

A Fast-and-Fair Collision Resolution Protocol Tailored for Ad Hoc Wireless Networks

Marcos F. Caetano, Lucas M. Guimaraes and Jacir L. Bordim

Department of Computer Science, University of Brasilia, Brasilia, Brazil

Article history

Received: 18-04-2016

Revised: 24-10-2016

Accepted: 28-10-2016

Corresponding Author:

Marcos F. Caetano
Department of Computer
Science, University of Brasilia,
Brasilia, Brazil
Email: mfcaetano@unb.br

Abstract: Wireless technologies have permeated a significant number of devices over the years. On the other hand, Medium Access Control (MAC) protocols strive to deal with an increasing number of contending nodes on crowded frequency bands. This work presents an efficient medium access control resolution protocol. The proposed protocol, termed Collision Resolution Protocol (CRP), uses *pulse/tone* signaling that works in both Collision Detection (CD) and Non-Collision Detection (NCD) environments. Given $n > 1$ contending nodes, CRP selects a single transmitting node in $16n$ and $40n$ time slots, respectively, for the CD and the NCD scenarios, with probability of at least $1 - 2^{-1.5n}$. Furthermore, CRP does not require an exact number, or estimation, of the contending nodes. Extensive simulations on the OMNeT++ shows that, for $n = 256$ and considering CD-capability, CRP successfully grants channel access in 0.03% of the time required by the IEEE 802.11 standard while improving the number of transmissions per second in 100%. In addition, CRP provides fair resource allocation and reduced channel access latency.

Keywords: Ad Hoc Networks, Channel Access, Contention Resolution, MAC Protocols

Introduction

Owing to its flexibility, low-cost and easy-deployment, wireless technologies have become commonplace in a wide-range of consumer electronics, such as smart phones, cameras and e-readers. Likewise, ad hoc networks have gained significant attention as they allow wireless devices to communicate with each other without the need of a fixed and usually costly network infrastructure. Ad hoc networks usually comprise of several channels, which are shared among the contending nodes (Perkins, 2008). Packet collision occurs when two or more adjacent nodes attempt to transmit concurrently on the common channel. In such case, these packets are garbled beyond recognition. An increase on the number of contending nodes usually incurs on high channel access delays and reduced network throughput (Kumar *et al.*, 2006). To decrease the chance of packet collision, various Medium Access Control (MAC) protocols have been proposed in the literature (Bharghavan *et al.*, 1994; 802.11b Working Group, 1997). Backoff strategies have been employed such that nodes wishing to communicate need to wait their turn before transmitting. The IEEE

802.11 standard (802.11b Working Group, 1997) applies the Binary Exponential Backoff (BEB) technique to decrease the chance of collisions among contending nodes. When a node transmits, the contending nodes interrupt their backoff counter until the channel becomes idle. After that, these nodes continue to defer their backoff counter. The IEEE 802.11 provides acceptable results for a limited number of contending nodes (Bianchi, 2000).

Mobile devices enhanced with wireless technologies have increased significantly in the last few years. Such increase exacerbates the demands more elaborated solutions to comply with the application's Quality of Service (QoS) constraints. Thus, several random access control protocols have been proposed in the literature (Bharghavan *et al.*, 1994; 802.11b Working Group, 1997; Song *et al.*, 2003; Venkitaraman *et al.*, 1999). Control frames, such as Request to Send (RTS) and Clear to Send (CTS), have been used to reduce the effects of hidden terminal problem (802.11b Working Group, 1997). The Dual Busy Tone Multiple Access (DBTMA) (Haas and Deng, 2002) and Busy Tone Multiple Access (BTMA) (Kleinrock and Tobagi, 1975) scheme uses BEB coupled with out-of-band signaling (a.k.a.

pulse/tone) to effectively address the hidden and exposed terminal problems. Ma *et al.* (2012), the authors use *pulse/tone* signals to reduce the collision probability of RTS/CTS control frames. Recently, (Guimarães *et al.*, 2015) proposed the use of *pulse/tone* signaling to reduce the overhead of control frames on IEEE-based MAC protocols. In (Lu *et al.*, 2015), the authors successfully implemented *pulse/tone* signals on a test-bed using the Microsoft Sora Platform for Cognitive Radios Networks (CRN) (Tan *et al.*, 2009).

Tree-based algorithms (split, stack or collision resolution) usually implement a partitioning strategy, allowing nodes to receive a free slot for communication (Wattanamongkhol, 2013). Mathys and Flajolet (2006) presented a group and split approach, called Q-ary split. The main idea is to gather the colliding stations in $Q > 2$ groups and employ a split approach to reduce collision probability in each group. Wattanamongkhol (2013) proposes the Uniform Selection (UNI) protocol where the contending stations uniformly elect a time slot for transmission. The outcome of their attempt is later propagated to the contending stations. Such information is then used to reduce the chance of collision. Tree-based algorithms usually assume that the number of contending stations is known a priori (Wattanamongkhol, 2013; Mathys and Flajolet, 2006). Nevertheless, on a distributed environment, it may be not possible to obtain a precise number, or even an estimate, of the contending stations (Nakano and Olariu, 2002). Felice *et al.* (2013), the authors attempt to determine the number of contending nodes by monitoring busy slots. However, their solution is not able to deal with a large number of transmitting stations.

This work presents the CRP, a collision resolution protocol that provides collision free transmission of data frames while providing for a fast selection of a transmitter among $n > 1$ contending nodes. CRP incorporates some of the features of the aforementioned tree based algorithms. Nevertheless, CRP does not require a prior knowledge on the number of contending stations. Similarly to the works in (Haas and Deng, 2002; Ma *et al.*, 2012; Guimarães *et al.*, 2015), CRP takes advantage of pulse/tone signaling to elect a transmitter among a group of contending stations. To assess the goodness of the proposed scheme, we first consider the case in which the nodes are equipped with collision detection (CD) mechanisms (Tseng *et al.*, 2002). As it will be explained in the subsequent sections, a CD-enhanced node can properly obtain the channel status in the current time slot. In particular, a CD-enabled station can determine if its own transmission resulted in collision or not. We prove that, given $n > 1$ CD-enabled stations, CRP can effectively select a transmitter among the contending nodes in $16n$ time slots with probability of at least $1 - 2^{-1.5n}$. Next, CRP is

modified to handle the case in which the nodes are not enhanced with CD-capabilities. We show that, even when the nodes are not enhanced with CD capabilities, CRP can effectively select a transmitter among $n > 1$ contending nodes in $40n$ time slots with probability of at least $1 - 2^{-1.5n}$. Lastly, CRP is compared with the IEEE 802.11 BEB mechanism using the OMNET++ simulator (OC, 2016a). Simulation results show that CRP outperforms BEB.

The remainder of this paper is organized as follows: Section 2 presents the network communication model considered in this work. Section 3 begins by presenting a fast and fair collision resolution mechanism, termed CRP, for the case where the nodes are empowered with collision detection capabilities. Next, the CRP is extended to deal with the non-collision detection scenario. Section 4 shows the simulation environment and performance assessment of the proposed mechanism. Finally, Section 5 presents the conclusions and directions for further investigation.

