A Collision Resolution Algorithm with QoS Guarantees for Ad-hoc Wireless LANs

David V. Cortizo*, Jorge García**
**Computer Architecture Department, Polytechnic University of Catalonia
Gran Capitá s/n, Mòdul D6, 08071, Barcelona, Spain
phone: +34.3.401.6798, fax: +34.3.401.7055, e-mail:jorge@ac.upc.es

Abstract

In this paper we propose and analyze a Medium Access Control protocol for high-speed ad-hoc WLAN. The protocol is a Collision Resolution Algorithm that groups the packets arrived during fixed-length time intervals and supports two distinct traffic classes, one receiving bounded packet delay guarantees. Packets belonging to the same set will be transmitted via a CSMA algorithm whereas the different sets will be served following a FIFO discipline. The results obtained by means of simulation and analytical models for Poisson traffic show that the protocol can support real-time services even when the network is heavily loaded, being the non Real-time traffic throughput the difference between the maximum throughput (0.88 with Poisson traffic) and the Real-time traffic throughput.

1 Introduction

The deployment of high speed wireless LANs has become a topic of increasing interest over the past few years. In this paper we propose and analyze a Medium Access Control (MAC) protocol for high-speed ad-hoc Wireless Local Area Networks (WLAN) that guarantees a maximum delay for real-time applications even when the network is heavily loaded. In wireless networks we need to solve some problems [Acampora 1996] which are not present or are of marginal importance in wired networks:

- **Wireless networks use either radio or infrared channels.** In either case the particular characteristics of wireless channels (multipath, interference, doppler spread, etc) bound the maximum achievable transmission rate and BER to worse figures than the ones that can be obtained with for instance optical fiber or twisted-pair copper channels [Falsafi et al. 1996].

- **Wireless networks are inherently shared medium networks.** Consequently, a Medium Access Control (MAC) mechanism is needed to avoid collisions due to simultaneous packet transmissions in the same frequency band. Unfortunately, it is much harder to get good delay-throughput characteristics using a MAC protocol than using centralized multiplexing and switching.

---

1This work was supported by projects XUGA 10503A96 and TIC95-0982-C02-01. Fundacion Caixa Galicia has supported D.V. Cortizo's stay at HUT, where part of the work was carried out.
In wireless networks it is important to minimize the drain of portable batteries and consequently power saving is a major constraint in the design of a WLAN.

The establishment of a connection with a wireless user is also problematic as the network will first have to locate that user. This implies the use of paging if the called user is known to be somewhere within a small cluster of cells or the use of a database which stores the user’s current location. Once the position of the user is known it must be continually tracked as the user roams among different cells. This roaming is accomplished by means of connection rerouting, also known as cell handoff, which is a crucial system design issue when QoS guarantees are to be supported.

Wireless LANs usually have two types of realizations: ad-hoc WLANs and infrastructured WLANs. In ad-hoc WLANs, mobile terminals may establish peer-to-peer communication among themselves in a small area without the help of any infrastructure such as a wired/wireless backbone. In infrastructured WLANs communication among mobile devices is supported via a communication infrastructure, typically a higher-speed wired or wireless network. The wireless medium contact points with the backbone are called access points. These access points can be either base stations or repeaters for enlarging the coverage area of communication.

In infrastructured WLANs Time Division Multiple Access (TDMA) access control protocols are often preferred [Bantz et al. 1994], [Chen 1994]. With TDMA-based protocols it is necessary to define a scheduling function, often located at the base stations, responsible for assigning the different time slots to the mobile nodes of the network. In ad-hoc WLANs, where every station has a similar functionality, Carrier Sense Medium Access (CSMA) based protocols may be advantageous because of their simplicity and because no station is compelled to assume special functions as in TDMA-based protocols. In this paper we investigate a modification of the basic CSMA protocol in order to get a suitable MAC protocol for ad-hoc WLANs.

The CSMA protocol assumes that nodes can sense whether other nodes are transmitting after a very small (compared to the packet transmission time) propagation and detection delay. The basic CSMA mechanism is well known and has been used in many multiaccess systems: before transmitting a packet, a node senses the channel. If the channel is sensed idle (i.e. no other packet is currently being transmitted) the packet is transmitted. Otherwise, if the channel is sensed busy, the packet transmission is deferred. Packet collisions may occur, however, when transmissions of packets from different nodes are started within a time window of length equal to twice the one-way propagation delay between the nodes. When two or more packets collide, they must be retransmitted again.

There are several characteristics in wireless networks that make the implementation of CSMA protocols harder than in the case of wired networks, arising problems such as the hidden terminal node, imperfect carrier sensing and, in case of cellular networks, spatial domain factors [Chen 1994]. In this paper we assume that these issues have been sorted out and their only effects are occasional errors in the reception of packets or signalling
information.

Moreover, if we want to integrate delay-sensitive and delay-insensitive traffic in the same WLAN we must keep end-to-end delay and delay variations below certain maximum values, at least for delay-sensitive traffic. Unfortunately, it is well known that CSMA protocols usually suffer large access delay quantiles (actually the basic CSMA algorithm is instable) due to the existence of packet collisions, which imply further retransmission of the colliding packets. *Indeed packets in different nodes do not follow a First In First Out (FIFO) discipline.* We have instead something similar to a random queueing discipline, which in turn originates much worse delay quantiles than those of FIFO queues.

Modifications to CSMA basic protocol, like the tree-type CSMA algorithms, have been proposed and analyzed [Jacquet 1993], [Panwar et al. 1993], [Polyzos et al. 1987], [Polyzos et al. 1993] in order to solve the instabilities that arise in the basic CSMA mechanisms and to improve the throughput-delay characteristic. These protocols introduce, on the one hand, a certain amount of overhead but, on the other, move the performance from that of CSMA closer to the FIFO discipline throughput-delay behavior. Typically there will be a trade-off between resemblance to FIFO discipline and complexity and overhead introduced by the protocol.

The basic MAC protocol that we present in this paper is a collision resolution algorithm [Georgiadis et al. 1987], [Huang et al. 1986] which aims to approximate the delay characteristics to those of an ideal multiplexer with a FIFO discipline. The protocol groups the arriving\(^2\) packets into sets which contain the packets generated during the same fixed-length time interval. Packets belonging to the same set will be transmitted using a CSMA algorithm and the different sets will be served following a FIFO discipline. The shorter the grouping time, the closer to FIFO performance, as (in average) there will be less packets per set. Unfortunately, the overhead introduced by the algorithm will also increase as we shorten the length of the grouping intervals.

