Workshop on Computer Networks

Occupancy Estimation in the IEEE 802.11 Distributed Coordination Function

Luigi Alcuri[1]
luigi.alcuri@tti.unipa.it
tel. 091 6566250 fax. 091 488452

Giuseppe Bianchi[1]
bianchi@elet.polimi.it
tel. 091 6566247 fax. 091 488452

Ilenia Tinnirello[1]
ileniati@tin.it
tel. 091 6566269 fax. 091 488452

Contact Author: Ilenia Tinnirello

[1]University of Palermo, Department of Electrical Engineering
Viale delle Scienze, 90128, Palermo, Italy

Abstract

IEEE 802.11 employs Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as primary Medium Access Control (MAC) scheme, namely the Distributed Coordination Function (DCF). Throughput performance of DCF is very sensitive to the number of competing stations. In this paper, a methodology to estimate this number from run-time measurements is proposed. The explicit knowledge of the occupancy status of an 802.11 Wireless network has several practical implications, as it can dynamically drive switching algorithms from Basic to RTS/CTS access model, as well as cell selection algorithms employing load balancing.

Keywords

Wireless Local Area Networks, Occupancy Estimation, Load Balancing

I. INTRODUCTION

The IEEE 802.11 standard [1] provides a detailed Medium Access Control and Physical layer specification for Wireless Local Area Networks (WLANs). In 802.11, the primary mechanism to access the medium is called *Distributed Coordination Function* (DCF). This is a random access scheme, based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. Retransmission of collided packets is managed according to binary exponential backoff rules.

In DCF, a packet can be transmitted according to two different "modes". The Basic Access is a two-way handshaking mechanism characterized by the transmission of an information packet followed by the immediate relay with a positive acknowledgement (ACK). The *Request-To-Send/Clear-To-Send* (RTS/CTS) Access mode is an optional four way handshaking technique, which reserves the channel for each packet transmission by preliminary exchanging two special *Request-To-Send* and *Clear-To-Send* short frames.

Several performance evaluation studies of the IEEE 802.11 DCF [2], [3], [4], [5] have shown that performance are very sensitive to the number of stations competing on the channel, especially when the Basic Access mode is employed. In fact, it has been proven [6], [7] that the backoff window should be made depend on the number of competing stations, if maximum performance are aimed at. However, according to the actual version of the standard [1], the backoff parameters are hard-wired in the PHY layer, and are far from the optimal setting in some network configuration conditions.

This paper deals with the estimation of the number of stations competing for access to the wireless channel. This information cannot be retrieved from the protocol operation. On one side, DCF does not rely on a privileged station to control the access to the channel. But even considering the existence of an Access Point (AP), as in Infrastructured 802.11 Networks, the information available at the AP is limited to the number of "associated" stations, a number which may be very different from the number of competing stations, i.e. stations that are actually in the process of transmitting packets.

The contribution of the paper consists in two different aspects. First, we obtain a relation between

PHY	Slot Time (σ)	$CW_{\min}$	$CW_{\max}$
FHSS	50 μs	16	1024
DSSS	20 μs	32	1024
IR	8 μs	64	1024

TABLE I

SLOT TIME, MINIMUM, AND MAXIMUM CONTENTION WINDOW VALUES FOR THE THREE PHY SPECIFIED BY THE 802.11 STANDARD: FREQUENCY HOPPING SPREAD SPECTRUM (FHSS), DIRECT SEQUENCE SPREAD SPECTRUM (DSSS), INFRARED (IR)

the number N of competing stations and performance figures that can be measured from the monitoring of the channel, with simple extensions of the analysis presented in [4]. Second, we provide a thorough performance and robusteness evaluation of our estimation rule, via simulation.

The rest of the paper is organized as follows. In section II we briefly review the Distributed Coordination Function of IEEE 802.11. In section III, we derive our estimation formula. Simulation results are given in section IV. In section V we briefly present two possible practical applications of the estimation of the number of competing stations. Finally, concluding remarks are given in Section VI.

II. 802.11 DISTRIBUTED COORDINATION FUNCTION

The IEEE 802.11 Distributed Coordination Function (DCF) is briefly summarized as follows. A station with a new packet to transmit monitors the channel activity. If the channel is idle for a period of time equal to a Distributed InterFrame Space (DIFS), the station transmits. Otherwise, if the channel is sensed busy (either immediately or during the DIFS), the station persists to monitor the channel until is measured idle for a DIFS. At this point, the station generates a random backoff interval before transmitting (this is the Collision Avoidance feature of the protocol), to minimize the probability of collision with packets being transmitted by other stations. In addition, to avoid channel capture, a station must wait a random backoff time between two consecutive new packet transmissions, even if the medium is sensed idle in the DIFS time[1].