Communication Model

This section presents the communication model and proposed Collision Resolution Protocol. The network is defined as a complete, undirected, graph $G = (V, E)$, where E is a set of edges (links) and $V = \{v_1, v_2, \dots, v_n\}$ is a set of nodes (vertices), with $E \subseteq V^2$. Each node is equipped with a radio transceiver, which implements the IEEE802.11 standard. As in (Haas and Deng, 2002; Shih *et al.*, 2009; Ma *et al.*, 2012), nodes are reassumed to be capable of receiving and sending data packets along with *pulse/tone* signals. *Pulse/Tone* signals have been used to notify nearby nodes about a particular condition of the network, when the identity of the source node is irrelevant. Different from control and data frames, *pulse/tone* signals contain no information (Deng and Haas, 1998). Hence, they can be transmitted and received faster than data and control frames. According to (Kleinrock and Tobagi, 1975), *pulse/tone* can be properly identified if they are transmitted for at least $5\mu s$. In this study, time is assumed to be slotted, matching the duration of a *pulse/tone* signal. Data packets, on the other hand, may require several time slots to be correctly transmitted/received.

This work assumes that the stations maintain a certain level of synchronization, similar to that considered in (Wattanamongkhol, 2013; Mathys and Flajolet, 2006; Nakano and Olariu, 2002; Felice *et al.*, 2013; Zhou and Lai, 2005). For instance, synchronization could be obtained using a mechanism like the Self-Adjusting Timing Synchronization Function (SATSF) algorithm (Zhou and Lai, 2005). The SATSF is compatible with IEEE 802.11 Timing Synchronization Function (TSF).

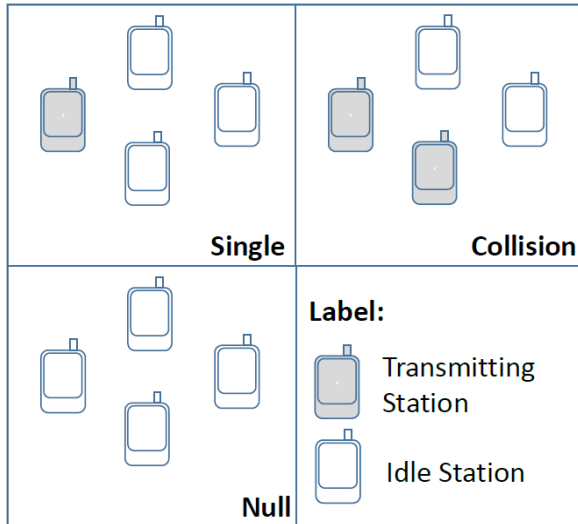


Fig. 1. Channel status at the end of a time slot

At any time, a node's transceiver v_i , ($1 \leq i \leq n$), can be: (i) transmitting; (ii) receiving; or (iii) idle. Furthermore, when two or more nodes transmit at the same time, the intended receiver fails to correctly decode the transmitted information (Tseng *et al.*, 2002). As illustrated in Fig. 1, the status of a channel, at the end of a time slot, is defined as:

- SINGLE: Only one node transmits; or
- COLLISION: Two or more nodes transmit; or
- NULL: No node transmits

A station equipped with Collision Detection (CD) is capable of correctly estimate the status of the current time slot of a channel. In other words, a CD-enabled node v_i ($1 \leq i \leq n$), after its own transmission, can distinguish whether the channel status is SINGLE, COLLISION or NULL. Nevertheless, wireless stations are typically built with half-duplex antennas. For this reason, after transmitting, the transmitter cannot determine the channel status. That is, a transmitting station without CD capabilities (NCD-node) is not able to identify the resulting channel status after its own transition. The proposed CRP, works for both CD and NCD nodes.

Collision Resolution Protocol

In this section, the CRP protocol is presented. The CRP uses *tone* signal to develop a data frame, collision-free, channel control access. CRP implements an effective and fast scheme for transmitter selection. To achieve this, CRP continuously reduces the number of contending stations until a single transmitter is selected. From an initial group of nodes G_1 , where $|G_1| > 1$, CRP is able to always select a smaller subset of nodes until a winner is selected. This process is illustrated in Fig. 2.

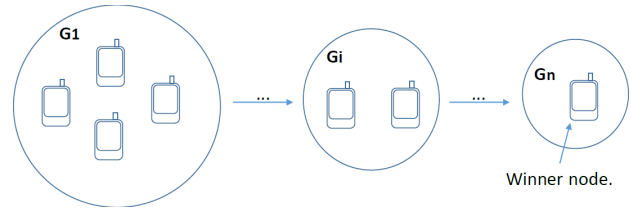


Fig. 2. From an initial group $|G_1| > 1$, CRP continuously reduces the number of contending stations until $|G_i| = 1$ where a winner is declared

The next subsection presents the CRP protocol for the scenario in which the nodes are empowered with Collision Detection capabilities (CD). When this is the case, we show that the CRP is able select a station among $n > 1$ contending nodes in $16n$ time slots with probability of at least $1 - 2^{-1.5n}$. We then go on to extend the CRP to the case in which the nodes have Non-Collision Detection (NCD) capabilities. We show that the CRP for NCD is able to select a station among $n > 1$ contending nodes in $40n$ time slots with probability of at least $1 - 2^{-1.5n}$. The details of the CRP for the CD scenario are presented below.

CRP for Collision Detection Scenario

As mentioned in Section 2, simultaneous transmission of two or more nodes leads to packet loss. Suppose that, at a given timeslot T , a subset $S \geq 2$ of nodes have data items ready to be transmitted. To guarantee a collision-free transmission of a data frame, a single node must be selected to transmit at a time. To achieve this, CRP implements the *GroupSplitting-CD* routine. As it will be shown later, the problem of selecting a transmitter can be computed in $O(n)$ time slots with probability at least $1 - 2^{-1.5n}$.

Let $S \geq 2$ denote the number of nodes contending for channel access at a time. The *GroupSplitting-CD* procedure divides the nodes in S into non-empty subsets S_1 and S_2 . The subset S_2 is then divided into two non-empty subsets S_2 and S_3 . The same procedure is then applied to S_3 . Generalizing, a subset S_i is divided into non-empty subsets S_i and S_{i+1} . This process is repeated until a subset containing a single element is found ($|S_i| = 1$). Let v_w be remaining station in station in S_i . Then, station v_w is proclaimed the *winner* of the contention period. After that, all nodes learn that a winner was correctly selected and leave the contention phase. These nodes postpone their transmission until the next iteration of transmission selection, which will take place after node v_w completes its data frame transfer. One can confirm that routine *GroupSplitting-CD* divides the set S_i into two non-empty subsets, S_i and S_{i+1} . Recall from Section 2 that every CD-enabled node knows whether $|S_i| = 1$ or $|S_i| \geq 2$. Likewise, it is possible to determine whether $|S_{i+1}| = 1$ or $|S_{i+1}| \geq 2$.