The basic access mechanism will then be modified in order to support two different traffic classes. One of them is assumed to carry delay sensitive data which will be referred to as Real-time traffic. This traffic will be served without losses and meeting its delay constraints as long as the offered Real-time load does not exceed the basic mechanism maximum throughput (around 0.88). The second traffic class is meant for data without temporal constraints and it will make use of whatever capacity is left by the Real-time traffic. Therefore the non Real-time throughput will be the difference between the maximum basic mechanism throughput and the Real-time offered traffic.

In the remainder of this section we present our assumptions on the system. In section II we describe the MAC protocol: how it groups the packets pending for transmission in sets called Transmission Sets and the mechanism by which these sets are served (i.e.

\(^2\)A packet arrival is defined to occur when a new packet is generated at any user site.
their included packets are successfully transmitted), how it could react against errors in reception, the system initialization mechanism, a sleep mode to minimize the drain of portable batteries and the extension for supporting the two different service classes. In section III we describe both the simulation model and the analytical model developed to study the performance of the protocol, along with their limitations. These models obtain the distribution function of the delay suffered by the packets from their generation until the completion of their successful transmission. The results obtained with the models are shown and analyzed in section IV.

1.1 Assumptions and Description of the System

In the design of the MAC protocol we have made the following assumptions:

- The network is supposed to be an ad-hoc high-speed wireless local area network. The possibility of handover has not been addressed.
- Either full connectivity among the stations or the existence of a higher level protocol to implement forwarding among nodes.
- Use of either radio or infrared channel providing a bit rate as seen by the MAC sublayer of 25 Mbps (or higher).
- Every successfully transmitted packet is positively acknowledged by the receiver via a short ACK packet. Multicast traffic has not been considered.
- We assume a constant packet length of 1000 bits. This includes payload, overhead and the time needed to generate, transmit and process an ACK. Therefore all users in the system transmit packets of constant duration $T_s \approx 40 \mu s$.
- The range of the network is supposed to be $R \approx 30 m$, so the maximum one-way propagation delay between any pair of users is $a \approx 0.1 \mu s$.
- We choose $T_s$ as the time slot length, and we further define mini-slots with length $t_s = 6a \approx 0.6 \mu s$. This way $T_s = Nt_s$, where $N \approx 66$.
- The physical level will identify the different states of the channel: idle (no carrier), pilot tone transmission and packet transmission.

2 Description of the Basic Access Mechanism

The basic medium access mechanism is CSMA with random backoff (an integer number of minislots) and adaptive backoff window size. However, this access mechanism is not applied to the whole set of stations having packets to transmit but only to a subset of them. The CSMA application to just some of the arrived packets pending for transmission, those that were first generated, will allow us to approximate the protocol performance to the behavior of the FIFO discipline. Let us recall here the meaning of packet arrival throughout the paper: a packet arrival takes place whenever a packet is generated for transmission at any user site.
The protocol will generate a collection of (possibly empty) Transmission Sets (TS) \( \{S_k\} \), each of them containing the packets arrived during successive disjoint intervals of constant length \( W_t \). These Transmission Sets are served (i.e. the associated packets are transmitted using CSMA) with a FIFO discipline, so that \( S_k \) is served before \( S_j \) if \( k < j \). This way, if every \( S_k \) had no more than one packet to be transmitted we would be facing a FIFO-MAC algorithm, with packets being sent in order of arrival.

The average number of packets per TS is closely related to the integer \( W \), which determines the length of the grouping interval where the arriving packets are collected into the same TS. The smaller \( W \) the more similar the behavior of the protocol will be to that of a FIFO system. On the contrary, the protocol will perform as pure CSMA if \( W \) is made big enough. The value of \( W \) will be a trade-off between the desired properties of a FIFO system and the overhead introduced by the protocol.

Every active station keeps track of two sequence numbers \( SN \) associated to the TSs:

- The sequence number of the TS that is being served, \( N_s(t) \).
- The sequence number of the TS that any packet arrival at time \( t \) must join, \( N_g(t) \).

This way the packets can be associated to two different sequence numbers:

- The sequence number of the TS the packet belongs to \( N_g(t_a) \), being \( t_a \) the packet arrival time.
- The sequence number of the TS that is being generated at \( t_{tx} \), when the packet begins to be transmitted \( N_g(t_{tx}) \).

The pair \( (N_g(t_a), N_g(t_{tx})) \) is transmitted along with the data payload as control information and it is part of the overhead introduced by the protocol.

In the next subsections we will address the questions of how to generate, process and eliminate the transmission sets; in order to solve them, the mobile stations will emit pilot tones of length one minislot. The protocol overhead also includes the minislots spent in the transmission of these pilot tones.

### 2.1 Generation of the Transmission Sets

We will assume in this section that the network has been already set up, as both the network initialization mechanism and the procedure followed by a terminal trying to hook on an established network are issues that will be addressed in the following sections.

Let us suppose that no packets have been generated in the system for a long period of time. The system will be then in an idle state, characterized by all terminals in the network having \( N_s(t) = N_g(t) \). When, in this situation, a packet is generated at any mobile node, it is immediately transmitted.
Once the first transmission attempt takes place a busy period of the system starts. During busy periods a new transmission set (either empty or not) is generated every $W$ minislots, and any packet arrival will defer transmission until the TS it belongs to begins to be served: a packet arriving at $t_1$ will be tagged as belonging to the TS number $N_g(t_1)$, and the station will not try to transmit it until a time $t_2$ such that $N_s(t_2) = N_g(t_1)$. The service of a non empty TS consists of the successful transmission of all its packets. The procedure employed by the system to serve the TSs, along with the mechanisms needed to delimit the start and end of service of a non empty TS and to identify the idle TSs, will be described in section 2.2.

If the network is stable the system will eventually get empty again, when $N_s(t) = N_g(t)$. At this point the generation of TSs is halted and the system enters an idle period. A TS will be empty if there has not been any packet arrival during its associated interval of length $W$, but it is important to note that only during the busy periods of the system empty TSs may be generated.

Packets which find at arrival time the system in idle state will directly attempt transmission, starting a new busy period, the generation of new TSs, and reinitiating the process described above. We can therefore conclude saying that Transmission Sets, empty or not, are periodically generated during the busy periods whereas no TSs at all are generated during the idle periods of the system.

### 2.2 Service of the Transmission Sets

The service of a Transmission Set consists of the successful transmission of all the packets arrived during the grouping interval of length $W$ associated to the TS. Nevertheless, it is also possible that the TS is an empty one, and the MAC protocol will have to handle these two possible situations. We need a mechanism allowing the active stations in the network to identify the beginning and end of service of a non empty TS, as well as to identify empty TSs.