For efficiency reasons, DCF employs a discrete-time backoff scale. The time immediately following an idle DIFS is slotted, and a station is allowed to transmit only at the beginning of each *Slot Time*.

DCF adopts an exponential backoff scheme. At each packet transmission, the backoff time is uniformly chosen in the range $(0, w-1)$. The value w is called *Contention Window*, and depends on the number of transmissions failed for the packet. At the first transmission attempt, w is set equal to a value $CW_{\min}$ called minimum contention window. After each unsuccessful transmission, w is doubled, up to a maximum value $CW_{\max} = 2^m CW_{\min}$. The values $CW_{\min}$ and $CW_{\max}$ reported in the final version of the standard [1] are PHY-specific and are summarized in table I.

The backoff time counter is decremented as long as the channel is sensed idle, "frozen" when a transmission is detected on the channel, and reactivated when the channel is sensed idle again for more than a DIFS. The station transmits when the backoff time reaches 0.

Figure 1 illustrates this operation. Two stations A and B share the same wireless channel. At the end of the packet transmission, station A waits for a DIFS and then chooses a backoff time equal to 9, before transmitting the next packet. We assume that the first packet of station B arrives at the time indicated with an arrow in the figure. After a DIFS, the packet is transmitted. Note that the transmission of packet B occurs during the Slot Time corresponding to a backoff value, for station A, equal to 4. As a consequence of the channel sensed busy, the backoff time is frozen to its value 4, and the backoff counter decrements again only when the channel is sensed idle for a DIFS.

Since the CSMA/CA does not rely on the capability of the stations to detect a collision by hearing

---

[1] As an exception to this rule, the protocol provides a fragmentation mechanism, which allows the MAC to split an MSDU (the packet delivered to the MAC by the higher layers) into more MPDUs (packets delivered by the MAC to the PHY layer), if the MSDU size exceeds the maximum MPDU payload size. The different fragments are then transmitted in sequence, with only a SIFS between them, so that only the first fragment must contend for the channel access.

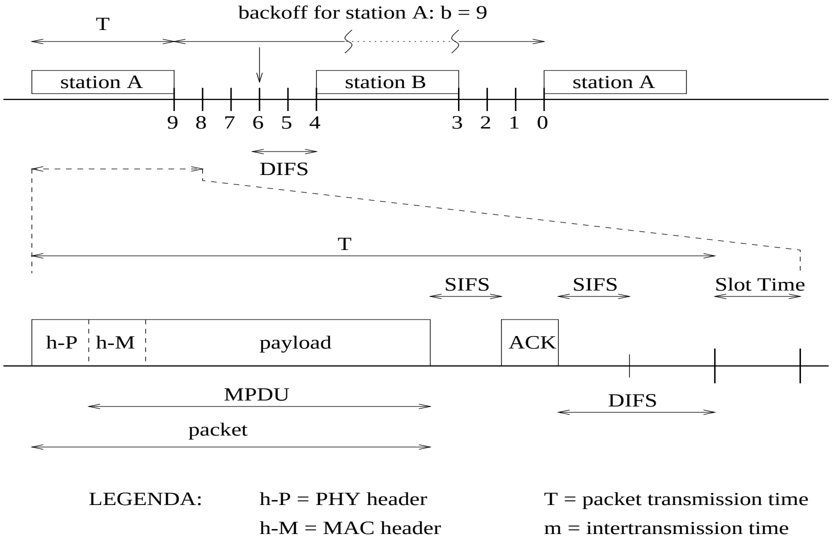

Fig. 1. Example of Basic Access Mechanism

their own transmission, a positive acknowledgement (ACK) is transmitted by the destination station to signal the successful packet reception. The ACK is immediately transmitted at the end of the packet, after a period of time called Short InterFrame Space (SIFS). As the SIFS (plus the propagation delay) is shorter than a DIFS, no other station is able to detect the channel idle for a DIFS until the end of the ACK. If the transmitting station does not receive the ACK within a specified ACK_Timeout, or it detects the transmission of a different packet on the channel, it reschedules the packet transmission according to the given backoff rules.

