Ad Hoc Networks 11 (2013) 625–638

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

A new adaptive receiver-initiated scheme for mitigating starvation in wireless networks

A. Leonardi [a,*], S. Palazzo [a], C. Rametta [a], E.W. Knightly [b]

[a] Dipartimento di Ingegneria Elettrica, Elettronica ed Informatica, University of Catania, Italy
[b] Department of Electrical and Computer Engineering, Rice University, Houston, TX, United States

ARTICLE INFO

Article history:
Received 6 September 2010
Received in revised form 22 August 2011
Accepted 5 September 2011
Available online 1 October 2011

Keywords:
MAC
CSMA/CA
Receiver-initiated
Evaluation
Wireless
Starvation
Simulation

ABSTRACT

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) has been adopted by the IEEE 802.11 standard and provides good performance when all transmitters are within the range of each other. Unfortunately, in multi-hop topologies, the asymmetric view of the channel state leads to a throughput distribution where a few flows may capture all the available bandwidth while many other flows get very low throughput and sometime meet starvation. To address this problem, in this paper we describe a solution called Carrier Sense Multiple Access with Collision Avoidance by Receiver Detection (CSMA/CARD) which makes use of collisions sensed by a receiver at the physical layer to help the handshake mechanism and mitigate the effect of such problem. More specifically, we propose a mechanism based on historical observations, where collisions can be used by the receiver to predict whether some sender attempted to initiate a transmission. The receiver then reacts accordingly by participating itself in a handshake sequence. We show some interesting results, obtained through analysis and simulations, when the CSMA/CARD is compared to the IEEE 802.11 protocol.

1. Introduction

Carrier Sense Multiple Access (CSMA) has been used in different packet-radio network protocols [18]. CSMA protocols require each station to listen to the channel before attempting to transmit in order to avoid having simultaneous transmissions over the same channel.

Several studies have demonstrated that when all nodes are within the transmission range of each other, CSMA protocols provide fair access opportunities to all flows. Unfortunately, in a multi-hop topology where nodes are not in the range of each other, channel state information is ineluctably incomplete because some transmitters are not able to sense when other nodes are transmitting [14]. This lack of awareness leads to poor performance, affecting transmissions in different ways, even if coordination enhancements like RTS/CTS control packet exchanges, as in CSMA/CA, are used [3]. In particular, the possibility to acquire the channel becomes different among the terminals and a throughput distribution occurs in which a few flows capture all bandwidth while several other flows get very low or even zero throughput [11].

To address this problem, this paper discusses a solution called Carrier Sense Multiple Access with Collision Avoidance by Receiver Detection (CSMA/CARD) whose basic idea has been introduced in [19]. This solution is based on a novel receiver-initiated mechanism which exploits some information at the physical level. We demonstrate, via analysis and simulations, that the detection of two or more overlapped signals at a potential receiver, when coupled with an appropriate mechanism, can be effective in

* Corresponding author. Tel.: +39 0957382391; fax: +39 0957382397.
E-mail addresses: aleonardi@dieei.unict.it (A. Leonardi), spalazzo@dieei.unict.it (S. Palazzo), crametta@dieei.unict.it (C. Rametta), knightly@rice.edu (E.W. Knightly).

1570-8705/$ - see front matter
doi:10.1016/j.adhoc.2011.09.005

providing extra channel state information for a protocol like the IEEE 802.11 DCF [30]. Using this information, the receiver can initiate an action to help the handshake mechanism and to avoid the starvation of some flows which may occur in some network scenarios.

In [19] we have already shown that the mechanism proposed alleviates the problem of starvation. However, results in [19] also show that when the number of contending nodes increases, the number of collisions among packets increases as well, and consequently it may happen that CSMA/CARD does not behave as desired.

In this paper we propose an improved version of CSMA/CARD which introduces a PHY-aware mechanism that, according to historical observations, allows to adaptively consider the dynamics of the number of nodes and the traffic conditions in the neighborhood.

The performance evaluation of the adaptive CSMA/CARD mechanism is carried out by comparing an appropriate existing contention-based protocol without the PHY-aware mechanism against the same protocol with the PHY-aware mechanism. We have here selected the IEEE 802.11 DCF due to its design simplicity and its consolidated analysis, as well as the expected familiarity of readers with this well-known protocol.

The rest of the paper is organized as follows. In Section 2 we provide a description of related works already existing in the literature, and we outline the distinguished approach of our solution as compared to them. In Section 3 we first show two illustrative application scenarios, next we describe the protocol mechanism and highlight, with the help of some pseudo-code, the main algorithms. In Section 4 some analytical results are derived and discussed. In Section 5 we analyze the performance of the proposed mechanism as compared to a classical approach like IEEE 802.11 DCF. Finally, in Section 6 some concluding remarks are drawn.

2. Related work

In the past, several solutions have been proposed to counteract the so-called hidden terminal problem in single-channel networks [17,27,7]. An example is Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [15] which is an improvement of CSMA and suggests a bi-directional handshake mechanism between each sender and receiver to detect collisions. A sender initiates a Request-To-Send packet (RTS) to the receiver, and the receiver replies with a Clear-To-Send packet (CTS) if it receives the RTS correctly. An RTS-CTS handshake mechanism derived from CSMA/CA, has been standardized and adopted by the IEEE 802.11 committee [6,30].

More recently, different techniques have been proposed in the literature to improve the throughput performance of the IEEE 802.11 protocol while preventing the hidden terminal problem. In particular, a major effort has been spent on the modification of the IEEE 802.11 MAC layer timers and handshake mechanisms [3] with the purpose of achieving more spatial reuse. In addition, there have also been papers that aimed at improving the IEEE 802.11 MAC bandwidth efficiency by exploiting the physical layer capture effect [28,20,5]. However, despite the numerous modifications suggested to improve their throughput performance, the IEEE 802.11 MAC protocol and its variants can suffer severe unfairness problems in multi-hop ad hoc networks, due to location-dependent contention issues and the backoff mechanisms.

Designing the appropriate backoff mechanism has been widely studied for achieving specific fairness guarantees [21,12,23,2]. However, irrespective of which backoff mechanism is used, the underlying channel access scheme remains largely inefficient. This is because the prevailing contention resolution mechanisms are sender-initiated whereas, in most cases, the receiver has better knowledge of the channel state than the sender, and consequently can help to reduce the contention.

Different receiver-initiated MAC protocols have been proposed in literature. Bharghavan et al. [3] suggested to use a Request-for-Request-To-Send (RRTS) packet mechanism initiated by the receiver to alleviate the unfairness problem. Receivers which decode an RTS packet and cannot reply with a CTS, because one of its neighbor nodes has already started a transmission, wait until their NAV expires. Once this occurs, the receivers transmit an RRTS packet to the senders, requesting the RTS to be sent back. In this way, the RRTS packet can help reduce the extra backing-off inefficiency and the consequent unfairness at the expense of introducing more overhead due to the RRTS packet itself.

Talucci and Gerla [26] proposed the MAC-BI as the first fully receiver-initiated MAC protocol that exhibits less overhead than MACAW in [3]. Fullmer and Garcia-Luna-Aceves [9] suggested further improvements beyond the MAC-BI in order to achieve better throughput performance in single-hop networks with high load. Garcia-Luna-Aceves and Tzamaloukas [1] advanced the work on the 802.11 MAC even further and proposed several data-collision-free receiver-initiated MAC protocols. The data-collision-free property in [1] is achieved by adding some short control packets to the handshake mechanism and modifying the current 802.11 timing parameters.

Despite their claimed benefits, receiver-initiated schemes have not seen wide application in practice. This is primarily because fully receiver-initiated schemes can sometimes initiate many unnecessary handshake packets that waste the network bandwidth (although this is also true in the case of fully sender-initiated schemes). Moreover, the receiver-initiated schemes introduced so far require a per-receiver traffic estimator that should successfully work under dynamic topology and traffic environments. Another important reason why receiver-initiated protocols have not seen wide acceptance is that the state-of-the-art receiver-initiated protocols cannot interoperate with the current widely deployed IEEE 802.11 MAC devices.

As opposed to other contention-based MAC protocols, the mechanism described in this paper is peculiar for the approach it follows and for its ability to overcome the shortcomings of classical receiver-initiated MAC protocols. More specifically, the mechanism that we introduce does neither imply a fully receiver-initiated scheme nor a fully sender-initiated scheme thereby avoiding the disadvantages of the two protocol classes which have been discussed above.