At the beginning of a TS service all the stations with packets belonging to that TS will emit a pilot tone during one minislot. This mechanism allows the stations to identify empty TSs (if there is no pilot tone present in the channel), discard them and proceed after only one minislot delay with the processing of the next TS.

If the TS is non-empty the stations, after emitting a pilot tone that identifies the beginning of the TS service, wait a random number of minislots (between zero and the current random backoff window value) before sensing the channel and starting transmission if sensed idle (CSMA with random backoff window). At the end of the transmission there will have been either a collision or a successful transmission, and the random backoff window value will be adapted accordingly.
After the successful transmission of a packet we may have two different scenarios:

- There are still more packets in the TS that have not been transmitted (i.e. we have not finished the service of the TS yet), and the stations involved should continue executing the random-backoff CSMA algorithm.

- There are no more packets to transmit associated to the currently served TS (i.e. we have already finished the service of the TS), and the MAC should start processing the next TS.

These different situations can be easily identified by the active stations if we force all the stations still having packets to transmit in the currently served TS to emit a pilot tone after every successful transmission. In the former case the stations waiting for the TS service to end will sense a pilot tone and keep on waiting. In the latter case, the waiting stations will sense an absence of carrier during one minislot, realize that the service of the \(i\)-TS has finished and start processing the \((i + 1)\)-TS. The end of service of a TS is therefore identified by the absence of carrier after a successful packet transmission.

In figure 1 we have tried to illustrate the temporal evolution of the protocol, assuming three Mobile Stations in the system St1, St2 and St3. When the system has been idle for a long enough period the first packet arrived at any station (St1 in the figure) is immediately transmitted and it also starts the generation of TSs (TS1 in this case) with consecutive sequence numbers. After the transmission of the message (St1Tx) St2 and St3 emit a pilot tone, as they have packets to transmit belonging to TS1, and generate a random backoff. In the example we suppose that there is no collision and that St3 gains access to the medium. After transmission of St3Tx, St2 emits a pilot tone, generate a random backoff and transmits its packet. As no mobile station has any other packet to transmit belonging to TS1 the next minislot remains idle, and so the service of the next TS (TS2) starts. However, as there has not been any packet arrival during the interval associated to TS2, there will be another idle minislot which will indicate that TS2 is empty. In the next minislot the service of TS3 starts with a pilot tone generation by the stations having packets belonging to TS3 (St1 and St2), a random backoff follows and then either a collision or a successful transmission will occur. In case of collision, at the end of the failed transmission all the stations with pending packets will transmit a pilot tone, adapt the size of their
random backoff window and generate a random backoff before reattempting transmission.

3 System Initialization and Activation of Stations in Sleep Mode

We address here the problem of hooking up to an ad-hoc WLAN already at work supposing that band frequency determination, bit synchronization and packet delineation are questions already solved by the physical layer.

When a terminal wishes to transmit but has to previously synchronize with the network (either because it had entered before the power-saving mode or because it is entering the network for the first time) it starts tracking the channel activity. If the medium is sensed idle (no carrier is detected) during $N_{idle}$ minislots the mobile will assume the system is idle, and directly attempt transmission. Otherwise (carrier detected), the terminal will keep on listening to the medium until a correct packet reception takes place. This first detected packet will provide the mobile with the pair $\left( N_g(t_a), N_g(t_{tx}) \right)$, which will allow the synchronization of both the TS's generation and service processes with the network. A good choice for this number $N_{idle}$ is the maximum window size of the random backoff algorithm.

This synchronization mechanism introduces a maximum misalignment of $(W - 1)$ minislots, which we consider perfectly acceptable, taking into account that successive transmissions will help to diminish the synchronization error.

The process described above may be implemented both by a station first entering the network and by terminals that have been in power-saving mode and now wish to transmit. A station in power-saving mode will also have to periodically wake up and listen to the channel to determine whether there is any station trying to send it a packet. Again, we will assume the existence of a higher level protocol used by the mobile entering the sleep mode to inform the rest of the terminals that it will only listen to the channel at certain periodically elapsed points in time.

One problem concerning both the idle periods and empty TSs service is that a station tracking the channel activity might mistake the service of a sequence of $N_{idle}$ (or more) empty TSs for an idle period. Besides, these long sequences of empty TSs inside a busy period may cause the loss of synchronization among the mobile terminals. A possible solution could be to force the active stations to broadcast packets with the control information $(N_s, N_g)$ whenever there is a long sequence of empty TSs inside a busy period.
3.1 Response in Presence of Errors

The analysis of errors taking place in minislots associated to the transmission of data packets will be obviated in this section, as these errors result in a negative acknowledge emitted by the receiver, and therefore will be treated in the same way as packet collisions. However, the protocol operation is based on the transmission and detection of pilot tones, which determine the start and end of service of every Transmission Set along with the coordination among the stations (also obtained by the exchange of the values $N_g(t_a)$ and $N_g(t_{tx})$). In this section we will describe the mechanisms employed to overcome the misdetection of the pilot tones and possible desynchronization among stations.

There are two mechanisms adopted to cope with possible erroneous detection of pilot tones and loss of synchronism among stations:

- A *time-out* establishing that if no carrier is detected for a certain number of minislots $N_{idle}$ the system will be assumed to be idle and pending transmissions will be directly attempted. This procedure has already been presented in previous sections, and it will solve deadlock problems that otherwise might arise, as we describe at the end of this section.

- Every station keeps the parameters $N_s(t)$ and $N_g(t)$ as well as the values $N_g(t_a)$ for every packet pending for transmission. If the terminal receives a packet with a field $N_g(t_{tx})$ different than its own $N_s(t)$ (no matter if it is higher or lower), it will adopt this new value first updating $N_s(t)$ and second adjusting the parameters $N_g(t)$ and $N_g(t_a)$ associated to the packets waiting in the transmission buffer, so that their difference with the old value of $N_s(t)$ is maintained with the updated $N_s(t)$.

In the remainder of the section we will analyze some typical scenarios that may arise as well as the operation of the protocol in presence of misdetection of the pilot tone.

If there is a pilot tone emitted by one or more terminals indicating either beginning or continuation of service of the TS $S_k$ but there is one node that senses as idle the corresponding minislot, that station will assume that the $S_k$ service has finished and proceed to serve $S_{k+1}$. We distinguish two situations:

- The first successfully transmitted packet since the misdetection instant belongs to the station that has erroneously detected the pilot tone. This will force the rest of the terminals to adjust their values $N_s(t)$, $N_g(t)$ and $N_g(t_a)$, as it has been described above. In this case although the FIFO service of the TSs is corrupted by at least one packet transmitted by the misaligned terminal, the effects on the overall operation of the system are negligible as the FIFO discipline among packets belonging to the same station remains unaffected.