The above described two-way handshaking technique for the packet transmission is called *Basic Access* mechanism. DCF defines an additional four-way handshaking technique to be optionally used for a packet transmission. This mechanism, known with the name RTS/CTS, is shown in figure 2. A station that wants to transmit a packet, waits until the channel is sensed idle for a DIFS, follows the backoff rules explained above, and then, instead of the packet, preliminarily transmits a special short frame called *Request To Send* (RTS). When the receiving station detects an RTS frame, it responds, after a SIFS, with a *Clear To Send* (CTS) frame. The transmitting station is allowed to transmit its packet only if the CTS frame is correctly received.

The RTS/CTS mechanism provides two fundamental advantages in terms of system performance.

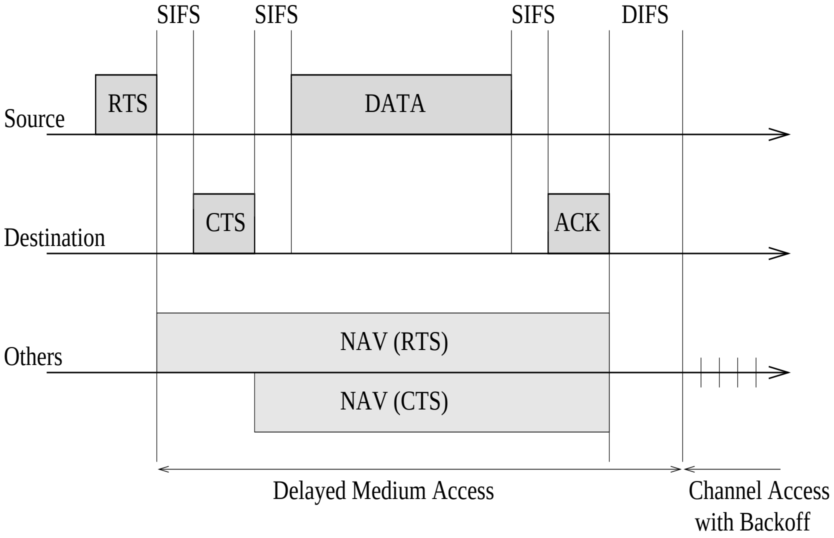

Fig. 2. RTS/CTS Access Mechanism

First, the RTS/CTS mechanism standardized in 802.11 has been specifically designed to combat the so called problem of *Hidden Terminals* [8], which occurs when pairs of mobile stations result to be unable to hear each other. In fact, the frames RTS and CTS explicitly carry, in their payload, the information of the length of the packet to be transmitted. This information can be read by any listening station, which is then able to update a *Network Allocation Vector* (NAV) containing the information of the period of time in which the channel will remain busy. Therefore, when a station is *hidden* from either the transmitting or the receiving station, by detecting just one frame among the RTS and CTS frames, it can suitably delay further transmission, and thus avoid collision. A quantitative investigation of the hidden terminal problem has been specifically considered in [9], [10].

Second, the RTS/CTS is proven to effectively increase the throughput performance even in ideal channel conditions, provided that it is adopted for the transmission of “long” packets (a detailed performance discussion can be found in [4]). If both transmitting stations employ the RTS/CTS mechanism, collision occurs only on the RTS frames, and it is early detected by the transmitting stations by the lack of CTS responses. Since, after the lack of CTS reception, packets are no more transmitted, the duration of a collision is considerably reduced when long packets are involved. The price to pay is a slightly increased transmission overhead (i.e. the RTS/CTS frame exchange) in the case of successful transmissions. To this purpose, the 802.11 standard suggests that the RTS/CTS access mode should be used only when the packet payload exceed a given RTS threshold.

III. ESTIMATION OF THE NUMBER OF ACTIVE STATIONS

In this section, we derive a simple formula that relates the number of active stations to performance figures that can be measured run-time by each station. We define a station to be "active" if it has packets available for transmission in its transmission buffer. A very interesting result is that the proposed estimation approach is independent of the access mode (Basic, RTS/CTS or a combination of the two). Thus, it can be applied to a scenario where each station independently chooses (eventually taking into account the estimation itself, as discussed in section V-A) whether to transmit a packet according to the basic or RTS/CTS access modes.

Our estimation is based on the analytical framework described in [4]. However, while the analysis presented in [4] was targeted to derive the DCF throughput, in this paper the goal is to derive a different result, i.e. the relation between the number of active stations and the probability of a collision seen by a packet being transmitted on the channel (hereafter referred to as *conditional collision probability*). Since the conditional collision probability can be independently measured by each station by simply monitoring the channel activity, it follows that each station can estimate the number of active stations.