The new protocol we propose makes use of collisions at the physical layer in order to predict the existence of a potential sender in a timely manner. Actually, other approaches which use collisions at the physical layer have been largely studied in several recent works, though with different aims. As an example, in [29,24], the authors propose two methods to detect collisions in order to differentiate between losses due to channel noise and losses due to packet collisions. In [25], a mechanism which exploits collision detection based on signal correlation, is presented and evaluated, with the purpose of attempting to approximate the CSMA/CD behavior in wireless networks. In [8], the authors use collision detection to propose a new backoff algorithm which substitutes the binary exponential backoff implemented in the classical IEEE 802.11 DCF.

As compared to other similar solutions, the scheme we propose does not require any traffic estimator [1,26]. This allows easier implementation with minimal modifications (only network driver changes are needed), simple protocol design, and interoperability with legacy devices implementing the 802.11 standard.[1]

3. CSMA/CARD

In this section we describe the CSMA/CARD, Carrier Sense Multiple Access with Collision Avoidance by Receiver Detection, whose basic mechanism has been introduced in [19]. CSMA/CARD makes use of events occurring at the physical layer (collisions) in order to mitigate the low performance shown by CSMA/CA protocols, such as the one implemented in the IEEE 802.11 DCF. Low performance arises in topologies where the different transmitters are out of range; in such type of topologies, in fact, the channel state information becomes incomplete and the possibility to acquire the channel becomes uneven among the terminals.

We start describing the topologies where CSMA/CA approaches fail, then, for the sake of completeness, we present an overview of the CSMA/CARD basic algorithm, and finally we introduce an adaptive technique which further improves the performance of CSMA/CARD as compared to the results obtained in [19].

3.1. Scenarios

The Symmetric Incomplete State (SIS) and Asymmetric Incomplete State (AIS) are two exemplary scenarios which have been introduced in [10].

The first one, shown in Fig. 1, is characterized by short-term unfairness (s.t.u.) and long-term fairness. The origin of s.t.u., mainly resides in the exponential backoff mechanism and in the probability of control packet loss perceived by both transmitters, *A* and *D*. Each transmitter is not aware of the activity of the other transmitter, consequently it is possible that a packet is sent (for example from *A*) when another transmission is already started (from *D*). In this case, the receiver *B* cannot answer due to the NAV allocation [30], consequently the transmitter (*A*) is forced to double its own contention window, so decreasing its possibility to acquire the channel in the near future, while the other (*D*), after a successful transmission, reduces its contention window to the minimum value. This condition, where a flow dominates the other in the channel contention, remains until the disadvantaged transmitter (*A*), once the retry limit is reached, drops the packet and puts its contention window to the minimum value. At this point, the two transmitters have the same probability to acquire the channel. In a long term, if the traffic offered is the same for the two transmitters, each of them, in the average, will access the channel for the 50% of the time. Such short-term unfairness affects real-time traffic (voice, video, etc.) quality.

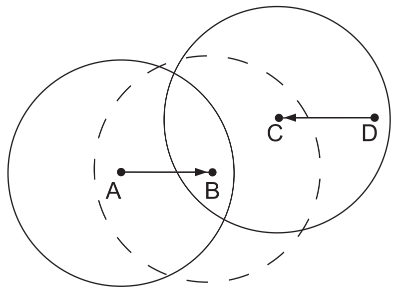


Fig. 1. Symmetric Incomplete State (SIS). *A* and *D* represent the transmitters, while *B* and *C* represent the receivers.

More problematic is the Asymmetric Incomplete State, shown in Fig. 2, which causes the starvation of a flow to the advantage of the other one which captures all the channel bandwidth. The fundamental difference with the previous topology is the asymmetric perception of the channel state from the two transmitters. In particular, transmitter *A* does not sense any packets belonging to flow f_{CD}, and consequently, fully ignores the activity of the other flow. On the other hand, sender *C* knows exactly when to contend for the channel, through the control packets sent by the receiver of flow f_{AB}. Therefore, sender *A* has to discover an available time-slot randomly without any coordination with sender *C*, resulting in many attempts of sender *A* without any response back from receiver *B*. Most of these random attempts occur in the middle of a transmission of flow f_{CD} and result in collision at receiver *B*. Consequently, sender *A* is forced to timeout and to repeatedly double its

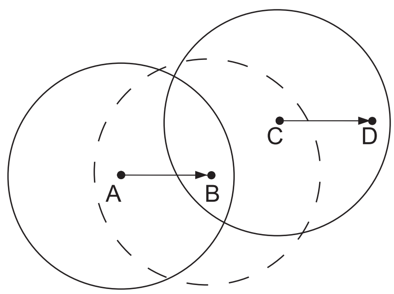


Fig. 2. Asymmetric Incomplete State (AIS). *A* and *C* represent the transmitters, while *B* and *D* represent the receivers.

[1] Specifically, devices using the 802.11 standard implementation simply discard the RRTS packets.

contention window, thus reducing the chances of attempting a new transmission in the next available time slot. As a consequence, the throughput of flow f_{AB} approaches zero, even if the fraction of time sensed busy by sender A is zero.

3.2. Basic mechanism

CSMA/CARD can be considered a hybrid solution, both sender-and-receiver-initiated, where each receiver can predict the existence of a potential sender in a timely manner while minimizing the probability of false predictions and maximizing that of true ones. The approach for counteracting such a challenge is by making use of the events occurring at the physical layer. In particular, detection of significant received signal power variations can be probabilistically interpreted as handshake messages initiated by a potential sender. The receiver then, anticipating a sender-initiated handshake attempt, reacts accordingly (once the channel is sensed idle) by participating itself in the handshake sequence.

Specifically, whenever a collision is detected[2] at a node (two packets are transmitted simultaneously within the range of this node), the node can assume that, with a certain probability, the collision took place with an RTS packet which was intended for it, and it broadcasts a Request-for-Request-To-Send (RRTS) packet accordingly. Bharghavan et al. [3] had a similar receiver cooperation approach as the one proposed here in the context of CSMA/CA protocols which does not rely on physical layer events, but it works only if the RTS is correctly decoded by the receiver (i.e., in the SIS case).

Note that CSMA/CARD is based on IEEE 802.11 DCF mechanisms, and we will use in the following the same terminology and most of the parameters already used in IEEE 802.11 DCF. If not differently specified, the value of the parameters (DIFS, SIFS, EIFS, RTS and CTS size) are the same of those in [30].

Now, denote a potential receiver by R and the set of all its potential sending neighbors by S_R. Upon an RTS transmission by a node $S \in S_R$, R will send back a CTS if R is in the idle state. Otherwise, if R is not in the idle state, we can distinguish two different cases depending on whether R is able to decode the RTS or not.

3.2.1. Decodable RTS

In this case, the CSMA/CARD behaves in the same way as described in [3]. R is able to decode the RTS packet of S but it cannot reply because it is in a defer state due to another transmission sensed within its radio range. This means that R has the NAV allocated and it cannot act until its expiration [30]. Right after R's NAV expires, R contends the channel at the minimum contention window, CW_{min} ([0,31]), and it sends an RRTS packet to S. When the RRTS packet is received by S, S defers for a SIFS period and it sends back an RTS packet to R. Any node $N_i \notin S_R$ which receives the RRTS packet will set its NAV to:

$$NAV(RRTS) = SIFS + RTS + SIFS + CTS. \quad (1)$$

When R receives the RTS, it will answer with a CTS because the channel has been reserved thanks to the RRTS.

The temporal evolution of the channel state perceived from the receiver R, the transmitter S and the other nodes, N_i, within the radio range of R, is represented in Fig. 3.

This mechanism can be applied to the SIS topology (Fig. 1) where it has been demonstrated [10] that it partially mitigates the observed short-term unfairness.

3.2.2. Non-decodable RTS

The main novelty of CSMA/CARD concerns the management of the case when the RTS cannot be decoded because it has been corrupted by other concurrent transmissions. From now on we will investigate this scenario.

In this case, R detects a collision for the duration of an RTS packet while R is not already in the process of another transmission procedure (i.e., R has not just sent or received a CTS or a DATA packet). Suppose that the collision involves an RTS packet sent from a potential sender $S \in S_R$ (the case when the collision is due to other packets than RTS will be considered later). In such a case, R broadcasts an RRTS packet deferring at the minimum contention window, $CW_{min}([0,31])$, after R's NAV expires by an Extended-IFS (EIFS) period [30] (this time has been defined in the IEEE 802.11 DCF standard as the time to wait after a collision). The RRTS is sent broadcast because in this case the receiver does not know the address of the sender.