- The first successfully transmitted packet since the misdetection corresponds to some other station. In this case the terminal that detected erroneously the pilot tone will readjust its parameters $N_s(t)$, $N_g(t)$ and $N_g(t_a)$. 


On the other hand, the opposite situation, where a station detects the presence of a pilot tone while the corresponding minislot has remained idle, is more dangerous as it might lead to a dead-lock situation: if there is not any packet transmission from the other terminals, bringing about a resynchronization of the misaligned node, the terminal will wait for the end of service of a TS which has already been completely served. Therefore, we distinguish again two different situations:

- The misaligned station detects activity in the channel (i.e. pilot tones, collision or successful transmission). In this case the terminal will wait for the first correctly transmitted packet to then adjust the values of $N_s(t)$, $N_g(t)$ and $N_g(t_a)$.
- There is no activity in the channel proceeding from other terminals. In this case, if the misaligned station senses the channel idle for an interval equal to the maximum window size of the backoff algorithm, it will immediately attempt transmission, reset its parameters $N_s(t)$ and $N_g(t)$ and appropriately modify the values of $N_g(t_a)$.

4 Quality of Service Support

The MAC protocol described in the preceding section sets up the primary channel access mechanism. In this section we describe a mechanism that allows the support of two different traffic classes, one of data without temporal constraints and another one associated to real-time traffic.

The protocol overhead conveys information about the network congestion level, as every transmitted packet includes the pair $(N_g(t_a), N_g(t_{tx}))$. The gap $G = N_g(t_{tx}) - N_g(t_a)$ tells us exactly how many TSs (empty or not) remain to be served and therefore gives an approximate idea of the congestion level in the network. This information is available at every listening station and may be used as an estimate of the network load.

The two traffic classes will be queued independently (there will be Transmissions Sets associated to data packets and TSs associated to real-time packet arrivals), but as long as the real-time gap $G$ remains under certain value $Th$ a data TS and a real-time TS are served simultaneously (i.e. it is equivalent to consider that we serve a TS which is the union set of the TS for real-time traffic and the TS for data traffic). When both of them get empty the two next TSs are processed.

When the real-time gap $G$ becomes bigger than $Th$ only real-time TSs are served (although the generation of both data TSs and real-time TSs proceeds normally), and it is not until the real-time $G$ goes below $Th$ that the data TSs service is resumed.

---

3This case implies that the station has no packets to transmit during the TS in service, as otherwise it would transmit instead of detect the pilot tone.
5 The Models

Our main objective is to evaluate the density function of the delay suffered by a packet that arrives in the system at an arbitrary point of time. This delay density function will allow us to assess delay percentiles and therefore estimate the goodness of the protocol to support time-bounded services. We have developed both a simulation model and an approximate analytical model.

As far as the input traffic is concerned we assume, for both models, that:

- We have $m$ stations in the network.
- The joint packet arrival process is Poisson with rate $\lambda$ (arrivals per slot $T_s$).

In the next subsections we will describe the assumptions, capabilities and limitations of the models to analyze the protocol behavior and performance.

5.1 Analytical Model

For the analytical model of the basic access method (i.e. without considering priorities) we have made the following simplifying assumptions [Cortizo et al. 1998]:

- The influence of signalling minislots in delay is neglected: we will not take into account the influence of empty Transmission Sets in the delay (these empty TSs have an associated constant processing time of one minislot). Similarly, we will not consider the minislot used by the non-empty TSs at the beginning of their service to emit a pilot tone and identify themselves as non-empty.
- We will only consider the case where $W = N$, that is, the length of the grouping intervals is made equal to the length of the packets transmitted. This imposes an important limitation on the model, which does not allow to measure the influence of this window size $W$ on the protocol performance. However, the simulation results will show that the choice $W = N$ is quasi-optimal for a wide range of traffic loads.
- The random backoff with adaptive window CSMA is replaced by a $p$-persistent CSMA. Instead of generating a random number $r$ and deferring transmission until $r$ minislots have elapsed, the stations will transmit with probability $p$ at every minislot. This way we obtain that the transmission deferral follows a geometric distribution. On the other hand, we are implicitly imposing a fixed size for the random backoff window. Adaptive mechanisms for the probability $p$ of transmission are not included in the analytical model.
- We will neglect the delay produced by the random backoffs introduced by the stations before transmission.
- We will bound the maximum number of packet arrivals per Transmission Set to a certain value $q_{max}$. 

11
From the items above we can conclude that in the simplified analytical model the system behaves as a queue which processes TSs. During the busy periods of our system we will have a sequence of Transmission Cycles (TC) with length $W$. In every TC a packet transmission occurs (either successful or collided) and a new Transmission Set is generated (either empty or not). When the system is idle neither TCs nor TSs are generated; it is the first next packet arrival who will reinitiate a busy period and the generation of both TSs and TCs. The main goal of the simplifications adopted is that they synchronize the instants when packets get successfully transmitted with the instants when new Transmission Sets are generated (see figure 2).

Another consequence of the simplifications above is that we will be able to deal most of the time with discrete-time density functions. Let us define the packet arrival instant $t_0$, the instant $t_1$ when the next Transmission Cycle begins and the effective transmission time $t_2$. The packet transmission delay is

$$d = t_2 - t_0 = t_2 - t_1 + t_1 - t_0 = R_1 + R_2$$

and consists of the sum of two independent terms: $R_1$, a fraction of the TC length which will be approximated by an integer number of minislots, and $R_2$, which is a multiple of the Transmission Cycle length $W$. The adopted procedure normalizes the TC length to unity and considers $R_2$ as a discrete-time function. As we will show in the following sections, the density function $f_{R_1}$ associated to $R_1$ is always a uniform ranging between zero and a certain maximum number of minislots less than $W$. We will assess the density function $f_{R_2}$ associated to $R_2$ (working with discrete functions), to finally evaluate the convolution $f_R = f_{R_1} * f_{R_2}$. It is very important to note that $f_R$ will also be a discrete-time function, but now the time unit is not anymore the slot length $W$, but the minislot length.
In order to determine at any time the state of the system let us define the pair \((G, q)\) such that:

- \(G\) is the number of non-empty Transmission Sets pending to be served.
- \(q\) is the number of packets pending to be transmitted in the Transmission Set in service.