We consider a scenario composed of a fixed number n of contending stations, each operating in *saturation* conditions (i.e. the transmission queue of each station is always not-empty). Channel conditions are ideal: no hidden terminal and no capture effect [9] is considered. We assume that, at each transmission attempt, and regardless of the number of retransmissions suffered, the probability p (*conditional collision probability*) that a packet being transmitted on the channel collides, is constant and independent for subsequent transmission attempts. It is intuitive that this approximation results more accurate as long as the contention window and the number of competing stations get larger.

In such assumptions, it has been shown in [4] that is possible to obtain an explicit expression for the probability τ that a station transmits in a randomly chosen slot time as a function of the collision probability p. It is:

$$\tau = \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^m)} \tag{1}$$

where W is the minimum contention window, and m is the maximum number of backoff stages. Observe that the probability p that a transmitted packet encounters a collision, is the probability that, in a time slot, at least one of the $n-1$ remaining stations transmits. The fundamental independence assumption given above implies that each transmission "sees" the system in the same state, i.e. in steady state. At steady state, each remaining station transmits a packet with probability τ. This yields:

$$p = 1 - (1-\tau)^{n-1} \tag{2}$$

Substituting τ, as expressed by (1), into equation (2), and solving the equation with respect to n, we finally obtain:

$$n = 1 + \frac{\log(1-p)}{\log\left(1 + \frac{2(1-2p)}{p(W+2)+pW(2p)^m-(W+1)}\right)} \tag{3}$$

Equation (3) explicitly expresses n versus the conditional collision probability p, and therefore shows that the estimation of n can be carried out by estimating p.

IV. SIMULATION RESULTS

In order to provide numerical results, we have developed an object-oriented event-driven simulator which reproduces the DCF as defined in [1]. We have restricted our simulations to the basic access procedure described in section II. According to our model, every station works in saturation conditions. In other words, station queues are always not empty. Packet size is fixed. No MSDU fragmentation occurs, since each MSDU corresponds exactly to an MPDU. Each MPDU is composed of a payload, a MAC header, and a PHY header, whose sizes, shown in table II, are those defined in [1], except fot the payload length that we have chosen equal to about half of the maximum value defined in the standard. The values of the other parameters used in the simulation program are also summarized in table II.

packet payload	8184 bits
MAC header	272 bits
ACK lenght	112 bits + PHY header
PHY header	128 bits
Channel Bit Rate	1 Mbit/s
Propagation Delay	1 μs
RxTx_Turnaround_Time	20 μs
Busy_Detect_Time	29 μs
SIFS	28 μs
DIFS	128 μs
ACK_Timeout	300 μs
Slot Time	50 μs

TABLE II

PACKET FORMAT AND PARAMETER VALUES ADOPTED IN THE SIMULATIONS

A. Analysis Validation

To validate the analytical relation between the number of active stations n and the conditional collision probability p, given in 3, we measured via simulation p, varying n. The measure of p has been carried out in a very effective manner as decribed in what follows. First, let us recall that the conditional collision probability p is defined as the probability that a packet transmitted by the considered station fails. This happens if, in the slot time selected for transmission, another transmission occurs. It might appear that the estimation of p requires each station to count the number of failed transmission and divide such a number for the total transmission attempts. However, it is immediate to understand that a much more efficient procedure is to monitor *all* the slot times (thus significantly increasing the number of samples), regardless of the fact that a transmission attempt is performed

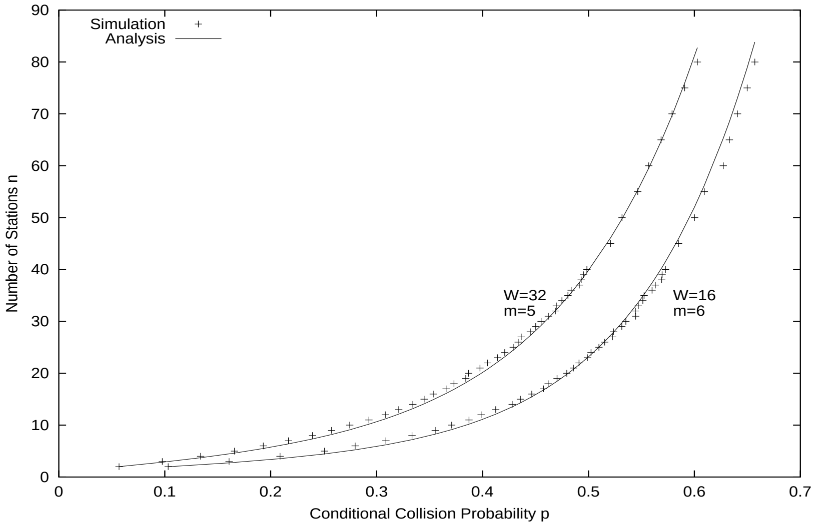

Fig. 3. Number of Stations versus the Conditional Collision Probability

or not. Since in each busy slot an eventual packet transmission would have failed, the conditional collision probability can be obtained by counting the number of experienced collisions, C_{coll}, as well as the number of observed busy slots, C_{busy}, and dividing this sum by the total number B of observed slots on which the measurement is taken, i.e.:

$$p = \frac{C_{busy} + C_{coll}}{B} \tag{4}$$

Figure 3 plots the number of contending stations n versus the conditional collision probability p. The lines represent the analytical relation given in 3, for two different PHY specifications (table I): FHSS, characterized by $W = 16$ and $m = 6$, and DSSS characterized by $W = 32$ and $m = 5$. In addition, the figure reports simulation results. Each simulation run is obtained for a scenario of a constant number n of stations, and lasts 1000 seconds. A 10 seconds warm-up time has been added at the beginning of the simulation. The figure shows that the agreement between simulation results (symbols) and analytical results (lines) is remarkable: the difference between simulation and analysis never exceeds 3%.

B. Run-Time Estimation: ARMA Filter

In order to provide a run-time adaptive estimation of n, it is sufficient to define a convenient run-time estimation algorithm, so that (depending on the specific application in mind) each station or Access Point, on the basis of channel monitoring, can independently evaluate the time-varying number of competing stations in the network.

In general, run-time estimation is provided by simple mechanisms, such as AR (Auto Regressive) or ARMA (Auto Regressive Moving Average) filters. In particular, we have evaluated the effectiveness of the following estimator:

$$\begin{cases} \hat{p}(t+1) = \alpha\hat{p}(t) + \frac{(1-\alpha)}{q}\sum_{i=0}^{q-1} C_{t-i} \\ \\ \hat{n}(t+1) = f(\hat{p}(t+1)) \end{cases} \tag{5}$$

In this equation, $\hat{p}(t)$ is an ARMA smoothing of the conditional collision probability p. The number of competing stations is estimated from $\hat{p}(t)$ by using the non linear function $f(.)$ given in equation (3). The estimation $\hat{p}(t)$ is built upon the computation of the number of busy/idle slots encountered on the channel. Specifically, C_{t-i}, with $i = 0, \cdots, q-1$ are the last q slot samples. C_i is equal to 0 if in the $i-$th slot, either the station does not transmits and sees an empty slot, or the station transmits with success. Conversely, C_i is equal to 1 if the channel is sensed busy during the $i-$th slot, or the station transmits without success[2]. We prefer an ARMA filter rather that a more traditional AR filter (i.e. q=1), since the moving average taken on the last q samples better smooths the fast time scale fluctuations due to the 0/1 gross quantization of each input sample C_{t-i}.

Figure 4 shows the temporal behavior of the running estimate for a reference station, varying the network occupancy status. The parameters characterizing the ARMA filter have been set to $\alpha = 0.999$ and $q = 10$. The q value is not very critical in terms of filter memory, as it is just needed to better smooth the very high frequencies of the fluctuations. Conversely, the selection of a suitable value α

---

[2] Note that, according to equation (3), f(1) is not defined. However, since $\hat{p}(t)$ can be equal to 1 only asintotically, our estimation rule $\hat{n}(t) = f(\hat{p}(t))$ can be pratically applied.

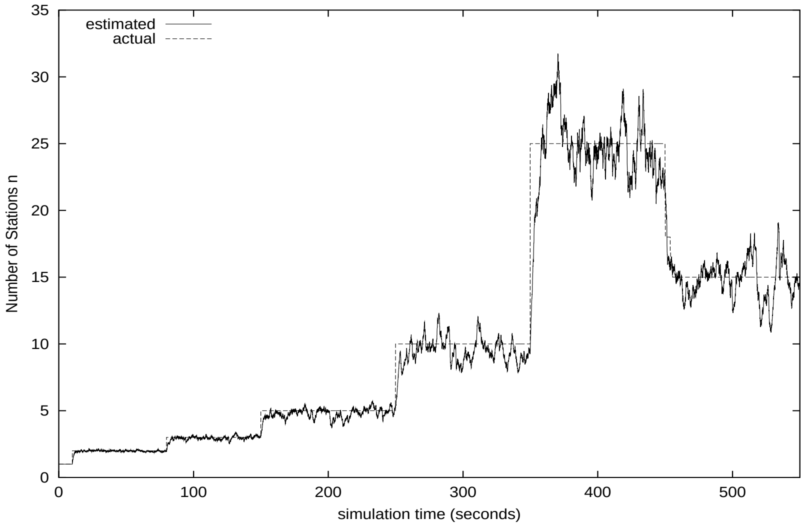

Fig. 4. Dynamic Estimate of the number of contending stations

determines the trade-off between accuracy of the estimation and response time in the case of changes in the number of active stations.