When the interested node S receives R's RRTS, it will contend for sending an RTS to R at the minimum contention window size $CW_{min}([0,31])$. As it is possible, there is more than one potential sender to R, sending such an RTS takes place by contention where a node S would start contending after deferring for a DIFS period. Consequently, upon the reception of the RRTS, all the other nodes in $N_i \notin S_R$ will set their NAV for a period of time equal to:

$$NAV(RRTS) = DIFS + CW_{min} + RTS + SIFS + CTS. \quad (2)$$

Fig. 4 illustrates the temporal evolution of the proposed mechanism. This procedure can be applied to the AIS topology, shown in Fig. 2.

Analytical and simulation results presented in [19] show that CSMA/CARD, when applied to the AIS scenario, achieves almost perfect fairness.

We report in Table 1 the most significant results obtained in [19]. In particular we report the value of the

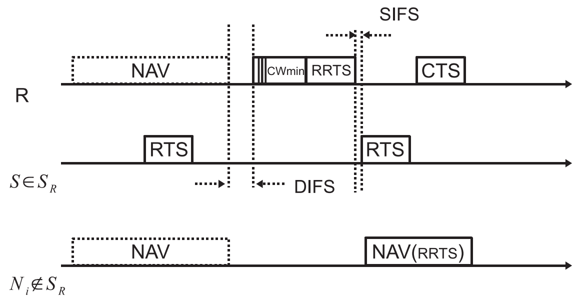


Fig. 3. CSMA/CARD behavior when the RTS received is decodable.

[2] Please refer to [19] where the authors show how to use the known methods of the detection theory [16] to solve the collision detection problem. Other approaches, based on transmission time information and RF energy, signature correlation, Received Signal Strength, can be found in [29,25,24], respectively.

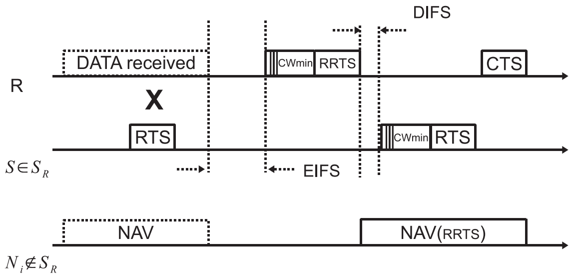


Fig. 4. CSMA/CARD behavior when the RTS received is non-decodable.

Table 1
Collision probability and normalized throughput achieved in AIS scenario.

	CSMA/CA	CSMA/CARD
$p(A)$	0.9364	0.5602
$T_P(A)$	0.0093	0.4170
$T_P(C)$	0.7961	0.4385

collision probability calculated for A, $p(A)$, and the normalized throughput of node A and C, respectively $T_P(A)$ and $T_P(C)$, when the CSMA/CA and CSMA/CARD are considered. The analytical results show that the collision probability evaluated for node A using the CSMA/CA approach is almost 1; it means that the flow f_{AB} is starved while, using the CSMA/CARD approach, this value of collision probability drops down to about 50%. Moreover, the value of throughput calculated for flow f_{AB} is almost the same of that evaluated for f_{CD}.

3.3. Adaptive CSMA/CARD

Performance results obtained for random topologies in [19] show that CSMA/CARD alleviates the problem of starvation. However, it is possible to observe that in several cases (high density of nodes and specific topologies) tuning the number of RRTS sent is a key factor in order to obtain better performance. This is due to the fact that, when the number of nodes contending for the channel increases, the number of collisions among packets increases as well, and consequently it may happen that RRTS packets are sent in response to collisions that are not due to an RTS but to other packets originated by neighbor nodes (note that this occurs only in the AIS topologies).

This wrong behavior of CSMA/CARD leads to an increase of collisions due to RRTS packets and could be fixed through a mechanism which adaptively adjusts the probability to send RRTS.

3.3.1. Probability to send RRTS

In order to tune the probability to send RRTS, we have created an algorithm which takes into account the number of responses to the RRTS received in the past. This algorithm, which runs in the receiver, relies in a variable, called *probability to send RRTS*, P_{RRTS}, and two constants, K_+ and K_-, which represent the increment and decrement factors.

When a collision occurs, the receiver station sends an RRTS with a probability, P_{RRTS}, and starts a timer which represents the estimated time the receiver station has to wait in order to receive an answer to the RRTS sent. The timer expires after an interval equal to:

$$timeout = TX(RRTS) + \delta + DIFS + CW_{min} + TX(RTS) + \delta, \quad (3)$$

where TX (RRTS) and TX (RTS) represent the time needed to transmit an RRTS and RTS respectively, and δ represents the propagation time. Each time the receiver does not receive a response by the expiration of the *timeout*, the P_{RRTS} is decreased by K_-, until a threshold value called $Threshold_{min}$. Otherwise, if a response arrives by the expiration of the *timeout*, the P_{RRTS} is increased by K_+, until a maximum value called $Threshold_{max}$.

In the first case, this mechanism limits the number of RRTS which will be sent in the future, when the collisions perceived in the past are not due to potential senders, while in the second case, it increases the probability to send RRTS if the situation of starvation occurs.

The values of K_+, K_-, P_{RRTS}, $Threshold_{min}$ and $Threshold_{max}$ will be discussed in Section 5.

3.3.2. Tuning timeout

The CSMA/CARD mechanism described until now is effective only when the potential transmitter can answer the RRTS almost immediately. Otherwise, when the transmitter is in defer state due to other on-going transmissions around its coverage area, it is possible that the estimated time chosen by the receiver to finalize the handshake, as calculated in Eq. (3), is not sufficient and consequently the *timeout* expires before the reception of an answer to the RRTS.

In order to solve this problem, the CSMA/CARD, before decreasing the P_{RRTS}, checks if the lack of answer to the RRTS could be solved by increasing the value of the *timeout*. Correspondingly, the algorithm increases the value of NAV (RRTS), included in the packet RRTS, which specifies to the neighbor nodes the period of time in order to set their NAV.

We can better explain the importance of this improvement by looking at Fig. 5. Let us assume all the senders are backlogged. Using a classical CSMA/CA approach or a CSMA/CARD without the right tuning of the timeout, the flow 3–4 reaches very low values of throughput, sometimes close to zero. This is because node 4, once a collision has been detected, waits for the end of the communication between 5–6, and then it sends an RRTS. Unfortunately, node 3 cannot answer this RRTS because it is in a defer state due to the communication between 1–2. Consequently, the channel will be occupied again by the

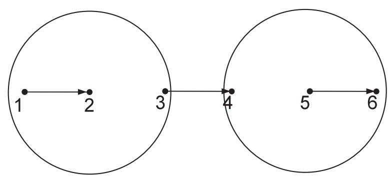


Fig. 5. Situation of starvation for the pair 3–4 without a right choice of timeout.

Function 1
handle_collision()

```
if (pkt_CTRL == NULL) then
  p = random(0, 1);
  if (p < P_RRTS) then
    if (inc_NAV == TRUE) then
      NAV(RRTS) = DIFS + k · CW_min + RTS + SIFS + CTS;
      timeout = TX(RRTS) + δ + DIFS + k · CW_min + TX(RTS) + δ;
    else
      NAV(RRTS) = DIFS + CW_min + RTS + SIFS + CTS;
      timeout = TX(RRTS) + δ + DIFS + CW_min + TX(RTS) + δ;
    end if
    pkt_RRTS = create_RRTS(BDCST, NAV(RRTS));
    defer(EIFS);
    backoff(CW_min);
    transmit(pkt_RRTS);
    set_timer(timeout);
  end if
end if
```

Fig. 6. Pseudo-code of the procedure called after a collision.

communication between 5–6. This situation can be solved increasing the period of time that the other nodes should defer their transmission, and consequently the *timeout*. In particular, once the RRTS is received, node 5 sets its NAV to the following value:

$$NAV(RRTS) = DIFS + k \cdot CW_{min} + RTS + SIFS + CTS, \quad (4)$$

while the receiver 4 will wait for the answer for a period of time equal to:

$$timeout = TX(RRTS) + \delta + DIFS + k \cdot CW_{min} + TX(RTS) + \delta. \quad (5)$$

Note that when $k = 1$, Eq. (4) becomes Eq. (2), i.e. the mechanism behaves as in the basic mode. The right choice of the value of the parameter k will be discussed in Section 5.