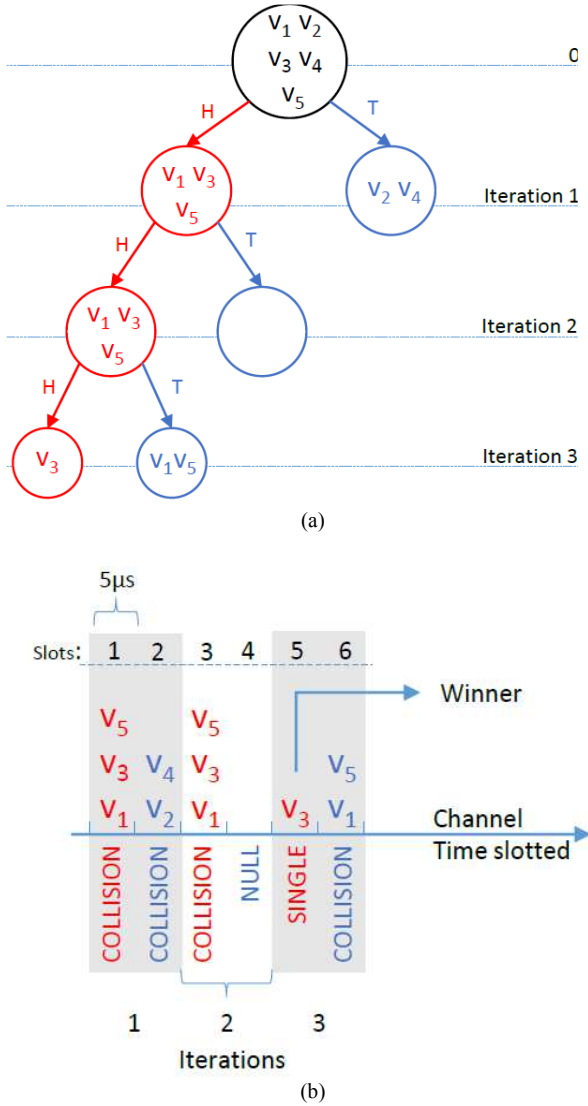


Fig. 3. An example of the CRP-CD execution for a group of 5 contending stations

GroupSplitting-CD uses *Status(head)* and *Status(tail)* to register the resulting channel status at the end of a time slot. Note that, during the partitioning process, tone transmissions always occur at different time slots, for instance: n and $n+1$ time slots (routine 1, lines 3 and 5).

Figure 3 presents an example of the CRP protocol's behavior. Figure 3a shows the splitting operation process for an initial group S , as defined in Routine 1. Figure 3b shows the mapping process throughout the execution of the CRP protocol. At the beginning of the protocol, all nodes in S are active. Whenever a winner is elected, all nodes exit the actual contention phase. According to the example, the initial subset S is formed by 5 contending stations v_1, v_2, v_3, v_4 and v_5 . In the first iteration, Figure 3a, the

stations executing Routine 1 are divided into two subsets: $S_i = \{v_1, v_3, v_5\}$ (*head*) and $S_{i+1} = \{v_2, v_4\}$ (*tail*). Figure 3b denotes the channel state at the end of the first and the second slot (1st iteration). In the second iteration, the subset S_i is divided into an unbalanced way. That is, the subset $S_{i+1} = \{v_1, v_3, v_5\}$ and $S_{i+2} = \emptyset$. These unbalanced subsets return true at the condition test $Status(tail) == NULL$ in Routine 1 (line 7), forcing another subdivision of the subset S_i . The third iteration divides the subset S_i in two non-empty subsets $S_{i+1} = \{v_3\}$ (*head*) and $S_{i+2} = \{v_1, v_5\}$ (*tail*). As $|S_{i+1}| = 1$, station v_3 declares itself as the winner of the current contention period. At this point, all stations wait for the channel allocation process, which is initialized by station v_3 . This process is further discussed in section 3.3.

The details of the CRP protocol are shown in Routine 2. Notice that the task of verifying whether $|S_i| = 1$ can be done through the channel status acquired in *GroupSplitting-CD*, without any further transmission. Next, we compute the time slots involved in routine *GroupSplitting-CD*. Assume that m , ($m \geq 2$), nodes are to be divided into two non-empty sets using routine *GroupSplitting-CD*. In this case, an iteration of the while-loop, in routine *GroupSplitting-CD*, is considered *successful* if it succeeds in dividing the set into two non-empty subsets. Let m be the random variable representing the number of nodes that chosen head. Since $m \geq 2$, the probability of an iteration to be successful can be expressed as:

$$\begin{aligned} \Pr[1 \leq X \leq m-1] &= 1 - \Pr[X = 0] - \Pr[X = m] \\ &= 1 - 2\Pr[X = 0] = 1 - \frac{1}{2^{m-1}} \geq \frac{1}{2} \end{aligned}$$

Routine 1 *GroupSplitting-CD*

- 1: **Do**
- 2: each node in S_i tosses a fair coin;
- 3: all nodes that tossed *head* transmit a *tone* on the channel;
- 4: let *Status(head)* be the resulting status of the channel;
- 5: all nodes that tossed *tail* transmit a *tone* on the channel;
- 6: let *Status(tail)* be the resulting status of the channel;
- 7: **While** (*Status(head)* == NULL or *Status(tail)* == NULL)
- 8: S_i = all nodes that have tossed *head*;
- 9: S_{i+1} = all nodes that have tossed *tail*;

Routine 2 Collision Resolution Protocol (CRP)

- 1: all nodes set $i = 1$;
- 2: **while** $i \geq 1$ **do**
- 3: **if** $|S_i| = 1$ **then**
- 4: the unique node in S_i is declared winner;
- 5: all nodes leave the protocol;
- 6: **else**
- 7: call to *GroupSplitting-CD* partition S_i ;
- 8: all active nodes set $i = i + 1$
- 9: **end if**
- 10: **end while**

Since a successful iteration generates two nonempty subsets, it is obvious that CRP demands, in the worst-case scenario, $\log n$ successful iterations of routine *GroupSplitting-CD*. Next, we evaluate the number of time slots to guarantee, with high probability, the existence of $\log n$ successful iterations of *GroupSplitting-CD*. Let Y be the random variable defining the number of successes among $8n$ Bernoulli trials, with parameter $p = 1/2$. Let $E[Y]$ denote the expected value of Y . Clearly, $E[Y] = np = 4n$. To analyze the tail of the binomial distribution, we use of the following estimate, commonly referred to as *Chernoff bound* (Motwani and Raghavan, 1995):

$$\Pr[Y \leq (1-\epsilon)E[Y]] \leq e^{-\frac{\epsilon^2}{2}E[Y]} \quad (0 \leq \epsilon \leq 1) \quad (1)$$

By virtue of (1) with $\epsilon = 3/4$, the probability that $8n$ trials contain fewer than $\log n$ successes are:

$$\begin{aligned} \Pr[Y < \log n] &< \Pr[Y \leq \left(1 - \frac{3}{4}\right)E[Y]] \\ &\leq e^{-\frac{9}{32}E[Y]} = e^{-\frac{9}{8}n} < 2^{-1.5n} \end{aligned}$$

We just proved that with probability at least $1 - 2^{-1.5n}$, among the first $8n$ iterations of the while-loop in routine *GroupSplitting-CD* there must exist at least $\log n$ successful iterations. Since each while-loop takes two time slots, the CRP protocol terminates in $8n \times 2 = 16n$ time slots with probability at least $1 - 2^{-1.5n}$. The above discussion is summarized below.