With the assumptions we have made so far the pair \((G, q)\) can be viewed as a Markov renewal stochastic process, and observing the state of the system at the end of every TC we obtain an embedded Markov Chain. During the busy periods the time between transitions will follow a deterministic distribution with density function given by

\[
f_{\text{nonidle}}(t) = \delta(t - W \cdot t_s)
\]  

(2)

On the contrary, when the system gets idle -state \((0,0)\)- the time needed for a transition will be a random variable with density function

\[
f_{\text{idle}}(t) = \frac{\lambda}{W} e^{-\frac{\lambda}{W}(t - W \cdot t_s)} u(t - W \cdot t_s)
\]  

(3)

being \(u(t)\) the step function. Therefore, the time elapsed between transitions of the embedded Markov chain varies depending on the initial state:

- If the initial state means "busy system" the transition time is deterministic and equal to the length of the Transmission Cycle: \(W \cdot t_s\).
- If the initial state means "idle system", \((0,0)\), the transition time is a random variable given by eq. (3) with average \(T_s/\lambda+W \cdot t_s=W \cdot t_s(1+1/\lambda)\).

5.1.1 System State Probabilities & Vector Solution of the Embedded Markov Chain

The packet delay density function associated to the MAC protocol will be evaluated by conditioning on certain system state\(^*\) \((G, q)^*\).

The values \(\pi(G, q)^*\) represent the probabilities that a packet arrival (occurred at an arbitrary point of time) finds that the last completed Transmission Cycle moved the system into state \((G, q)\). \(\pi(G, q)^*\), therefore, is not the probability that an arbitrary packet arrives being the state of the system \((G, q)\), but the probability that the arbitrary packet finds that the last state transition of the embedded Markov chain was into the value \((G, q)\).

To calculate the system state probabilities \(\pi(G, q)^*\), we require the evaluation of the vector solution \(\mathbf{v}\) of the embedded Markov chain. In order to calculate this solution of the embedded Markov chain we will determine its one-step transition matrix \(\mathbf{P}\) and then

\(^*\)we use the (\(^*\)) to note that the system state at any time is taken as the system state at the last transition of the embedded markov chain.
the probability vector \( \mathbf{v} \) solution of the equation \( \mathbf{v} = \mathbf{vP} \).

Previous to the assessment of the vector solution \( \mathbf{v} \) we will truncate the maximum value of \( q \) to a certain value \( q_{\text{max}} \). This is equivalent to limiting the maximum number of arrivals per TS to \( q_{\text{max}} \). If this bound is chosen large enough its influence on the accuracy of the results will be negligible but, as we show below, this bounding will allow us to turn the embedded Markov chain into a Quasy-Birth-and-Death (QBD) process.

Let us define the \( M_{q_{\text{max}} \times q_{\text{max}}} \) matrices \( \mathbf{C}, \mathbf{D}, \) and \( \mathbf{E} \), as the matrices \( \mathbf{F} \in M_{q_{\text{max}}+1 \times q_{\text{max}}+1} \), \( \mathbf{E}_1 \in M_{q_{\text{max}} \times q_{\text{max}}+1} \) and \( \mathbf{D}_1 \in M_{q_{\text{max}}+1 \times q_{\text{max}}} \).

\[
\begin{align*}
\mathbf{C}(q, r) &= P[(G, q) \to (G, r)] & G \geq 1 & q, r = 1, \ldots, q_{\text{max}} \\
\mathbf{D}(q, r) &= P[(G - 1, q) \to (G, r)] & G \geq 1 & q, r = 1, \ldots, q_{\text{max}} \\
\mathbf{E}(q, r) &= P[(G, q) \to (G - 1, r)] & G > 1 & q, r = 1, \ldots, q_{\text{max}} \\
\mathbf{F}(q, r) &= P[(0, q - 1) \to (0, r - 1)] & q, r = 1, \ldots, q_{\text{max}} + 1 \\
\mathbf{D}_1(q, r) &= P[(0, q - 1) \to (1, r)] & q = 1, \ldots, q_{\text{max}} + 1 & r = 1, \ldots, q_{\text{max}} \\
\mathbf{E}_1(q, r) &= P[(1, q) \to (0, r - 1)] & q = 1, \ldots, q_{\text{max}} & r = 1, \ldots, q_{\text{max}} + 1.
\end{align*}
\]

The expressions for these transition probabilities are (only the nonzero probabilities are indicated)

\[
\begin{align*}
\mathbf{F}(1, i) &= \begin{cases} 
 e^{-\lambda} & i = 1 \\
 x(0, A)x(i - 1, B) + \sum_{j=1}^{i-2} x(j, A)x(i - 2 - j, B) & i = 2, \ldots, q_{\text{max}} + 1
\end{cases} \\
\mathbf{F}(2, i) &= x(i, \lambda) & i = 1, \ldots, q_{\text{max}} + 1 \\
\mathbf{F}(i, i) &= e^{-\lambda}c(i, p) & i = 2, \ldots, q_{\text{max}} + 1 \\
\mathbf{F}(i, i - 1) &= e^{-\lambda}t(i, p) & i = 2, \ldots, q_{\text{max}} + 1.
\end{align*}
\]

\[
\begin{align*}
\mathbf{D}(q, q) &= \mathbf{D}_1(q + 1, q) = (1 - e^{-\lambda})c(q, p) \\
\mathbf{D}(q, q - 1) &= \mathbf{D}_1(q + 1, q - 1) = (1 - e^{-\lambda})t(q, p) \\
\mathbf{C}(q, q) &= e^{-\lambda}c(q, p) \\
\mathbf{C}(q, q - 1) &= e^{-\lambda}t(q, p) \\
\mathbf{C}(1, q) &= x(q, \lambda) \\
\mathbf{E}(1, q) &= \mathbf{E}_1(1, q + 1) = e^{-\lambda}x(q, \lambda)/(1 - e^{-\lambda})
\end{align*}
\]
Being $p$ the transmission probability in one minislot of the $p$-persistent CSMA, $A = \lambda/W$, $B = \lambda - A$ and

$$x(i, \Delta) = \begin{cases} 
    e^{-\Delta \frac{\Delta^i}{i!}} & i = 0, \ldots, q_{\text{max}} - 1 \\
    1 - \sum_{j=0}^{q_{\text{max}}-1} x(j, \Delta) & i = q_{\text{max}}
\end{cases}$$

$$t(i, p) = \frac{ip(1-p)^{i-1}}{1-(1-p)} \quad i = 1, \ldots, q_{\text{max}}$$

$$c(i, p) = 1 - t(i, p) \quad i = 1, \ldots, q_{\text{max}}.$$  

With the above definitions matrix $P$ can be written as:

$$P = \begin{pmatrix}
    F & D_1 & 0 & 0 & 0 & \cdots \\
    E_1 & C & D & 0 & 0 & \cdots \\
    0 & E & C & D & 0 & \cdots \\
    0 & 0 & E & C & D & \cdots \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix}.$$  

From the structure of $P$ we see that it corresponds to a Quasi-Birth-and-Death (QBD) process. Therefore we will assess the vector solution $v$ applying any of the existing algorithms [Neuts 1981], [Stewart 1994] to solve QBD markov chains.