From the figure, we see that the selected value α provides a fairly rapid adaptation to sudden changes in the network configuration. However, it can be noted that the adaptation performance decreases with an increasing number of stations. This behavior is motivated by the fact that, even if the value α is constant, the time constant of the filter (i.e. the filter memory), when expressed in seconds, is given by:

$$\frac{E[\text{slot time duration}]}{1-\alpha}$$

Therefore, it varies with the number of stations, as the average slot duration depends on the collision

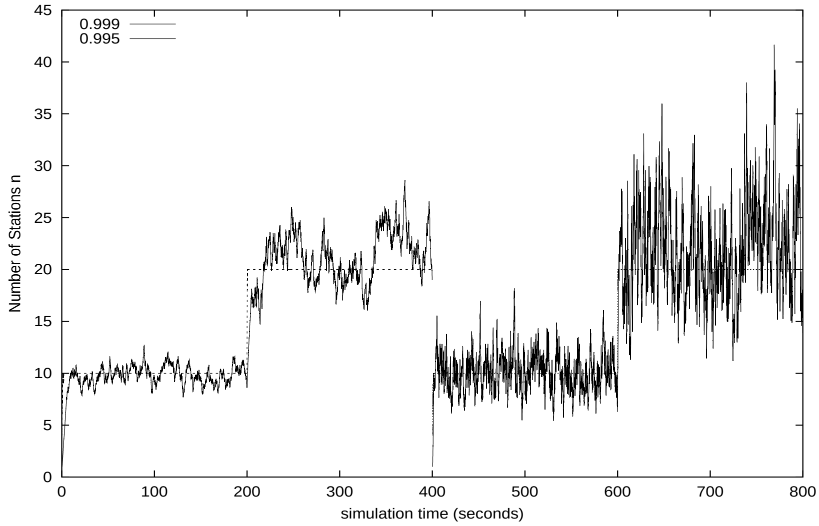

Fig. 5. Run-time ARMA Estimate of the number of contending stations n - $\alpha = 0.999$ and $\alpha = 0.995$

probability.[3]

From the figure, we see also that the accuracy of the estimation degrades as the number of stations increases.This phenomenon is due to the slope of the curve shown in figure 3, which plots $n = f(p)$ as given by equation (3). As the number of stations increases, the slope increases too. This implies that the errors in the collision probability estimate are amplified in the evaluation of the number of contending stations. Moreover, due to the non linearity of the relation $n = f(p)$, the estimation of n results a little biased, as it is $f(E[\hat{p}]) \neq E[f(\hat{p})]$.

This effect is better visualized in figure 5. In the figure, two simulation plots of 400 seconds each

---

[3] In fact, recall that the slot time is variable, and is given by a slot time σ, when the slot is idle, and by the duration of a data packet frame (including overhead and ACK), if the slot time is busy. The average slot duration can be readily obtained from the steady state transmission probability τ, eq. (1) as:

$$E[\text{slot time duration}] = (1-\tau)^n\sigma + n\tau(1-\tau)^{n-1}T_s +$$

$$+ \left[1 - (1-\tau)^n - n\tau(1-\tau)^{n-1}\right] T_c$$

where T_s is the average time the channel is sensed busy (i.e. the slot time lasts) because of a successful transmission, and T_c is the average time the channel is sensed busy by each station during a collision. Explicit expressiond for T_s and T_c for both Basic and RTS/CTS access modes can be found in [4]. For an example, with the parameters of table II we obtain an average slot duration of 1.6 ms when n=5, while this value increases up to 3.8 ms when n=25.

are reported (the simulation is restarted at time 400s with different parameters). The initial number of stations in the network is set to 10, and this number is doubled after 200 seconds, to simulate an abrupt change in the network state. In the figure, the leftmost plot (seconds 0 to 400) shows the case of $\alpha = 0.999$, while the righmost plot shows the case of $\alpha = 0.995$. In both cases, $q = 10$ has been used in the filter.