Lemma 1

The CRP Protocol can Elect a Transmitting Node Among n CD-Enabled Contending Stations, in a Single-Hop Ad Hoc Network, in at Most $16n$ Time Slots with Probability at Least $1 - 2^{-1.5n}$

Our next task is to show that the CRP can be extended to handle the case in which the contending nodes are not empowered with CD capabilities.

CRP for Non-Collision Detection Scenario

This section enhances the CRP protocol for handling the Non-Collision Detection (NCD) scenario. Recall from Section 2 that a NCD-station is not able to recognize the status of the channel after its transmission. That is, after transmitting a packet, the transmitting NCD-station cannot specify whether the resulting channel status is SINGLE, COLLISION or NULL.

Routine 3 *GroupSplitting-NCD*

```

1:   Do
2:     each node in  $S_i$  tosses a fair coin;
3:      $H_i$  = all nodes that have tossed head;
4:      $T_i$  = all nodes that have tossed tail;
5:     all nodes in  $H_i$  transmit a tone on the channel;
6:     all nodes in  $T_i$  record the channel status in their
       Status(head);
7:     all nodes in  $T_i$  call Channel(Status(head));
8:     all nodes in  $T_i$  transmit a tone on the channel;
9:     all nodes in  $H_i$  record the channel status in their
       Status(tail);
10:    all nodes in  $H_i$  call Channel(Status(tail));
11:    While (Status(head) == NULL or Status(tail) == NULL)
12:       $S_i = H_i$ ;
13:       $S_{i+1} = T_i$ ;
    
```

Routine 4 Channel(Status(HT))

```

1:   all the stations check the resulting the content of Status(HT) as
       follows:
2:   if (Status(HT) == COLLISION) then
3:     all the stations transmit a tone for one time slot;
4:   else if (Status(HT) == NULL) then
5:     all the stations are silent for two time slots;
6:   else if (Status(HT) == SINGLE) then
7:     all the stations are silent for one time slot and transmit a
       tone for one time slot;
8:   end if
    
```

Thus, the aim of this section is to enhance the CRP to tackle the above issue. For this purpose, the *GroupSplitting-CD* routine is revised to allow NCD-nodes to acquire the status of the channel. The details of the NCD version of the *GroupSplitting-CD* routine, termed as *GroupSplitting-NCD*, are shown in Routine 3. The Do While-loop in Routine *GroupSplitting-NCD* can be viewed as two main phases with distinct goals: *Notification* and *transmission*. Starting with the transmission phase, an iteration of Routine 3 splits the nodes into two nonempty groups H_i and T_i . The stations in H_i transmit on the channel while the stations in T_i monitor and record the resulting status of the channel. After that, the monitoring group notifies the transmitting group about the resulting channel status recorded. In the notification phase, the challenge is to successfully notify the transmitting stations about the resulting channel status overheard. As discussed in section 2, a *tone* signal carry no information. In this context, the codification of the NULL, SINGLE or COLLISION status are made varying the transmission time of the *tone* signal transmissions.

Clearly, three possible states can be encoded using two regular times slots of $5\mu s$ each in an obvious way. Routine 4 shows the details of the encoding scheme. When the observed resulting channel status is SINGLE, the notification process is as presented in the Fig. 4. That is, as the transmission phase is detected as SINGLE, the stations wait one slot and transmit a *tone* signal in the second one. In case of COLLISION, the stations transmit a *tone* for one time slot only, as shown in Fig. 5. In case

the resulting channel status is NULL, the stations remain silent for two consecutive time slots (Fig. 6). The above scheme allows the stations in the transmitting group to know the result status of the channel after their transmissions. Clearly, Routine 3 can be executed by a station without Collision Detection capability.

From the above discussion, one can confirm that each group (*head/tail*) is capable to determine the resulting channel status after its transmission. That is, the routine *GroupSplitting-NCD* is able to partition the input set S_i into two non-empty subsets, S_i and S_{i+1} . That is, each node knows whether $|S_i|=1$ or $|S_i|\geq 2$ from the channel *Status (head|tail)*. Likewise, it can be decided whether $|S_{i+1}|=1$ or $|S_{i+1}|\geq 2$. The next step is to estimate the time slots that the protocol takes to finish its execution.

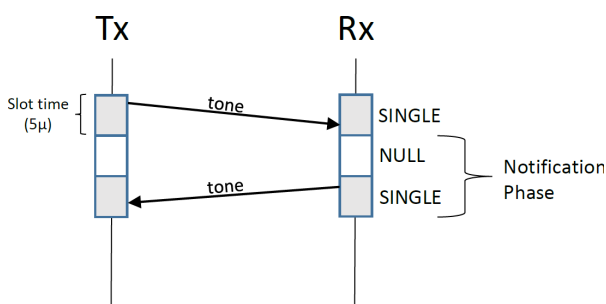


Fig. 4. Notification phase: After detected a SINGLE tone signal, all the Rx stations will wait one slot and they will transmit, in the second one, a tone signal

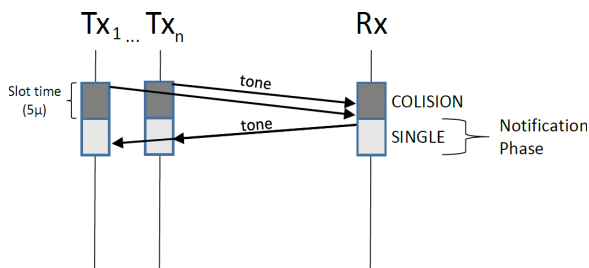


Fig. 5. Notification phase: After detected a COLLISION, all the Rx nodes will transmit, for one time slot, a tone signal

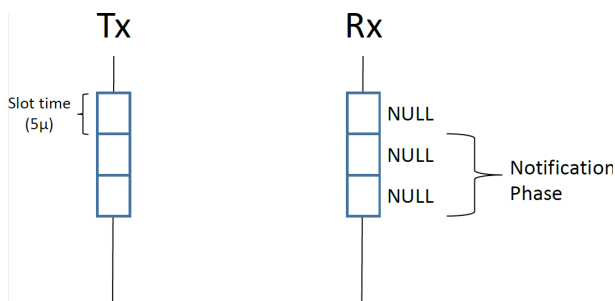


Fig. 6. Notification phase: After detected a NULL, all the Rx stations will wait for two consecutive time slots