3.4. Algorithms

In this section we describe the main algorithms which realize the procedures described above. More specifically, some of these algorithms use new functions, others use modifications of the classical IEEE 802.11 functions.

3.4.1. Handle_collision()

Once a collision has occurred, the algorithm checks if the node (in this case the receiver) is not already involved in another transmission, i.e., there are no control packets (CTS or ACK) ready to be sent. Then, a random number is generated and compared to the value of P_{RRTS}. If the response is positive (i.e., the receiver is able to send an RTS), it checks a boolean variable called *Inc.*_NAV and in accordance with its value the receiver chooses to use either an extended NAV (RRTS) (see Eq. (4)) or a standard one (see Eq. (2)). After that, it sets the value of *timeout* accordingly. Then, the node prepares the RRTS packet with destination BROADCAST (note that we are in the AIS case) and the chosen NAV (RRTS). It waits a period of time equal to *EIFS* and then it starts a backoff with a value of contention window equal to CW_{min}. At the expiration of the backoff, if the channel is free, the packet will be transmitted and a timer of duration *timeout* starts. Pseudo-code of the *handle_collision()* algorithm is reported in Fig. 6.

3.4.2. Receive_RTS()

When a node receives an RTS, it checks if this RTS has been sent in response to an RRTS. If this is the case, the algorithm increases the value of P_{RRTS} by K_{+}, and it checks if the new value is less than $Threshold_{max}$. Then, the variable *RRTS_replied*, which takes into account the number of consecutive successful transmissions, is updated. If *RRTS_replied* reaches the value of *RRTS_replied_limit* and the value *Inc.*_NAV is *TRUE*, this means that with high probability the congestion occurred in the past has terminated, and consequently the variable *Inc.*_NAV could be set to FALSE (this will set the following NAV (RRTS) to a standard value). Moreover, the variable *RRTS_no_replied* is set to zero. Finally, a CTS is sent. The value of *RRTS_replied_limit* will be discussed in Section 5. Pseudo-code of the *receive_RTS()* algorithm is reported in Fig. 7.

3.4.3. Handle_timer()

If the timer expires without response to the RRTS, it means that the collision sensed is due to either noise or contention among packets belonging to other flows, or the potential sender cannot reply because the channel sensed is busy. Accordingly, the P_{RRTS} (if its value is higher than $Threshold_{min}$) is decreased by K_{-}. Then, the variable *RRTS_no_replied*, which takes into account the number of consecutive unsuccessful transmissions, is increased and if it reaches the value *RRTS_no_replied_limit*, it is possible clue that the channel sensed by the transmitter is congested. Consequently, if *Inc.*_NAV is set to FALSE, it will be set to TRUE, otherwise, it means that increasing the duration field has not improved the situation, consequently the NAV is brought to its previous value, and the value of *RRTS_no_replied_limit* is doubled. This last action allows to wait until a possible congestion around the

Function 2
$receive\_RTS()$

if $(pkt\_RRTS\ != NULL)$ **then**
  $P_{RRTS} = P_{RRTS} \cdot K_{+};$
  if $(P_{RRTS} > Threshold_{max})$ **then**
    $P_{RRTS} = Threshold_{max};$
  end if
  $RRTS\_replied++;$
  if $(RRTS\_replied == RRTS\_replied\_limit)$ AND $(inc\_NAV == TRUE)$ **then**
    $(inc\_NAV = FALSE)$
  end if
  $RRTS\_no\_replied = 0;$
end if
$send\_CTS();$

Fig. 7. Pseudo-code of the procedure called after an RTS has been received.

Function 3
$handle\_timer()$

$P_{RRTS} = P_{RRTS} \cdot K_{-};$
if $(P_{RRTS} < Threshold_{min})$ **then**
  $P_{RRTS} = Threshold_{min};$
end if
$RRTS\_no\_replied++;$
if $(RRTS\_no\_replied == RRTS\_no\_replied\_limit)$ **then**
  if $(inc\_NAV == TRUE)$ **then**
    $(inc\_NAV = FALSE)$
    $inc\_no\_replied\_limit();$
  else
    $inc\_NAV = TRUE;$
  end if
  $RRTS\_no\_replied = 0;$
end if
$RRTS\_replied = 0;$

Fig. 8. Pseudo-code of the procedure called after the expiration of the timeout.

sender is finished. Then, the variable *RRTS_no_replied* is set to zero. Finally, the variable *RRTS_replied* is set to zero. The value of *RRTS_no_replied_limit* will be discussed in Section 5. Pseudo-code of the *handle_timer()* algorithm is reported in Fig. 8.

4. Analysis

In this Section we derive the analysis of the proposed mechanism in the scenario shown in Fig. 9. This scenario can be considered an extension of the asymmetric incomplete state (AIS), already shown in Fig. 2, in which we consider *N* transmitter–receiver pairs which contend for the medium with another flow, *A*–*B*. Therefore, there are *N* transmitters all of which are in the same range and also in the range of receiver *B*.[3]

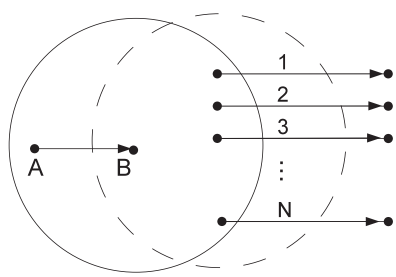


Fig. 9. Scenario.

The model we follow in this section borrows from that derived in [4,10]. In particular, we build a model representing the channel state as seen by the individual channel sources and we compute the per-flow throughput that can be achieved using CSMA/CARD.

According to this model, the behavior of an arbitrary station employing a CSMA/CARD protocol can be identified by four different states: (*i*) idle, (*ii*) occupied by a successful transmission of the station, (*iii*) occupied by a collision

[3] It is worth pointing out that the aim of the proposed approach is to mitigate starvation occurring in scenarios similar to the AIS, consequently the scenario in Fig. 9 represents the most suitable to evaluate both the effect of the proposed approach for improving the throughput of the flow *A*–*B*, and the influence of the RRTS packets on the data transmissions of neighbor nodes.

involving or not a transmission of the station, and (iv) busy due to the activity of other nodes.

The time intervals during which the station remains in each of the four states above are denoted by σ, T_s, T_c and T_b, respectively.

Because we are interested to the saturation throughput, we will assume in the following that each node has always a packet ready to be sent.

We start analyzing the exponential backoff mechanism associated with the CSMA/CARD mechanism and the related collision probability. As already happens in the IEEE 802.11 DCF, also in the CSMA/CARD each station starts its backoff process after the channel has been sensed idle for a time interval equal to DIFS. In particular, as a first attempt, the node tries to send a packet using a minimum contention window equal to CW_{min}. Each time a collision occurs, the contention window is doubled until a maximum value CW_{max}, or a maximum retransmission limit, M, is reached.

If we call m the maximum number of backoff step, we have:

$$m = log_2\left(\frac{CW_{max}}{CW_{min}}\right). \tag{6}$$

Assuming $\overline{W}$ the average backoff window in the saturation case and τ the probability of a node's transmission in a slot, it follows that $\tau = 1/\overline{W}$. The expression of $\overline{W}$ has been calculated in [22] and it is given by:

$$\overline{W} = \begin{cases} \frac{1-p-p(2p)^m}{1-2p}\cdot\frac{CW_{min}}{2} & M \geqslant m \\ \frac{1-p-p(2p)^{M-1}}{1-2p}\cdot\frac{CW_{min}}{2} & M < m \end{cases}, \tag{7}$$

where p is the probability that a transmission of the station is not successful due to a collision.

In order to calculate p, we note that, assuming a scenario with N nodes, the probability that a transmission is successful is the probability that none of the other $N-1$ nodes transmit in that time slot, i.e. $1-p = (1-\tau)^{N-1}$.

Assuming $M \geqslant m$, it follows that:

$$p = 1 - \left(1 - \frac{1-2p}{1-p-p\cdot(2p)^m}\cdot\frac{2}{CW_{min}}\right)^{N-1}. \tag{8}$$

Now, we derive the throughput, ρ, which can be calculated as:

$$\rho = \frac{\Pi_S}{\Delta}, \tag{9}$$

where Π_S represents the probability that a node transmits a packet successfully, and Δ is the average cycle time.