Once we have found the vector solution $v$ for the embedded markov chain we make use of both

- The PASTA (Poisson Arrivals See Temporal Averages) property, which applies as we have assumed Poisson arrivals.
- The key Markov renewal theorem [Cinlar 1975].

To calculate the state probabilities $\pi(G, q)^*$:

$$\pi(G, q)^* = \begin{cases} 
    \frac{v(G, q)}{1+v(0,0)/\lambda} & (G, q) \neq (0,0) \\
    \frac{v(0,0)}{1+v(0,0)/\lambda}(1 + \frac{1}{\lambda}) & (G, q) = (0,0)
\end{cases}.$$  

These values will be needed to assess the density functions of the delays suffered from the data packets from their generation up to the instant when they are completely and successfully transmitted.

### 5.1.2 Delay Associated to the "Non-idle" States

The state $(G, q)^*$ means "non-idle" as long as $(G, q)^* \neq (0,0)^*$. In this section we will determine the delay $d$ associated to the packets arriving at non-idle states.

In order to obtain expressions for the total delay $d(G, q)$, we find it convenient to express $d$ as the sum of four mutually independent random variables $d_0, d_1, d_2$ and $d_3$, in
a similar way as is done in [Polyzos et al. 1993]. The state \((G,q)^*\) determines that there is a Transmission Set already in service which has \(q\) packets still pending to transmit and \(G\) non-empty TSs queueing to be served before the start of service of the TS associated to the packet arrival. \((d_0 + d_1)\) is the residual life of the TS already in service (transmission of \(q\) packets). Recalling the eq. \((1)\), \(d_0 = R_0\). This means that \(d_0\) is the residual number of minislots until the end of the TC in service at the packet arrival. \(d_1\) is the number of TCs needed to complete the service of the TS in service at the packet arrival. \(d_2\) is the delay (in number of TCs) associated to the service of \(G\) non-empty TSs (with an undetermined number of packets every one of them) and \(d_3\) is the time (also in number of TCs) elapsed between the start of service of the TS associated to the packet arrival and the effective transmission of the packet of interest.

\[
d = d_0 + d_1 + d_2 + d_3
\]  

(10)

and as \(d_0, d_1, d_2\) and \(d_3\) are i.i.d. we have that

\[
f_d[k] = f_{d_0}[k] \cdot f_{d_1}[k] \cdot f_{d_2}[k] \cdot f_{d_3}[k]
\]

(11)

being \(f_{d_i}[k]\) the density functions associated to \(d_i, i = 0, 1, 2, 3\).

It is trivial to show that

\[
f_{d_0}[k] \sim U\text{minislots}(0, W)
\]

(12)

In order to evaluate \(f_{d_i}(k), i = 1, 2, 3\), which are treated as discrete-time functions being the time unit the slot length \((W\) minislots), let us first define

\[
f_{Tx,n}[k] = (1 - p_{col,n})p_{col,n}^k u[k] * \delta[k - 1]
\]

(13)

being \(p_{col,n}\) the probability of collision if there are \(n\) stations trying to transmit with probability \(p\), that is (see eq. \(7\))

\[
p_{col,n} = c(n, p)
\]

(14)

\(f_{Tx,n}[k], n = 1, ...,\) are the density functions of the delays associated to the effective transmission of just one packet when there are \(n\) stations trying to transmit with probability \(p\). We can use these functions to iteratively construct the density functions \(f_{TS/n}[k]\)

\[
f_{TS/n}[k] = \delta[k - 1]
\]

\[
f_{TS/n}[k] = f_{Tx,n}[k] \cdot f_{TS/n-1}[k] \quad n = 2, ...
\]

(15)

which are the density functions of the service time required by a Transmission Set with \(n\) packets. This way we have that the density function \(f_{TS}[k]\) of the time needed to serve an arbitrary Transmission Set is
\[ f_{TS}[k] = \sum_{n=1}^{\infty} P(n \text{ arrivals given that } n > 0) f_{TS/n}[k] = \sum_{n=1}^{\infty} \frac{e^{-\lambda} \lambda^n}{n!(1 - e^{-\lambda})} f_{TS/n}[k] \]  

(16)

but as we have limited the maximum number of arrivals per TS to \( q_{max} \) we have

\[ f_{TS}[k] = \sum_{n=1}^{n=q_{max}} \frac{x(n, \lambda)}{(1 - e^{-\lambda})} f_{TS/n}[k] \]  

(17)

with \( x(n, \lambda) \) defined in eq. 7.

With the functions defined above we can finally conclude that

\[ f_d(q)[k] = t(q, p) f_{TS/q-1}[k] + c(q, p) f_{TS/q}[k] \]  

(18)

\[ f_d(G)[k] = \begin{cases} \delta[k] & G = 0 \\ \frac{G}{f_{TS}[k] \ast \cdots \ast f_{TS}[k]} = f_{TS}[k]^G & G > 0 \end{cases} \]  

(19)

\[ f_d_s[k] = \sum_{n=1}^{n=q_{max}} \frac{x(n, \lambda)}{1 - e^{-\lambda}} f_{TX/TS=n}[k] \]  

(20)

\[ f_{TX/TS=n}[k] = \frac{1}{n} (f_{TX,n}[k] + f_{TX,n-1}[k] + \cdots + f_{TX,1}[k]) = \frac{1}{n} \left( \delta(k) + \sum_{j=1}^{j=n-1} f_{TX,n-1}[k] \ast \cdots \ast f_{TX,n-j}[k] \right) \]  

(21)

In the assessment of the expression (20) we condition on the number of packets existing in the Transmission Set, knowing that there is at least one, and then we consider equally likely all the possibilities in the transmission order of the packets: if there have been \( n \) packet arrivals, we consider equally likely \((1/n)\) that the packet of interest is transmitted in first, second, ..., \( n \)-th position. \( f_{TX/TS=n}[k] \) is the density function associated to the delay (in number of slots) caused by the service of the own TS, given that it has exactly \( n \) packets.