Note that the CRP protocol has not been modified and it requires no transmission. In the while-loop of Routine 3, an iteration takes two time slots to complete the transmission phase. On the other hand, the notification phase is completed with 2 (two) or 3 (three) time slots. The notification phase takes two time slots when $|S_i|\geq 2$ and $|S_{i+1}|\geq 2$ in the while-loop of Routine 3. This is the case when COLLISION is detected in both sets as illustrated in Figure 5. Otherwise, the notification phase takes three time slots, which occurs when one of the set is empty, that is $|S_i|\geq 2$ and $|S_{i+1}|=\emptyset$, or vice-versa. Likewise, $|S_i|\geq 2$ and $|S_{i+1}|=1$ and vice-versa also takes three time slots. Hence, an iteration, of the while-loop, takes at most 5 (five) time slots: two for the transmission phase and three for the notification phase. Using the same reasoning of the previous section and by virtue of Lemma 1, the CRP protocol for NCD-scenario ends in $8n \times 5 = 40n$ time slots with probability at least $1-2^{-1.5n}$. Thereby, Lemma 2 is defined as follows:

Lemma 2

The CRP can Elect a Transmitting Node Among n Contending NCD-nodes, in a Single-Hop Ad Hoc Network, in at Most 40n Time Slots with Probability at Least $1-2^{-1.5n}$

An iteration of *GroupSplitting-NCD* algorithm can be completed in $25\mu s$, admitting that a *tone* signal can be transmitted in a $5\mu s$ slots. In this case, an iteration takes $10\mu s$ in the transmission phase and $15\mu s$ in the notification phase. In what follows, we show how the IEEE 802.11 standard can take advantage of the CRP as a replacement of the current BEB mechanism (802.11b Working Group, 1997).

IEEE 802.11 DCF Enhanced with CRP

This section presents a brief overview of the IEEE 802.11 Distributed Coordination Function (DCF) operation mode (802.11b Working Group, 1997) and shows how the DCF can be enhanced with the proposed CRP protocol. The IEEE 802.11 standard uses the listen-before-talk strategy. That is, a node wishing to communicate must first verify the channel status. In case the channel status is idle for a time equal to Distributed Inter-Frame Space (DIFS), the contending node selects a random backoff, which is decremented at each idle time slot. When the backoff reaches zero, the node transmits the desired packet. Collisions may occur when two or more stations transmit at the same time. To reduce the chance of collision, the Binary Exponential Backoff (BEB) is employed. The BEB requires that, on a collision event, the backoff window to be extended (802.11b Working Group, 1997).

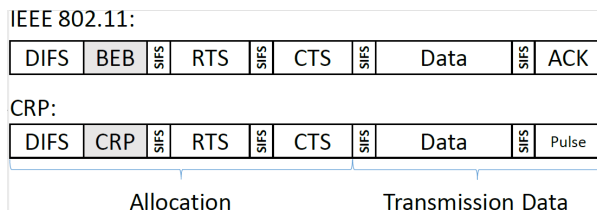


Fig. 7. IEEE 802.11 cycle: BEB Vs CRP mechanism

Figure 7 shows a successfully communication cycle of IEEE 802.11 protocol using both BEB and CRP. A successfully communication cycle comprehends the exchange of control frames, data frame and inter-frame spacing. This cycle can be viewed as two distinct parts: (i) contention phase and (ii) data transmission phase. The contention phase, when RTS/CTS frames are considered, involves the time necessary to complete the following sequence: DIFS + BEB + SIFS + RTS + SIFS + CTS. The data transmission phase involves SIFS + DATA + SIFS + Ack (802.11b Working Group, 1997). While some parts of the above cycle are fixed, such as DIFS and SIFS, others depend on the data rate used, number of contending nodes, etc. Of interest is the time taken by the BEB mechanism, which is influenced by the number of collisions and contending nodes. According to Fig. 7, one may observe that CRP can replace the BEB mechanism without further modifications to the DCF communication cycle. Once the nodes running the CRP have the capacity to issue in-band *pulse/ tone* signals, these signals can be employed to decrease network latency. Consequently, in what concerns to Data transfers, nodes running the CRP may employ *Data-pulse* sequence. It is important to state that *pulse* and *tone* signals can be differentiated either by phase, frequency, or amplitude (Liberti and Rappaport, 1999), allowing the nodes to deal properly with these signals. As it will be shown in the subsequent section, the CRP provides substantial gains as compared with the current BEB mechanism.

Performance Evaluation

This section presents the performance evaluation of the proposed CRP. To this end, the CRP has been implemented in the OMNeT++ 4.6 Simulator and compared it against the BEB mechanism (Binary Exponential Backoff) (802.11b Working Group, 1997). The next subsections detail the simulation environment and present the main simulation results.

Simulation Environment

The CRP has been implemented in the OMNeT++ 4.6 (OC, 2016a). For modeling the physical and data link layers, the MiXiM framework has been used (OC, 2016b).

Table 1. Values of parameters considered to achieve presented results

Description	Value
Data frame payload	1024 bytes
Propagation delay	1 μ s
Slot time	50 μ s
Channel bit rate	2 Mbps
DIFS	128 μ s
SIFS	10 μ s
RTS	272 μ s
CTS, ACK	248 μ s
Total Simulation time	15 s

The MiXiM framework focuses on physical and data link layers, providing a rich environment for the simulation of wireless networks. Table 1 presents the main simulation parameters used in this work. In the simulation, both CRP and BEB employ channel reservation frames, that is, Request-to-Send (RTS) and Clear-to-Send (CTS) frames. Data frames are only sent after a successful RTS-CTS exchange. Simulation results consider a varying number of contending stations. Particularly, the simulations considered $n = 8, 16, 32, 64, 128$ and 256 . Every contending node has a destination node that cannot belong to the set of contending nodes. That is, $n \leq |V|/2$. A single hop setting with static routes is considered. The simulation results were calculated as an average of 10 simulations, with a confidence level of 95%.

Channel Allocation Time

This subsection evaluates the number of channel reservations provided by the CRP for both CD and NCD scenarios. CRP behavior consists of decreasing the number n of contending nodes until a transmitter is selected. Therefore, the first evaluation is related to the time spent to properly elect a single station in a group of $n > 1$ stations. This evaluation considers the number of iterations, since the time will be proportional to it. Fig. 8 shows the average number of times slots for different number of contending nodes. The X-axis presents the number of contending stations while the Y-axis presents the time slots spent by CRP to choose a single transmitting station.

As previously referred, CRP needs at most $\log n$ successful iterations to properly choose a single transmitter. The simulation results confirm that CRP takes, on average, $\approx 2 \times \log_2 n$ times slots for CRP (CD) and $\approx 6 \times \log_2 n$ for CRP (NCD) to select a transmitter. These values are within the interval defined by Lemma 1, which is $16n$ times slots for CRP (CD) and $40n$ times slots for CRP (NCD) as defined by Lemma 2.

Next, we present an evaluation related to the number of channel reservations that can be done in a determined time. In this scenario, Data and Ack frames are assumed to take no time to be transferred, so that the nodes will

continuously run the contention mechanisms. RTS and CTS frames, however, are still considered as they are used to inform the remaining stations that a communication is taking place. This setting is closely related to that one in which many data channels are available, although the

contention resolution occurs in the control channel, similarly to the works in (Almotairi and Shen, 2010; Li *et al.*, 2003). In other words, it would be interesting to have an evaluation about the maximum number of stations per second that can be granted channel access.