We denote with $T_{s,N}$ the duration of a successful transmission performed by one of the N transmitters, with $T_{s,AB}$ the duration of a successful transmission belonging to the pair A–B (note that we are in the AIS configuration and consequently we can assume that the communication between A and B usually occurs after an RRTS sent by node B), and with T_c the time needed in order to detect a collision. In particular:

$$T_{s,N} = RTS + SIFS + CTS + SIFS + E\{P\} + SIFS + ACK + DIFS,$$

$$T_{s,AB} = RRTS + DIFS + CW_{min}/2 + RTS + SIFS + +CTS + SIFS + E\{P\} + SIFS + ACK + DIFS,$$

$$T_c = RTS + DIFS, \tag{10}$$

where $E\{P\}$ represents the average size of the packet. Moreover, we are assuming that the propagation delay is negligible and we are supposing that A transmits the RTS packet solicited by an RRTS using a constant contention window, CW_{min}; consequently, the average backoff time is $CW_{min}/2$.

Now we call τ_B and τ_N the probability of node B's transmission and the probability of one of the N's transmission in a slot, respectively. Node B always transmits the RRTS packet using CW_{min}, thus:

$$\tau_B = \frac{2}{CW_{min}}, \tag{11}$$

and

$$\tau_N = \frac{(1-2p)}{1-p-p\cdot(2p)^m}\cdot\frac{2}{CW_{min}}. \tag{12}$$

Moreover, knowing p and τ_N, we can calculate c, which represents the probability that a collision occurs among the packets sent from the N transmitters. This probability can be calculated as the probability that at least two of the N nodes transmit in the same time:

$$c = 1 - (1-\tau_N)^N - N\cdot\tau_N\cdot(1-p). \tag{13}$$

Now we can proceed with the calculation of the throughput, ρ_N, of the N transmitters:

$$\rho_N = \frac{\Pi_{s,N}}{\Delta_N}, \tag{14}$$

where

$$\Delta_N = \Pi_{s,N}\cdot T_{s,N} + \Pi_{idle}\cdot\sigma + \Pi_c\cdot T_c + \Pi_{b,B}\cdot T_{s,AB} + \Pi_{b,N}\cdot T_{s,N}, \tag{15}$$

$\Pi_{s,N}$ represents the probability that one of the N nodes transmits a packet successfully, Π_{idle} the probability that the channel remains idle, Π_c the probability that a collision occurs, $\Pi_{b,B}$ and $\Pi_{b,N}$ the probabilities that the channel is busy due to a transmission by node B, or a transmission by one of the other $N-1$ nodes, respectively. The previous probabilities can be calculated as:

$$\begin{aligned} \Pi_{s,N} &= \tau_N\cdot(1-p) \\ \Pi_{idle} &= (1-\tau_B)\cdot(1-\tau_N)^N \\ \Pi_c &= c \\ \Pi_{b,B} &= \tau_B\cdot(1-\tau_N)^N \\ \Pi_{b,N} &= (N-1)\cdot\tau_N\cdot(1-p). \end{aligned} \tag{16}$$

Concerning the pair A–B, the throughput can be calculated as:

$$\rho_{AB} = \frac{\Pi_{s,AB}}{\Delta_{AB}}, \tag{17}$$

where

$$\Delta_{AB} = \Pi_{s,AB}\cdot T_{s,AB} + \Pi_{idle}\cdot\sigma + \Pi_c\cdot(T_c + T_{s,N}) + \Pi'_{b,N}\cdot T_{s,N}, \tag{18}$$

$\Pi_{s,AB}$ represents the probability that B transmits a packet successfully, and $\Pi'_{b,N}$ the probability that the channel is

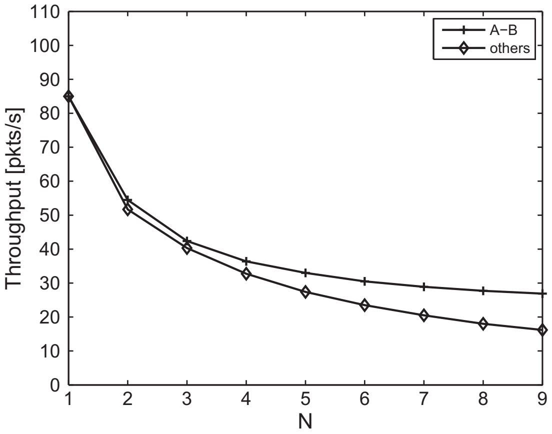


Fig. 10. Analytical value of CSMA/CARD throughput.

busy due to a transmission by the N nodes. Analogously to Eq. (15), these probabilities can be calculated as:

$$\Pi_{s,AB} = \tau_B \cdot (1 - \tau_N)^N$$
$$\Pi'_{b,N} = N \cdot \tau_N \cdot (1 - p).$$

Concerning the collision time, it is composed of the time needed to detect a collision, T_c, as calculated above, plus the time needed, by one of the N nodes, to perform a successful transmission, $T_{s,N}$.

In Fig. 10, we show the values of ρ_N and ρ_{AB} when the number of transmitters, N, varies in the range $\{1 \ldots 9\}$. Results obtained in the figure show that ρ_N and ρ_{AB} start very close to each other but, when the number of transmitters increases, the throughput calculated separates by as much as 10 packets per second. This difference is due to the increasing packet collision and the following increase of the average contention window, W, for the N transmitters which are disadvantaged with respect to node B which sends the RRTS using a CW_{min}.

5. Performance evaluation

In this section we consider the adaptive version of CSMA/CARD and we discuss the results obtained using the well-known ns-2 platform [31]. For comparison, the results for the CSMA/CA protocol specified in the IEEE 802.11 standard are also shown.

Each simulation was run for 100 s after a 5-s warm-up period. If not specified otherwise, backlogged traffic (i.e. each transmitter in the network has always packets to send) and stationary conditions were assumed. In all the simulations we supposed both sensing range and transmission range equal to 250 m, and we used for the physical layer the default values of the two-ray ground parameters.

Other simulation parameters are summarized in Table 2. Extensive simulations show that these values represent the optimal values for the considered parameters in order to obtain the best performance.

5.1. Simulation scenarios

We considered two different scenarios.

Table 2
Simulation parameters.

Parameter	Value
Slot time t_{slot}	20 µs
SIFS	10 µs
DIFS	50 µs
RTS size	36 bytes
RRTS size	36 bytes
CTS size	30 bytes
DATA size	1000 bytes
ACK size	30 bytes
CW_{min} (for CSMA/CARD)	[0,31]
CW_{min} (for CSMA/CA)	[0,31]
CW_{max}	1024
Data rate	2 Mb/s
$Threshold_{max}$	1
$Threshold_{min}$	0.1
K_+	1.2
K_-	0.8
P_{RRTS}	0.5
RRTS_no_replied_limit	4
RRTS_replied_limit	4

5.1.1. Scenario 1

This scenario is the same described in Section 4, Fig. 9. When CSMA/CA is used in this scenario, flow A–B is starved. If the basic mechanism of CSMA/CARD is used (assuming a contention window associated to the transmission of RRTS shorter than the standard one), we note that, when N increases, the flow A–B achieves a higher throughput as compared to the other flows. In particular, it moves from a starved state to a role of dominant flow. The adaptive CSMA/CARD, as shown in the following, solves this problem by properly adapting the probability to send RRTS.

5.1.2. Scenario 2

In this scenario we consider a sparse topology where 25 source–destination pairs are randomly distributed inside areas of different sizes (2000 m × 2000 m, 1500 m × 1500 m, 1000 m × 1000 m). Note that all the nodes have at least one neighbor and that channel contention, hidden terminals, and information asymmetry effects are present. This scenario allows us to describe the CSMA/CARD behavior and the throughput enhancements in comparison to CSMA/CA taking into account the density of nodes in the considered area and, consequently, the incidence of all the possible configurations which can occur in random cases.

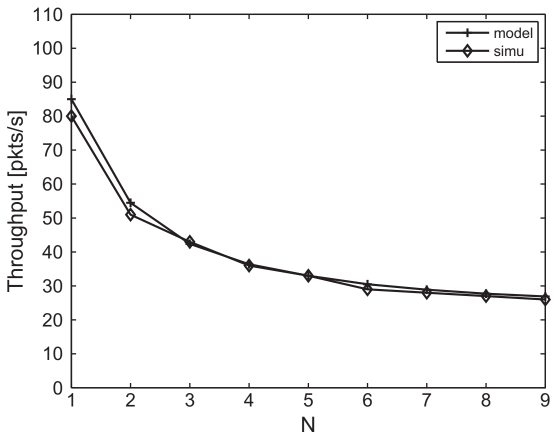
5.2. Results

First of all, we compare the analytical and simulation results and we discuss the choice of the value of the parameter k which tunes the duration of the *timeout*. Then, in order to show how the proposed approach outperforms the standard CSMA/CA protocol, we compare the flows' throughput and Jain's fairness index [13] for the scenarios shown above. In particular, our simulations highlight how, using CSMA/CA protocol, the starvation situation arises and many flows get very low or even zero throughput and how, instead, CSMA/CARD alleviates the problem of starvation in all the scenarios considered, increasing the throughput of disadvantaged connections and considerably reducing the number of starved flows.