Once we have calculated the expressions in eqs. (18), (19), (20), we must assess \( f_d[k] \ast f_d[k] \ast f_d[k] \) which is also a discrete-time function with time unit the Transmission Cycle length (\( W \) minislots). Finally we must evaluate
\[ f_{d/non-empty}(G,q)[k] = f_{d_0}[k] * (f_{d_1}(q)[k] * f_{d_2}(G)[k] * f_{d_3}[k]) = \]
\[ = U_{\text{minislot}}(0,W) * (f_{d_1}[k] * f_{d_2}[k] * f_{d_3}[k]) \]

which is a discrete-time density function with time unit the minislot length.

5.1.3 Delay Associated to the "Idle" State

We consider that the state means "idle" if \((G,q)^* = (0,0)^*\). This does not imply that the system is actually empty when the packet arrives. The meaning of state \((0,0)^*\) is that the last transition of the embedded Markov chain, at the end of the last completed Transmission Cycle, left the system idle.

The time elapsed between transitions when the initial state is idle is a random variable given by eq. 3. The average time, normalized to the slot length \(T_s = Wt_s\), spent in the empty state \((1/\lambda + 1)\) is the sum of two terms:

- The average time required for the arrival of the first packet \((1/\lambda\), recalling that we have Poisson arrivals).
- The time elapsed from the first packet arrival instant to the next transition of the embedded Markov chain, which is deterministic and equal to 1.

Therefore, when a packet arrives and \((G,q)^* = (0,0)^*\), which happens with a probability given by eq. 9, there are two possible situations:

- The packet is the first one since the last transition and finds the system empty, with probability

\[ P(\text{first arr}) = \frac{1/\lambda}{1 + 1/\lambda} \pi_{(0,0)^*} = \frac{v(0,0)/\lambda}{1 + v(0,0)/\lambda} \]  

(23)

- There has been at least one packet that arrived before, but its transmission has not been completed yet, with probability

\[ P(\overline{\text{first arr}}) = \frac{1}{1 + 1/\lambda} \pi_{(0,0)^*} = \frac{v(0,0)}{1 + v(0,0)/\lambda} \]  

(24)

We will evaluate the delay density function \(f_{d/empty}\) by conditioning on these two possibilities

\[ f_{d/empty} = P(\text{first arr}) f_{d/first arr} + P(\overline{\text{first arr}}) f_{d/first arr} \]  

(25)

If the system is actually idle when the packet arrives then this arrival starts a Transmission Cycle where the packet will be transmitted successfully as long as there are no arrivals during the minislot following the packet arrival. If, on the contrary, there is any arrival in the shadowed interval (see figure 3) there will be a collision. After the failed
First P.A. starting transmission
P.A. causing a collision
P.A. deferring tx.
Transitions of the Embedded Markov chain

Figure 3: Packet Arrivals (P.A.) when the State is (0, 0)*

first TC is over, the transmission delay of the packet will be similar to $d_3$ (see eqs. 10, 20) and will depend on the number of Poisson arrivals during the first TC.

Whether the first arrived packet is successfully transmitted or not during the first TC, its delay will be an integer multiple of TCs, and so we will evaluate the delay density function $f_{d/first\ arr}$ as a discrete-time function, being the time unit the slot duration $T_s$.

\[
\hat{f}_{d/first\ arr} = P(suc)\hat{f}_{d/(first\ arr, suc)} + P(col)\hat{f}_{d/(first\ arr, col)}
\]

(26)

\[
P(suc)\hat{f}_{d/(first\ arr, suc)} = e^{-\lambda/W}\delta[k - 1]
\]

(27)

\[
P(col)\hat{f}_{d/(first\ arr, col)} = \left( \sum_{n=2}^{n=q_{max}} P_{first\ arr, col}(n) f_{Tx/TS=n[k]} \right) \delta[k - 1]
\]

(28)

where $f_{Tx/TS=n[k]}$ is given in eq.(21) and

\[
P_{first\ arr, col}(n) = \sum_{i=1}^{i=n-1} P(i\ arr.\ en\ t_s) P(n-1-i\ arr.\ en\ T_s-t_s) = \sum_{i=1}^{i=n-1} \frac{e^{-\lambda} \lambda^{n-1}}{i!(n-1-i)!} \frac{1}{W}^i \left( 1 - \frac{1}{W} \right)^{n-1-i}
\]

(29)

If the packet is not the first one, this arrival may take place with probability $(1/W)$ in the shadowed area of figure 3, forcing a collision, or, with probability $(1-1/W)$, may happen in the interval of length $(T_s - t_s)$. In the former case the delay is an integer multiple of $T_s$ while in the latter is not, as we show below.

The density function $\hat{f}_{d/first\ arr}$ can then be expressed as

\[
\hat{f}_{d/first\ arr} = \frac{1}{W} \hat{f}_{d/first\ arr, col} + \left( 1 - \frac{1}{W} \right) \hat{f}_{d/first\ arr, nocol}
\]

(30)

where
The density function \( f_{d/first\, arr,\, col} \), however, is not a multiple of \( T_s \), as there is a residual delay \( r \) from the packet arrival of interest until the end of the first TC originated by the first packet arrival. Anyway, as its associated density function \( f_r \sim U_{minislots}(0, W - 1) \) we can express

\[
f_{d/first\, arr,\, col} = f_r * f^*_{d/first\, arr,\, nocol} = U_{minislots}(0, W - 1) * f^*_{d/first\, arr,\, nocol}
\]

(32)

where \( f^*_{d/first\, arr,\, nocol} \) is a discrete-time delay density function which time unit is \( T_s \)

\[
f^*_{d/first\, arr,\, nocol} = \sum_{n=1}^{n=q_{max}} P_{no\, first\, arr}(n) f_{Tx/TS=n}[k]
\]

(33)

\[
P_{no\, first\, arr}(n) = \begin{cases} 
  e^{-\lambda} & n = 1 \\
  e^{-\lambda}(1 - \frac{1}{p}) & n = 2 \\
  e^{-\lambda} \sum_{i=1}^{n-2} \frac{1}{i!(n-2-i)!} (\frac{1}{p})^i (1 - \frac{1}{p})^{n-2-i} + \frac{\lambda}{(n-1)!} (1 - \frac{1}{p})^{n-1} & n \geq 3
\end{cases}
\]

(34)

5.2 Throughput of the System

In this section we will address the analytical evaluation of the maximum throughput of the system as a function of

- The Grouping Interval length \( W \) (in minislots).
- The packet length \( N \) (in minislots).
- The transmission probability per minislots \( p \) (associated to the size of the random backoff window).