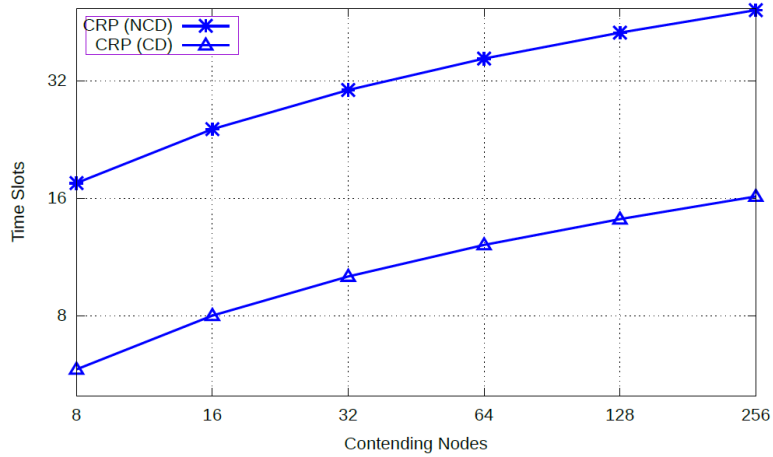


Fig. 8. The average number times slots per contention resolution

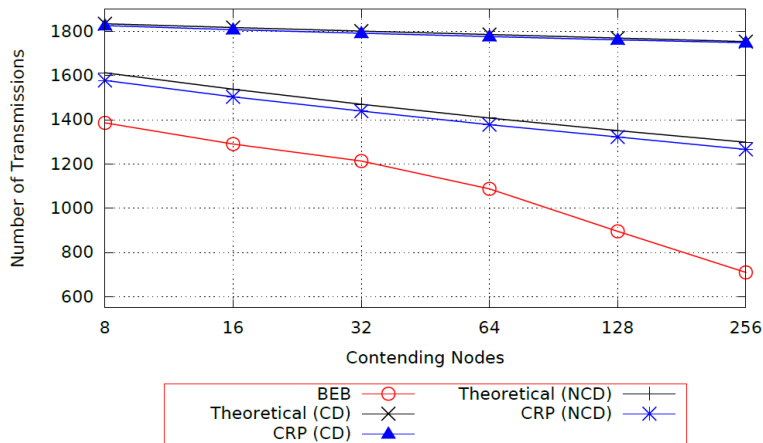


Fig. 9. Successful transmissions per second

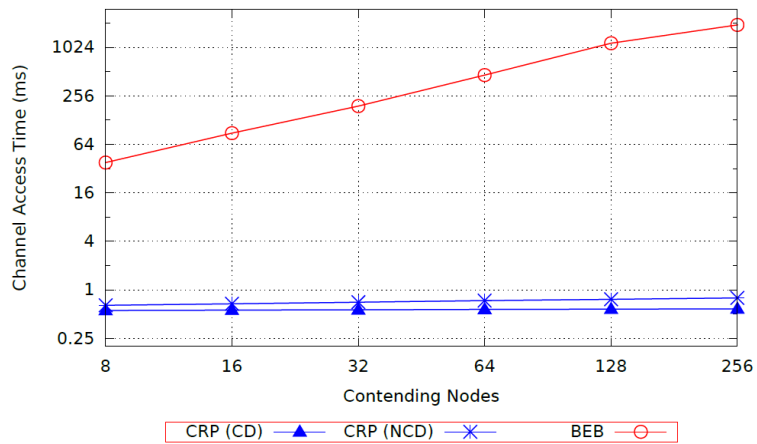


Fig. 10. Medium access time for an increasing quantity of contending stations

Figure 9 shows the successful transmissions per second for an increasing number of stations. Note that BEB decays fast when the number of contending stations is high. In turn, CRP reaches a significant increase in the overall transmissions per second, when compared to BEB. As mentioned in Section 3.2, compared with a CD-station, a NCD-station uses 4 additional time slots per iteration in the splitting process. Consequently, CD-stations attain better performance. The maximum theoretical number of transmissions has its calculation detailed in (Jun *et al.*, 2003). So, our work uses this approach to calculate the maximum theoretical number of transmissions for the CRP protocol (Theoretical (NCD) in the Fig. 9). As can be seen, the CRP is close to its theoretical limit. Compared to BEB, with $n = 256$, CRP (CD) and (NCD) can improve the number of successful transmission up to $\approx 246\%$ and $\approx 178\%$, respectively.

Figure 10 presents the average time spent for each protocol to grant channel access to a varying number of contending stations. CRP protocol grants channel access in a portion of the time spent by BEB. This occurs because CRP decreases the contending nodes by half in each iteration. Indeed, the time spent by BEB is exponentially related to the number of contending nodes. In turn, the time spent by CRP has a logarithmic relation to the stations contending for channel access.

Next, we evaluate the time taken by an individual station to be granted channel access. Figure 11 shows the simulation results for $n = 256$. Also, Fig. 11 presents the trend using linear regression and the respective best-fit equations. From this figure, it is important to note that CRP implementation grants channel access in a portion of the time spent by BEB. Clearly, with $n = 25$, BEB cannot meet the constraints to operate with real-time applications due to its high latency (Filho *et al.*, 2012). In fact, some applications such as Voice over IP (VoIP) require round-trip time to be at most 300 ms (Zhai *et al.*, 2006). However, for $n = 256$, BEB is not able to meet

this requirement. As an alternative, CRP can provide channel access in a time shorter than 1ms. Therefore, CRP can take advantage of its low latency to handle applications that have severe QoS constrains, such as online gaming (Chen *et al.*, 2006).

Evaluation of the Communication Process

This subsection evaluates the performance of the CRP protocol when considering Data frames with a significant payload size (1024 bytes). As previously mentioned, after a successful RTS-CTS exchange, CRP employs Data-tone sequence instead Data-Ack sequence, which is used in BEB.

Figure 12 presents the number of transmissions per second for an increasing number of contending stations. Compared against the BEB mechanism, with $n = 256$, CRP (CD) delivers nearly twice the number of transmissions. CRP (NCD) delivers ≈ 1.93 times more transmission than the BEB. The average channel access time, presented in Fig. 13, exhibits a similar behavior of that observed in Fig. 10. However, with a payload of 1024 bytes, the performance of the BEB deteriorates faster. On average, each contending node waits at least 1.9 sec to obtain channel access for $n = 256$. On the other hand, CRP protocol grants channel access in less than 1ms. That is, CRP allows channel access in 0.03% of the time spent by BEB. The performance difference is due to the control frame collisions, as can be observed in Fig. 14. The figure shows the RTS frame collisions experienced by the BEB mechanism. This setting considers a payload of 1024 bytes (denoted as BEB (1024)) without considering the Data-Ack sequence, denoted simply as BEB. Note that the latter uses the same approach as defined in Subsection 4.2. As can be seen, when the number of contending nodes is 256, approximately 654 collisions per second occur when BEB scheme is considered (1024 bytes). That is, for each contending station there are 2.55 collisions on average.