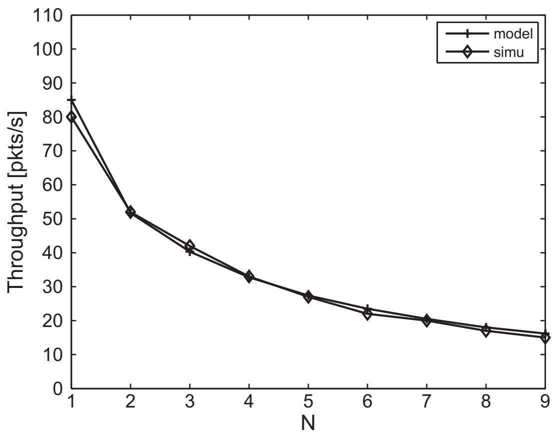
Moreover, CSMA/CARD allows to better distribute the bandwidth and, consequently, achieves higher values of fairness if compared to CSMA/CA.

5.2.1. Model validation

Fig. 11 shows the simulated and estimated values of throughput (pkts/s) obtained by means of the simulator and by means of Eqs. (14) and (17). We have considered Scenario 1, when the number of contenders, N, varies in the range $\{1 \ldots 9\}$. In particular, Fig. 11a shows the value of throughput for the flow A–B, while Fig. 11b shows the value of the average throughput for the other flows. First of all, the estimated and simulated values of throughput are very close to each other. Secondly, in some cases, simulated values are lower than estimated values, because the simulator takes into account several realistic factors, such as channel losses, which are not included in the analytical study.



(a) Flow A-B



(b) Other flows

Fig. 11. Comparison between model and experimental results in Scenario 1.

5.2.2. Choice of the timeout value

As we have already discussed in Section 3.3, a good choice of the *timeout* helps to solve the situations where a transmitter cannot answer an RRTS because is in a defer state due to other transmissions. If we assume that all nodes transmit packets of the same length and that in the average the collision occurs in the middle of a successful transmission, it is convenient to choose a value of k so that $k \cdot CW_{min}$ (which represents the additional waiting time) belongs to the interval $[T_s/2, T_s]$, where T_s is the time interval occupied by a successful transmission (see Section 4).

In Fig. 12, we present the value of throughput for the connection 3–4 shown in Fig. 5, when the parameter k varies in the range $\{1 \ldots 10\}$. As it can be observed, connection 3–4, without a good choice of parameter k, is starved. Moreover, a value of k greater than 6 does not increase further the value of throughput for the connection 3–4, rather it may excessively slow down the other connections. Therefore, in our simulations, we have chosen a value of k equal to:

$$k = [T_s/(2 * CW_{min})] + 1. \tag{19}$$

In the following, we evaluate the other parameters by considering the value of the *timeout* calculated as discussed above.

5.2.3. Throughput enhancements

In Fig. 13, we concentrate our attention on the throughput of connection A–B in Scenario 1, specifically, when the number of concurrent pairs increases (from 1 to 9). We observe that when the CSMA/CARD is used, the connection A–B no longer suffers starvation (i.e., throughput close to zero), reaching a value of throughput which is from three to ten times higher than the throughput obtained with CSMA/CA. Obviously, when the number of contending pairs increases, the throughput of pair A–B decreases, as well as the other pairs. In the following we analyze in detail the distribution of the bandwidth among the nodes.

Fig. 14, shows the number of flows starved (i.e., the number of flows having a throughput close to zero) using CSMA/CA and CSMA/CARD in 100 different random

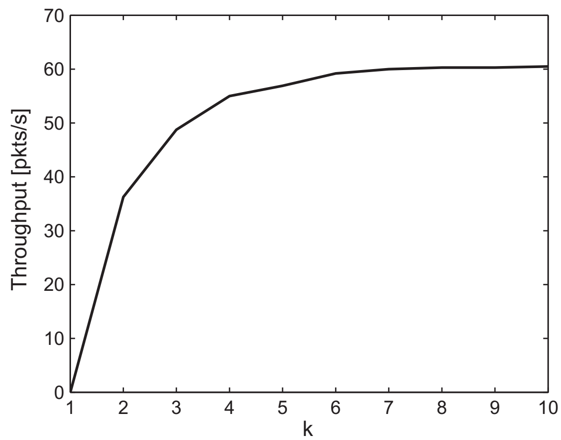


Fig. 12. Throughput of the flow 3–4 vs. k in the scenario shown in Fig. 5.

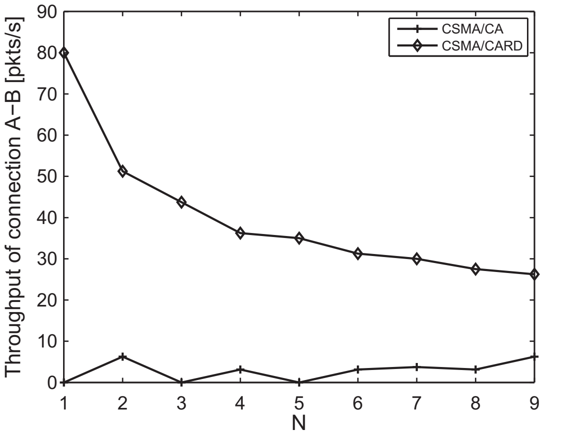


Fig. 13. Comparison between CSMA/CA and CSMA/CARD throughput for the connection *A–B* in Scenario 1.

distribution of nodes in Scenario 2, and for three different area sizes.

First of all, we observe that CSMA/CARD reduces both the maximum and the average number of flows starved. Moreover, we observe that, when CSMA/CA is used, and the area size is reduced, the number of flows starved increases. This happens because, in the considered scenario, the number of nodes contending for the channel increases and consequently the number of critical configurations which cannot be solved by the classical CSMA/CA (e.g. AIS and SIS) increases as well. Instead, when CSMA/CARD is used and the area size changes, the number of flows starved remains about the same (see the average number in the plots). The starved flows for each area size is due to some critical topologies like the flow in the middle [11], which can be solved only with approaches different from the CSMA/CA family (e.g. multichannel or out-of-band control channel).

Finally, in Fig. 15 we show the value of the throughput averaged among the five connections having the lowest value of throughput (the five connections more disadvantaged) for the three cases of Scenario 2, when both CSMA/CA and CSMA/CARD are used. As shown in this figure, CSMA/CARD alleviates the problem of starved flows, providing a higher value of the throughput for the disadvantaged connections in all the cases examined.

It is worth pointing out that the previous results take into account the effect of the *timeout* which acts on the nodes in the coverage area of the receiver.

5.2.4. Fairness index

We now show the fairness improvements achievable by using the CSMA/CARD in the two scenarios.

