and as general co-chair of *IEEE WoWMoM 2006*, and *ACM MobiOpp 2007*. He is currently serving as general co-chair of *IEEE PerCom 2010* and as program co-chair of *ACM MobiOpp 2010*.