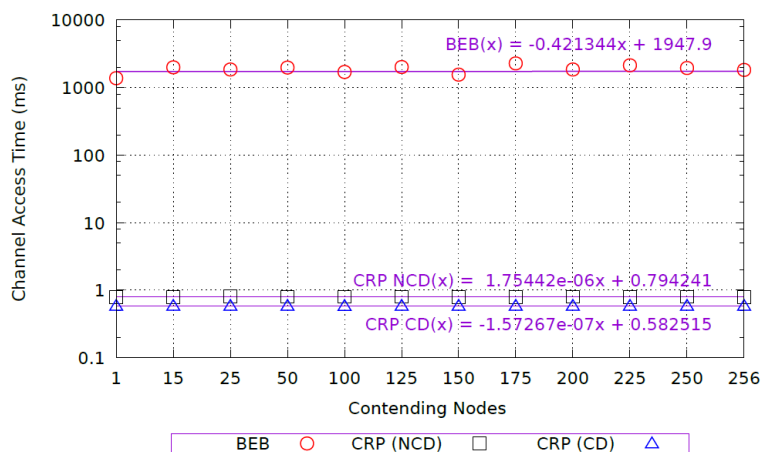


Fig. 11. The average channel access time per contending node with $n = 256$

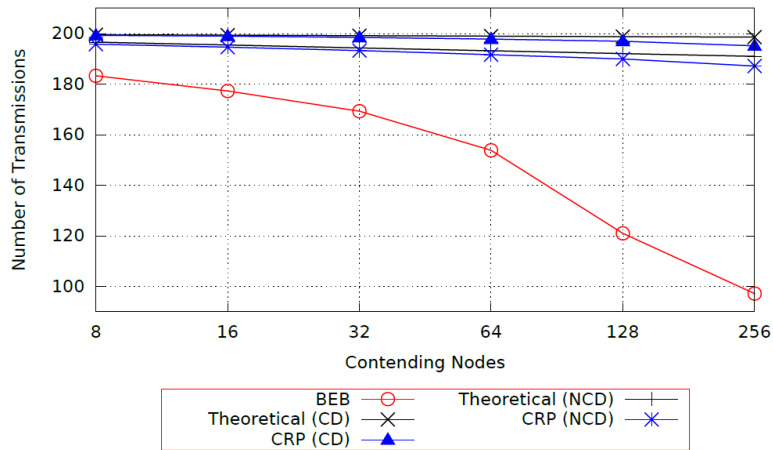


Fig. 12. Successful transmissions per second with a payload of 1024 bytes

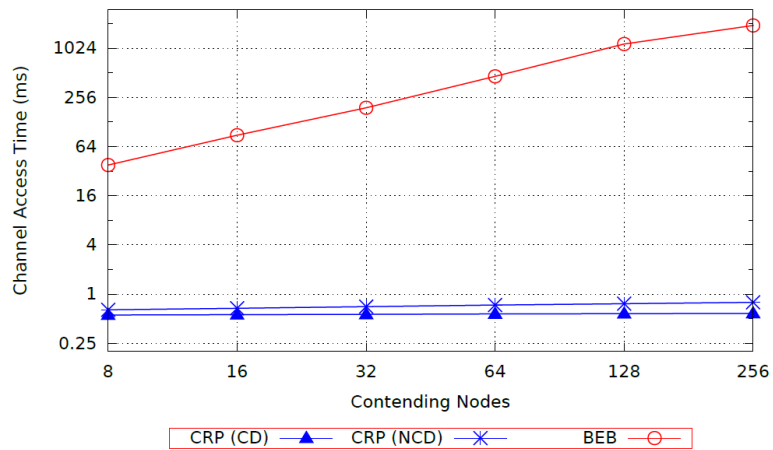


Fig. 13. Average channel access time with a payload of 1024 bytes

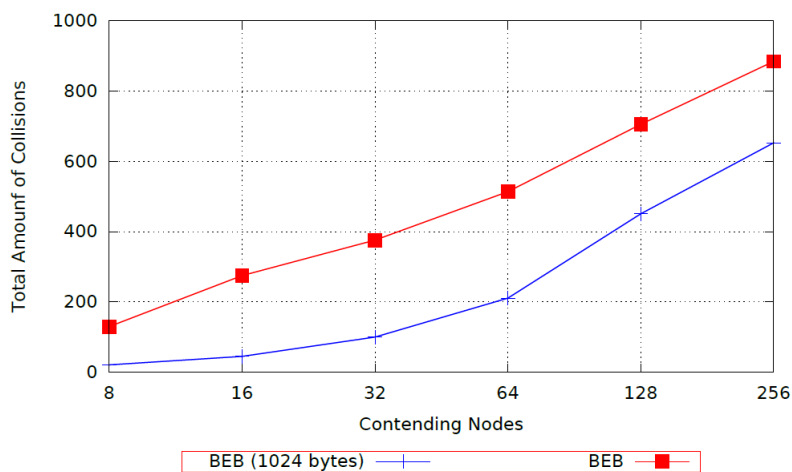


Fig. 14. BEB collisions with and without Data-Ack frames

As exposed in this section, CRP can outperform BEB when assessed under several metrics. So, CRP usage

implies a positive impact on ad hoc networks. Concluding remarks are presented next.

Conclusion

This paper proposed a channel reservation technique that takes advantage of pulse and tone signals to obtain a collision-free channel access. Our technique, termed Collision Resolution Protocol (CRP), recursively decreases the number of contending stations. Using this process, CRP provides a fast channel access. Furthermore, CRP works with stations empowered with and without Collision Detection (CD) capabilities.

This paper proves that the task for choosing a single transmitter from a subset of $n \geq 2$ contending stations can be performed in $16n$ and $40n$ time slots, for the CD and Non-CD stations, respectively, with probability at least $1 - 2^{-1.5n}$. Empirical evaluation on the OMNeT++ simulator has been used to assess the goodness of the proposed CRP protocol. The traditional Binary Exponential Backoff (BEB), defined in the IEEE 802.11 standard has been selected for comparison. The simulation results show that CRP grants channel access using 0.03% of the time spent by the BEB mechanism. Moreover, the results show that CRP can deliver 100% more transmissions per second than BEB. The results show that CRP can grant channel access in a fraction of time when compared with the traditional BEB while providing a fair resource allocation. These features are highly desirable when severe QoS constraints are considered. Hence, we plan to explore the use of CRP technique to improve the resources allocation management in a multichannel environment as well as to consider mobility and multi-hop settings.

Funding Information

Our work was partially supported by research grants from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and DPP/UnB. CAPES–Proc. BEX 11859/13-0.

Author's Contributions

All authors equally contributed in this work.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other author have read and approved the manuscript and no ethical issues involved.

References

802.11b Working Group, 1997. Part 11: Wireless LAN Medium Access Control (MAC) and physical layer (phy) specifications.