From the analysis of figure 5, we see that, as the α coefficient increases, the tracking capability of the filter improves, but the accuracy of the estimates degrades. Moreover, as evedent in the rightmost plot of figure 5, the average estimated value $\hat{n}$ is greater than the real value n.

This fact is further shown in figure 6, which plots the probability distributions $P_p(\hat{p})$ and $P_n(\hat{n})$ of both the collision probability estimate and the resulting network occupancy estimate. The plots have been obtained for the case of $n = 20$ contending stations, for $q = 10$, and for both values α considered in the previous figure 5. The x-axis is graduated in terms of percentual deviation from the nominal value $p = f^{-1}(n = 20)$. The spread of the P_p distributions depend on the filter parameters α and q. The little bias from the value 0 is due to the small mismatch between the analytical relation (3) and the simulation results. While the distributions P_p are almost symmetric, their images P_n through the non linear function $n = f(p)$ given in (3), are very distorted. The distorsion is more and more evident as the spread of the $\hat{p}$ distribution increases (i.e. as the α coefficient decreases). Thus, in order to contain the distorsion, the p estimates must be very accurate. Unbiased estimates of n as function of the p estimates are possible only asyntotically, if the p estimates are very accurate, i.e. the filter memory is set to a very large value.

Summarizing, the considered ARMA estimation approach is very simple, but requires some tradeoffs. In particular, the value of the filter memory is critical and has to be set as a tradeoff between tracking capability and estimation accuracy/bias.

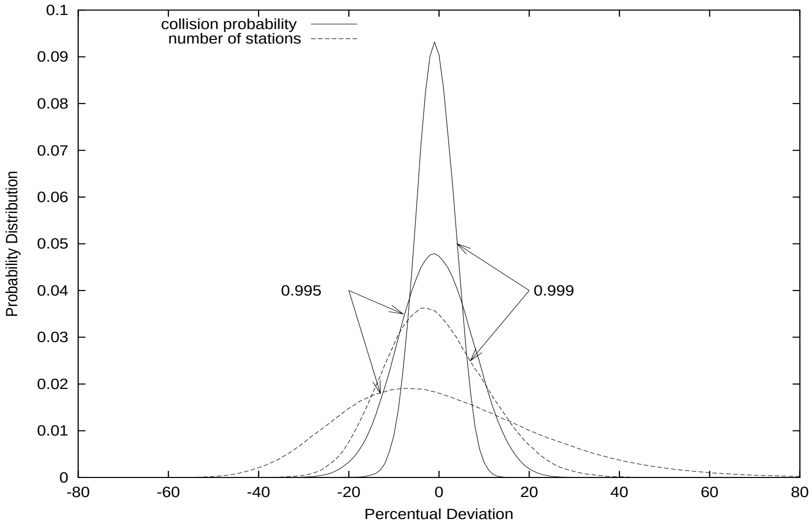

Fig. 6. Probability Density Function of the Estimates $\hat{p}$ and $\hat{n}$

V. Applicative Scenarios

Scope of this section is to briefly summarize two possible practical applications of the proposed estimation technique. A detailed investigation of the proposed approaches is out of the scopes of the present paper, and is object of current research work.

A. Adaptive tuning of the RTS threshold

As described in section II, the 802.11 standard is designed to allow both Basic and RTS/CTS access modes to coexist. The decision whether to use the Basic of the RTS/CTS access modes is not network-wise, but it is independently taken by each transmitting station. In particular, it is suggested that the RTS/CTS access mode should be used by an individual station, when the packet payload exceed a given RTS threshold.

Previous literature work has suggested explicit RTS threshold values which maximize the system performance. A frequently encountered problem is that such an optimization is carried out for a specific network configuration, where the number of mobile stations was fixed in the simulation experiments. However, it has been recently shown [4] that the RTS threshold which maximizes the system throughput is not a constant value, but significantly depends on the number n of competing stations. It is immediate to conclude that the proposed estimation technique has a straightforward application in the definition of a dynamic RTS threshold value depending on n. Specifically, as detailed in section III, the estimation approach proposed in this paper is independent of the access mode adopted. Therefore, it is suited for application to an hybrid Basic-RTS/CTS scenario.

B. Cell selection in an Infrastructured scenario

The IEEE 802.11 standard details the architecture of an *Infrastructured Network*, meant to extend the range of a wired LAN to wireless "cells". The Infrastructured Network employs a number of fixed stations, denoted Access Point (AP), interconnected via an infrastructure called Distribution System (DS). The standard specifies which messages shall be exchanged between an Access Point and a station to support mobility, although it does not explicitly address how mobility is implemented in the DS. Some proposals have appeared in the literature concerning the implementation of a DS, LAN/WAN integration (e.g. logical addressing), etc.