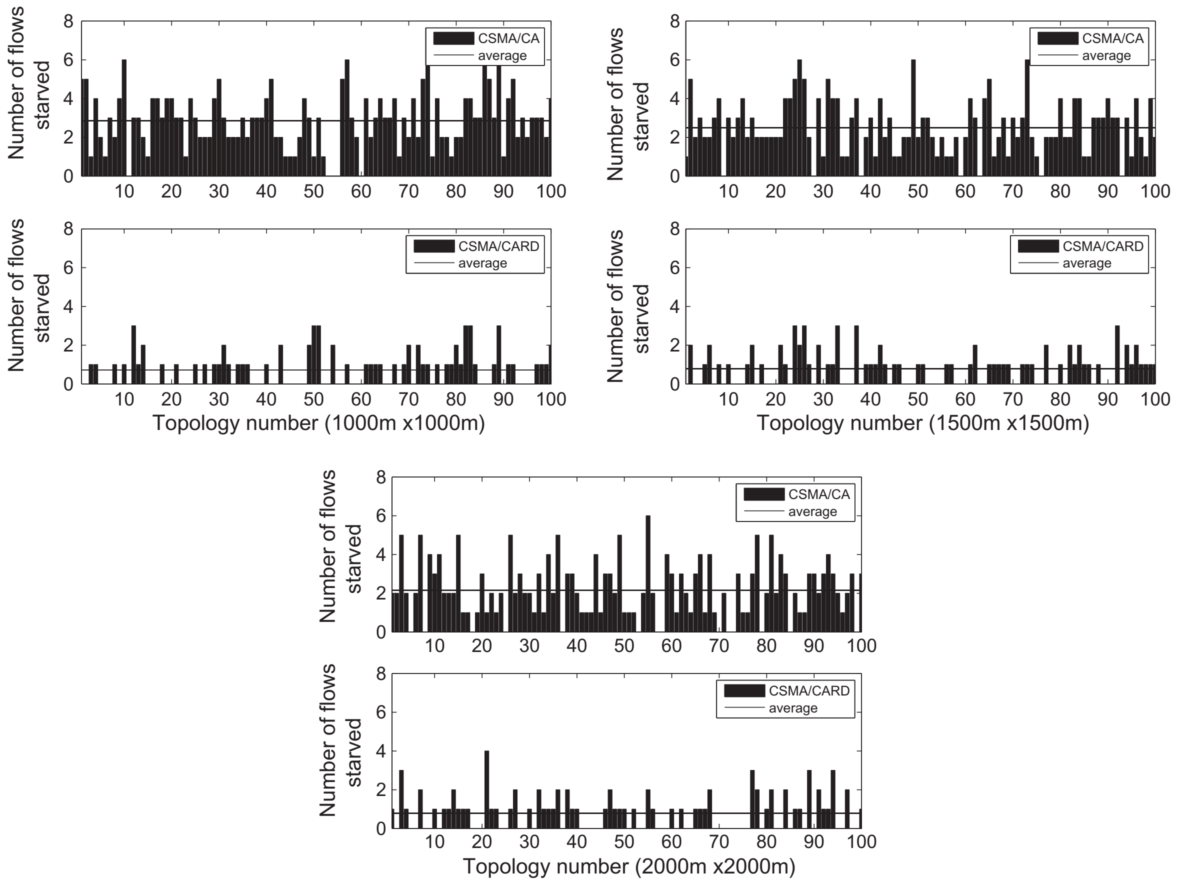


Fig. 14. Number of flows starved in Scenario 2 for 100 topologies and different area sizes.

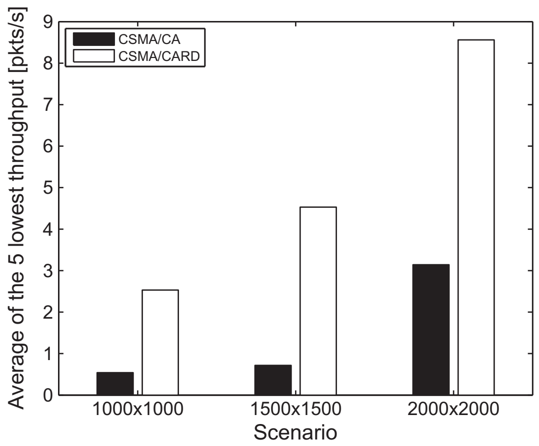


Fig. 15. Comparison between CSMA/CA and CSMA/CARD average minimum throughput in Scenario 2.

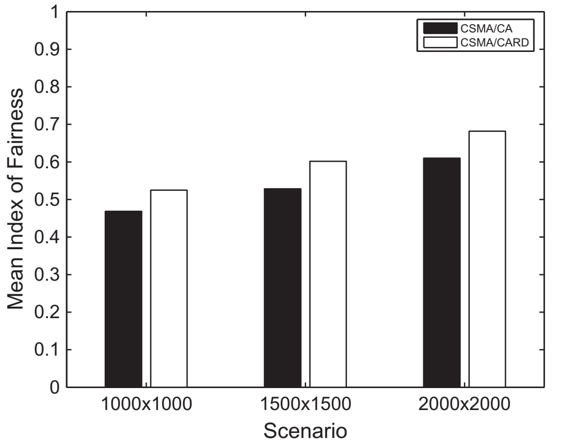


Fig. 17. Comparison between CSMA/CA and CSMA/CARD fairness index in Scenario 2.

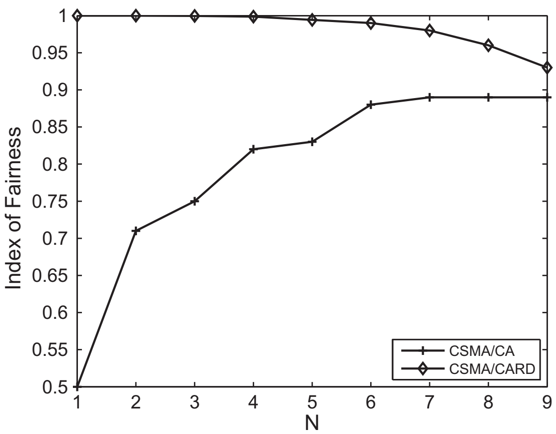


Fig. 16. Comparison between CSMA/CA and CSMA/CARD fairness index in Scenario 1.

In the literature several fairness indexes have been proposed. Here we consider Jain's fairness index [13] defined as follows:

$$\sigma(\tau_1, \tau_2, \ldots \tau_M) = \frac{\left(\sum_{i=1}^{M} \tau_i\right)^2}{M \sum_{i=1}^{M} \tau_i^2}, \quad (20)$$

where τ_i, with $i \leqslant M$, is the throughput for the ith connection.

When CSMA/CARD is used in Scenario 1, it avoids starvation of flow *A–B* allowing fair access to the medium even if the number of contending neighbors increases. Fig. 16 shows how CSMA/CARD outperforms CSMA/CA, reaching values of fairness index approximately equal to 1 also for a large number of contending senders in the range of receiver 2. When the number of nodes, *N*, increases, the index of fairness decreases. This is due to the behavior of the flow *A–B* which becomes dominant in the transmissions because it always uses a minimum congestion window value, CW_{min}.

We also note that, when the number of contending pairs increase, CSMA/CA fairly distributes the bandwidth among the remaining transmitters (in the same range) even if the flow *A–B* remains starved (see Fig. 13).

In Fig. 17 we compare the values of fairness index using CSMA/CARD and those obtained using CSMA/CA for the three cases of Scenario 2. As expected, the CSMA/CARD behaves better than the CSMA/CA protocol; moreover, the fairness index decreases for both protocols when the density of nodes, and consequently the incidence of critical configurations, such as asymmetric incomplete state and flow in the middle, increases.

6. Conclusions

In this paper a new mechanism, called CSMA/CARD, has been introduced. This solution is based on a novel receiver-initiated mechanism which exploits some information at the physical level. More specifically, collision sensed by the receiver can be used to predict whether some sender attempted to initiate a transmission towards that receiver. Using this information, the receiver can initiate an action, by sending some packets called RRTS, to help to expedite the handshake mechanism and to avoid the starvation of some flows.

We have discussed an adaptive version of this mechanism which, according to traffic conditions occurring in the neighborhood, tunes the number of RRTS to be sent in order to reduce extra collisions due to RRTS packets, and consequently improves the performance. The results obtained show that the proposed mechanism alleviates the problem of starvation in all the scenarios considered.

References

[1] J.J. Garcia-Luna-Aceves, A. Tzamaloukas, Receiver-initiated collision avoidance in wireless networks, ACM Wireless Networks 8 (2002) 249–263.

[2] B. Bensaou, Y. Wang, C.C. Ko, Fair medium access in 802.11 based wireless ad-hoc networks, in: Proceedings of MobiHoc 2000, Boston, MA, USA, August 2000.