- Almotairi, K. and X. Shen, 2010. MMAC-HR: Multi-channel medium access control with hopping reservation for multi-hop wireless networks. Proceedings of the Global Telecommunications Conference, (GTC' 10), IEEE Xplore Press, pp: 1-5. DOI: 10.1109/GLOCOM.2010.5684198
- Bharghavan, V., A. Demers, S. Shenker and L. Zhang, 1994. Macaw: A media access protocol for wireless lan's. SIGCOMM Comput. Commun. Rev., 24: 212-225. DOI: 10.1145/190809.190334
- Bianchi, G., 2000. Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J. Selected Areas Commun., 18: 535-547. DOI: 10.1109/49.840210
- Chen, K.T., P. Huang, G.S. Wang, C.Y. Huang and C.L. Lei, 2006. On the sensitivity of online game playing time to network QoS. Proceedings of the 25th IEEE International Conference on Computer Communications, Apr. 23-29, Barcelona, Catalunya, Spain, pp: 1-12. DOI: 10.1109/INFOCOM.2006.286
- Deng, J. and Z.J. Haas, 1998. Dual Busy Tone Multiple Access (DBTMA): A new medium access control for packet radio networks. Proceedings of the International Conference on Universal Personal Communications, Oct. 5-9, IEEE Xplore Press, pp: 973-977. DOI: 10.1109/ICUPC.1998.733652
- Felice, M.D., L. Bedogni and L. Bononi, 2013. Group communication on highways: An evaluation study of geocast protocols and applications. Ad Hoc Netw., 11: 818-832. DOI: 10.1016/j.adhoc.2012.09.011
- Filho, P.H., M.F. Caetano and J.L. Bordim, 2012. A packet aggregation mechanism for real time applications over wireless networks. Int. J. Network. Comput., 2: 18-40.
- Guimarães, L.M., J.L. Bordim and K. Nakano, 2015. Using Pulse/Tone Signals as an Alternative to Boost Channel Reservation on Directional Communications. IEICE Trans. Fundamentals Electron. Commun. Comput. Sci., E98-A: 1647-1656. DOI: 10.1587/transfun.E98.A.1647
- Haas, Z. and J. Deng, 2002. Dual Busy Tone Multiple Access (DBTMA)-a multiple access control scheme for ad hoc networks. IEEE Trans. Commun., 50: 975-985. DOI: 10.1109/TCOMM.2002.1010617
- Jun, J., P. Peddabachagari and M. Sichitiu, 2003. Theoretical maximum throughput of IEEE 802.11 and its applications. Proceedings of the 2nd IEEE International Symposium on Network Computing and Applications, Apr. 18-18, IEEE Xplore Press, pp: 249-256. DOI: 10.1109/NCA.2003.1201163
- Kleinrock, L. and F.A. Tobagi, 1975. Packet switching in radio channels: Part i--carrier sense multiple-access modes and their throughput-delay characteristics. IEEE Trans. Commun., 23: 1400-1416. DOI: 10.1109/TCOM.1975.1092768

- Kumar, S., V.S. Raghavan and J. Deng, 2006. Medium access control protocols for ad hoc wireless networks: A survey. *Ad Hoc Netw.*, 4: 326-358. DOI: 10.1016/j.adhoc.2004.10.001
- Li, J., Z.J. Haas, M. Sheng and Y. Chen, 2003. Performance evaluation of modified IEEE 802.11 MAC for multi-channel multi-hop ad hoc network. *J. Interconnect. Netw.*, 04: 345-359. DOI: 10.1109/AINA.2003.1192896
- Liberti, J.C. and T.S. Rappaport, 1999. *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications*. 1st Edn., Prentice Hall PTR, Upper Saddle River, ISBN-10: 0137192878, pp: 376.
- Lu, F., G.M. Voelker and A.C. Snoeren, 2015. Managing contention with medley. *IEEE Trans. Mob. Comput.*, 14: 579-591. DOI: 10.1109/TMC.2014.2315822
- Ma, J., H. Sekiya, A. Nagasaki, N. Komuro and Sakata, 2012. MAC protocol for ad hoc networks using smart antennas for mitigating hidden and deafness problems. *IEICE Trans. Commun. E95-B*, 11: 3545-3555. DOI: 10.1587/transcom.E95.B.3545
- Mathys, P. and P. Flajolet, 2006. Q-ary collision resolution algorithms in random-access systems with free or blocked channel access. *IEEE Trans. Inf. Theor.*, 31: 217-243. DOI: 10.1109/TIT.1985.1057013
- Motwani, R. and P. Raghavan, 1995. *Randomized Algorithms*. 1st Edn., Cambridge University Press, New York, ISBN-10: 0521474655, pp: 476.
- Nakano, K. and S. Olariu, 2002. A survey on leader election protocols for radio networks. *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks*, May 22-24, IEEE Xplore Press, pp: 63-68. DOI: 10.1109/ISPAN.2002.1004263
- OC, 2016a. Omnet simulator. Omnet Community.
- OC, 2016b. MiXiM Project. Omnet Community.
- Perkins, C.E., 2008. *Ad Hoc Networking*. 1st Edn., Pearson Education India, ISBN-10: 8131720969, pp: 384.
- Shih, K.P., W.H. Liao, H.C. Chen and C.M. Chou, 2009. On avoiding RTS collisions for IEEE 802.11-based wireless ad hoc networks. *Comput. Commun.*, 32: 69-77. DOI: 10.1016/j.comcom.2008.09.024
- Song, N.O., B.J. Kwak, J. Song and M. Miller, 2003. Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm. *Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference*, Apr. 22-25, IEEE Xplore Press, pp: 2775-2778. DOI: 10.1109/VETECS.2003.1208898
- Tan, K., J. Zhang, H. Wu, F. Ji and H. Liu *et al.*, 2009. Sora: High performance software radio using general purpose multi-core processors. *Proceedings of the 6th ACM/USENIX Symposium on Networked Systems Design and Implementation*, (SDI' 09), pp: 75-90.
- Tseng, Y.C., S.Y. Ni, Y.S. Chen and J.P. Sheu, 2002. The broadcast storm problem in a mobile ad hoc network. *Wireless Netw.*, 8: 153-167.
- Venkitaraman, N., T.E. Kim and K.W. Lee, 1999. Design and evaluation of congestion control algorithms in the future internet. *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, May 01-04, Atlanta, GA, USA, pp: 212-213. DOI: 10.1145/301453.301575
- Wattanamongkhol, N., 2013. Performance comparison of collision resolution algorithms with amount of feedback information. *Proceedings of the International Symposium on Wireless and Pervasive Computing*, Nov. 20-22, IEEE Xplore Press, pp: 1-4. DOI: 10.1109/ISWPC.2013.6707439
- Zhai, H., J. Wang and Y. Fang, 2006. Providing statistical QoS guarantee for voice over IP in the IEEE 802.11 wireless LANs. *IEEE Wireless Commun.*, 13: 36-43. DOI: 10.1109/MWC.2006.1593523
- Zhou, D. and T.H. Lai, 2005. A compatible and scalable clock synchronization protocol in IEEE 802.11 ad hoc networks. *Proceedings of the International Conference on Parallel Processing*, (ICPP' 05), ACM, pp: 295-302. DOI: 10.1109/ICPP.2005.6