Infrastructured networks introduce, in 802.11, problems typical of cellular systems. One traditional issue in cellular systems is the selection of the best serving station (i.e. AP) to select to, when, upon handover or new call origination, more than one AP is found in the radio coverage area of the mobile station.

We argue that the algorithm in charge of selecting the specific AP to "associate" to (using 802.11 notation) may take advantage of the knowledge of the occupancy status of each cell. As the number of competing stations is a major factor that affects performance, an efficient cell selection algorithm may be designed to minimize the maximum number of competing stations in different cells. Such a number can be independently estimated[4] by each AP, by way of the approach proposed in this paper,

---

[4] There are a number of issues that deserve some comment. First, the underlying assumption is that DCF is employed as MAC protocol, rather than the Point Coordination Function. This is not unreasonable, as PCF is not supported by all 802.11 implementations, and the use of PCF is limited to time constrained services. Second, even if PCF is employed, the proposed approach is still capable of correctly estimating the number of stations competing via DCF (a PCF frame is seen by DCF stations as a single, long, slot time). Third, even in the presence of APs, the information of the number of competing stations is not

and notified, via signaling exchange, to handovering and newly originating stations to allow them a proper cell selection.

VI. Conclusions

In this paper, a methodology to estimate the number of competing stations in an 802.11 Distributed Coordination Function Wireless LAN has been proposed. We have derived a closed form expression that relates the number of competing stations with the conditional collision probability, i.e. the probability of a collision seen by a packet being transmitted on the channel by a selected station. Comparison with simulation results show that the proposed model is extremely accurate.

We have then proposed a practical estimation technique that can be implemented over each individual station. Each station independently monitors the transmissions eventually occurring within each slot-time, and updates a running estimate, smoothed vith an ARMA filter, of the conditional collision probability. Simulation results show that both the accuracy and the responsiveness of the estimate reduces as the number of stations increases. Currently, we are studying better estimation approaches based on adaptive filtering.

The results presented in this paper have significant practical applications. For example, the explicit knowledge of the number of competing stations might dynamically drive the threshold on the packet size over which a station switches from Basic to RTS/CTS access mode. Another application is the use of the proposed estimation for the selection of the less loaded access point in an Infrastructured 802.11 network scenario.

References

[1] IEEE Standard 802.11 - 1999; Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications; November 1999.

[2] B. P. Crow, I. Widjaja, J. G. Kim, P. T. Sakai, "IEEE 802.11 Wireless Local Area Networks", IEEE Commun. magazine, September 1997, pp. 116-126.

available at the upper layers, as the AP is just capable of communicating the number of "associated" stations, which may be very different from the number of active ones (associated stations may be idle)

[3] J. Weinmiller, M. Schlager, A. Festag, A. Wolisz, “Performance Study of Access Control in Wireless LANs IEEE 802.11 DFWMAC and ETSI RES 10 HIPERLAN”, Mobile Networks and Applications, Vol. 2, 1997, pp. 55-67.

[4] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function”, IEEE Journal of Selected Areas in Communication, March 2000.

[5] Y.C. Tay, K.C. Chua, “A capacity analysis for the IEEE 802.11 MAC protocol”, ACM/Baltzer Wireless Networks Vol. 7, No. 2, March 2001, pp. 159-171.

[6] G. Bianchi, L. Fratta, M. Oliveri, “Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs”, Proc. PIMRC 1996, October 1996, Taipei, Taiwan, pp. 392-396.

[7] F. Cali', M. Conti, E. Gregori, “Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit”, Trans. on Networking, Vol. 8, No. 6, Dicember 2000, pp. 785-799.

[8] L. Kleinrock, F. Tobagi, “Packet Switching in radio channels, part II - the Hidden Terminal Problem in Carrier Sense Multiple Access and the Busy Tone Solution”, IEEE Trans. Comm., Vol. COM-23, No. 12, December 1975, pp. 1417-1433.

[9] K. C. Huang, K. C. Chen, “Interference Analysis of Nonpersistent CSMA with Hidden Terminals in Multicell Wireless Data Networks”, Proc. of IEEE PIMRC ’95, Toronto, Canada, September 1995, pp. 907-911.

[10] H. S. Chhaya, S. Gupta, “Performance Modeling of Asynchronous Data Transfer Methods of IEEE 802.11 MAC Protocol”, Wireless Networks, Vol. 3, 1997, pp. 217-234.