[3] V. Bharghavan, A. Demers, S. Shenker, L. Zhang, MACAW: a media access protocol for wireless LAN's, in: Proceedings of SIGCOMM '94, London, UK, September 1994.
[4] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal on Selected Areas in Communications 18 (3) (2000) 535–547.
[5] M. Cesana, D. Maniezzo, P. Bergamo, M. Gerla, Interference aware (IA) MAC: an enhancement to IEEE802.11b DCF, in: Proceedings of IEEE VTC '03, Orlando, Florida, USA, October 2003.
[6] K. Chen, Medium access control of wireless LANs for mobile computing, IEEE Network 8 (5) (1994) 50–63.
[7] J. Deng, Z. Haas, Dual busy tone multiple access (DBTMA) – a multiple access control scheme for ad hoc networks, IEEE Transactions on Communications 50 (6) (2002).
[8] J. Deng, P.K. Varshney, J. Haas, A new backoff algorithm for the IEEE 802.11 distributed coordination function, in: Proceedings of Communication Networks and Distributed Systems Modeling and Simulations (CNDS 04), San Diego, CA, USA, January 2004.
[9] C.L. Fullmer, J. Garcia-Luna-Aceves, FAMA-PJ: a channel access protocol for wireless LANs, in: Proceedings of ACM MobiCom '95, Berkeley, CA, USA, November 1995.
[10] M. Garetto, J. Shi, E. Knightly, Modeling media access in embedded two-flow topologies of multi-hop wireless networks, in: Proceedings of ACM MobiCom 2005, Cologne, Germany, August 2005.
[11] M. Garetto, T. Salonidis, E. Knightly, Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks, in: Proceedings of IEEE INFOCOM 2006, Barcelona, Spain, April 2006.
[12] X. Huang, B. Bensaou, On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation, in: Proceedings of ACM MobiHoc '01, Long Beach, CA, USA, October 2001.
[13] R. Jain, D. Chiu, W. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer systems, DEC Research Report, September 1984.
[14] Y. Jian, S. Chen, Can CSMA/CA networks be made fair? in: Proceedings of ACM MobiCom'08, San Francisco, CA, USA, September 2008.
[15] P. Karn, MACA – a new channel access method for packet radio, in: Proceedings of ARRL/CRRL Amateur Radio 9th Computer Networking Conference, September 1990.
[16] S.M. Kay, Fundamentals of Statistical Signal Processing – Detection Theory, Prentice-Hall, 1998.
[17] L. Kleinrock, F.A. Tobagi, Packet switching in radio channels, IEEE Transactions on Communication 23 (12) (1975) 1400–1416.
[18] B.M. Leiner, D.L. Nielson, F.A. Tobagi, Issues in packet radio design, IEEE Proceedings 75 (1987).
[19] A. Leonardi, S. Palazzo, A. Kabbani, E. Knightly, Exploiting physical layer detection techniques to mitigate starvation in CSMA/CA wireless networks, in: Proceedings of IEEE WONS'07, Obergurgl, Austria, January 2007.
[20] J.P. Monks, V. Bharghavan, W.-M.W. Hwu, A power controlled multiple access protocol for wireless packet networks, in Proceedings of IEEE INFOCOM '01, Alaska, April 2001.
[21] T. Ozugur, M. Naghshineh, P. Kermani, C.M. Olsen, B. Rezvani, J.A. Copeland, Balanced media access methods for wireless networks, in: Proceedings of ACM MobiCom'98, Dallas, Texas, USA, October 1998.
[22] J. Peng, L. Cheng, B. Sikdar, A wireless MAC protocol with collision detection, IEEE Transactions on Mobile Computing 6 (12) (2007).
[23] K.K. Ramakrishnan, H. Yang, The ethernet capture effect: analysis and solution, in: Proceedings 19th Conference on Local Computer Networks (LCN '94), Minneapolis, MN, USA, October 1994.
[24] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, S Banerjee, Diagnosing wireless packet losses in 802.11: separating collision from weak signal, in: Proceedings of INFOCOM 2008, Phoenix, AZ, USA, April 2008.
[25] S. Sen, R.R. Choudhury, S. Nelakuditi, CSMA/CN: carrier sense multiple access with collision notification, in: Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking (MobiCom '10), Chicago, IL, USA, September 2010.
[26] F. Talucci, M. Gerla, MACA-BI (MACA by invitation): a receiver oriented access protocol for wireless multihop networks, in: Proceedings of PIMRC'97, Oulu, Finland, 1997.
[27] C. Wu, V.O.K. Li, Receiver-initiated busy-tone multiple access in packet radio networks, in: Proceedings of ACM SIGCOMM '87 Workshop: Frontiers in Computer Communications Technology, Stowe, VT, USA, 1987.
[28] S.-L. Wu, Y.-C. Tseng, C.-Y. Lin, J.-P. Sheu, A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks, Computer Journal 45 (1) (2002).
[29] J.-H. Yun, S.-W. Seo, Collision detection based on RF energy duration in ieee 802.11 wireless LAN, in: Proceedings of Comsware, 2006, Delhi, India, January 2006.
[30] P802.11 – Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications, IEEE, January 1996.
[31] Network Simulator – ns2 – <http://www.isi.edu/nsnam/ns/>.

Alessandro Leonardi received the Laurea Degree in Electronics Engineering and the Ph.D. in Computer and Telecommunications Engineering from the University of Catania, Italy, in 2002 and 2007, respectively. Since 2002 he is a member of CNIT (National Inter-University Consortium for Telecommunications). From September 2005 to February 2006 he has been visiting the Rice Networks Group (Rice University, Houston TX, USA) under the guidance of Prof. Edward Knightly. Currently he is a Post Doctoral Researcher with the Dipartimento di Ingegneria Elettrica, Elettronica ed Informatica (DIEEI) of the University of Catania. His current research interests include protocols and algorithms for mobile, ad hoc, sensor and mesh networks.

Sergio Palazzo was born in Catania, Italy, on December 12, 1954. He received his degree in electrical engineering from the University of Catania in 1977.
Until 1981, he was at ITALTEL, Milano, where he was involved in the design of operating systems for electronic exchanges. He then joined CREI, which is the center of the Politecnico di Milano for research on computer networks. Since 1987 he has been at the University of Catania, where is now a Full Professor of Telecommunications Networks. In 1994, he spent the summer at the International Computer Science Institute (ICSI), Berkeley, as a Senior Visitor. He is a recipient of the 2003 Visiting Erskine Fellowship by the University of Canterbury, Christchurch, New Zealand.
Since 1992, he has been serving on the Technical Program Committee of INFOCOM, the IEEE Conference on Computer Communications. He has been the General Chair of the ACM MobiHoc 2006 Conference and currently is a member of the MobiHoc Steering Committee. In the recent past, he has been the Program Co-Chair of the 2005 International Tyrrhenian Workshop on Digital Communications, focused on "Distributed Cooperative Laboratories: Networking, Instrumentation, and Measurements", the General Vice Chair of the ACM MobiCom 2001 Conference, and the General Chair of the 2001 International Tyrrhenian Workshop on Digital Communications, focused on "Evolutionary Trends of the Internet".
He currently serves the Editorial Board of the journal Ad Hoc Networks. In the recent past, he also was an Editor of IEEE Wireless Communications Magazine (formerly IEEE Personal Communications Magazine), IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, Computer Networks, and Wireless Communications and Mobile Computing. He was a Guest Editor of Special Issues in the IEEE Journal of Selected Areas in Communications ("Intelligent Techniques in High-Speed Networks"), in the IEEE Personal Communications Magazine ("Adapting to Network and Client Variability in Wireless Networks"), in the Computer Networks journal ("Broadband Satellite Systems: a Networking Perspective"), in the EURASIP Journal on Wireless Communications and Networking ("Ad Hoc Networks: Cross-Layer Issues"). He also was the recipient of the 2002 Best Editor Award for the Computer Networks journal.
His current research interests include mobile systems, wireless and satellite IP networks, intelligent techniques in network control, multimedia traffic modeling, and protocols for the next generation of the Internet.

Corrado Rametta received the Laurea Degree in Electronics Engineering from the University of Catania, Italy in 2008.
Currently he is a PhD student with the Dipartimento di Ingegneria Ingegneria Elettrica, Elettronica ed Informatica (DIEEI) of the University of Catania. His current research interests include protocols and performance evaluations of wireless networks.

Edward Knightly is a Professor of Electrical and Computer Engineering at Rice University in Houston, Texas. He received his Ph.D. and M.S. from the University of California at Berkeley and his B.S. from Auburn University. He is an IEEE Fellow, a Sloan Fellow, and a recipient of the National Science Foundation CAREER Award. He received the best paper award from ACM MobiCom 2008. His research interests are in the areas of mobile and wireless networks and high-performance and denial-of-service resilient protocol design. He leads the Rice Networks Group. The group's current projects include deployment, operation, and management of a large-scale urban multihop multitier IEEE 802.11 network in a Houston under-resourced community. This network, TFA Wireless, is serving over 4,000 users in several square kilometers and employs custom-built programmable and observable access points. The group is also developing a clean-slate-design hardware platform for highperformance multi-hop wireless. The TAPs/WARP platform is now operational, and ongoing research includes cross-layer protocol design and implementation.