We will find the expressions associated to both the average time needed for a non-empty Transmission Set to be generated and the average time needed to process it, values which will mainly depend on the Poisson arrival rate. For the system to be stable, it is obvious that the processing time cannot exceed the average interarrival time, and thereby we will evaluate the maximum throughput of the system as the Poisson arrival rate for which the processing time equals the interarrival time of non-empty Transmission Sets.
Let $P_{A\neq N\in E}(n)$ be the probability of $n$ packet arrivals in a Transmission Set given that there has been at least one packet arrival (i.e. the TS is non-empty). Using the notation introduced in eq. 7,

$$P_{A\neq N\in E}(n) = \begin{cases} \frac{\lambda^{|n\lambda W/N|}}{1-\frac{\lambda W}{N}} & n > 0 \\ 0 & n \leq 0 \end{cases}$$

(35)

Then the density function associated to the processing time of a non-empty TS is

$$f_D(k) = \sum_{n=1}^{\infty} P_{A\neq N\in E}(n) f_D/n(k)$$

(36)

The set of functions $f_D/n(k)$ can be calculated recursively

$$f_D/1(k) = f^1_{\text{cycle}}(k)$$

(37)

$$f_D/n(k) = f_D/n-1(k) * f^n_{\text{tx}}(k)$$

(38)

where

$$f^n_{\text{cycle}}(k) = (1 - (1-p)^n)(1-p)^{n(k-N-1)}u(k-N-1)$$

(39)

$$f^n_{\text{tx}}(k) = \sum_{i=1}^{\infty} t(n,p)(1 - t(n,p))^{i-1}[f^n_{\text{cycle}}(k)]^{*i}$$

(40)

$[f^n_{\text{cycle}}(k)]^{*i} = f^n_{\text{cycle}}(k) * \ldots * f^n_{\text{cycle}}(k)$ and $t(n,p)$ is as defined in eq. 7. It is also important to note that through the functions $f^n_{\text{cycle}}(k)$ we are taking into account both the idle minislots associated to the random backoff windows as well as the pilot tones emitted after every packet transmission. Finally, if we delay the resulting density function by one minislot $f_D(k) \leftarrow f_D(k) * \delta(k-1)$ we will include even the pilot tone emitted at the beginning of the TS processing by all stations having a packet to transmit.

The average processing time of the non-empty Transmission Sets is then

$$\mu_{TS}(\lambda, N, W, p) = E[f_D(k)]$$

(41)

Let us now address the calculation of the non-empty Transmission Sets’ interarrival time. The TSs are built by gathering the packet arrivals taking place during grouping intervals of length $Wt_s$. As we have assumed Poisson arrivals at rate $\lambda$ packets per slot (with length $Nt_s$), the probability of getting an empty TS is

$$P_{noA} = e^{-\lambda W/N}$$

(42)

We may then consider the probability of getting a non-empty TS as a geometric distribution with parameter $P_{noA}$ and therefore the average time needed for a non-empty TS to show up is
The stability condition is given by

$$IA_{TS}(\lambda) \geq \mu_{TS}(\lambda, N, W, p)$$

In the former expression we have not taken into account the processing time of the empty TSs. From eq. 43 we have that $\frac{1}{1-e^{-\lambda W/N}} - 1$ is the average number of grouping intervals until the first non-empty one. As the associated empty TSs have a processing time of one minislot we finally have that the stability condition is

$$IA_{TS}(\lambda) \geq \mu_{TS}(\lambda, N, W, p) + \frac{e^{-\lambda W/N}}{1-e^{-\lambda W/N}}$$

The maximum throughput of the system will be assessed by finding the maximum value of $\lambda$ which verifies the former inequality. For the values $p = 0.2$ and $W = N = 64$, which will be shown to be a good choice in the section of Simulations and Numerical Results, the maximum throughput is 0.8827 (see figure 4).

### 5.3 Simulation Model

Most of the limitations of the analytical model are overcomed by the simulation model, whose main features are:

- For the simulation model we have considered the influence on the delay distribution function of both the pilot tones involved in the protocol and the idle minislots associated to the random backoff in the CSMA algorithm.
As in the analytical model, we have replaced the random backoff with adaptive window CSMA by a \( p \)-persistent CSMA. We have also included as an option the following scheme for updating, after every Transmission Cycle, the probability \( p \)

\[
p_i = \begin{cases} 
  p_{\text{init}} & \text{if } i = 1 \\
  \min(p_{\text{max}}, p_{i-1} + 0.1) & \text{if } TC(i-1) \Rightarrow \text{Successful Transmission} \\
  \max(p_{\text{min}}, p_{i-1}/2) & \text{if } TC(i-1) \Rightarrow \text{Collision}
\end{cases}
\]  

which requires the choice of the values \( p_{\text{init}}, p_{\text{min}} \) and \( p_{\text{max}} \). Therefore, in the simulation model we can study the impact on performance of adapting the random backoff window size and compare it to the simpler case where the probability \( p \) (i.e., the random backoff window size) is non-adaptive, although we have not used this feature in this paper.

- The simulation model does not impose any predetermined length (\( W \) in minislots) to the intervals associated to the generation of Transmission Sets. This way, we will be able to study the variations in CDV due to different choices of this value \( W \).
- The priority policy via threshold has been implemented.

6 Simulations and Numerical Results

In this section we will analyze the behavior of the fixed-size random backoff window basic access mechanism (i.e. without priorities) using the models developed in the preceding sections. Although the possibility of updating the transmission probability \( p \) is included in the simulation model, we have limited in this paper the study to the non-adaptive \( p \)-persistent CSMA. We will also study the performance of the threshold policy for supporting real-time services.

For our study, and following the assumptions of section 1.1, we have chosen a minislots length of \( t_s = 0.6 \mu s \) and a packet length of \( N = 64 \) minislots. The graphs obtained by simulation present the average of five runs of the program with length 6400000 minislots. Also, the statistical distribution of the delays is shown in all graphs by depicting \( \log_{10}(1 - F_D(t)) \), being \( F_D(t) \) the delay distribution function.

In figure 5 we can see, for a fixed load \( p = 0.6 \), the dependence of the basic access mechanism performance on the transmission probability \( p \), optimal for a value \( p = 0.1/0.2 \), which strongly affects the distribution of the packet delay. This value might seem surprisingly low but, as it is a probability of transmission per minislot, we must observe that the longer delays due to idle minislots (with \( p = 0.2 \) a packet will be transmitted, on average, after 5 random backoff minislots) are well paid off by the drastic reduction of the collision probability (each collision adds a delay of 64 minislots).
Figure 5: Determination of the optimal $p$.

Figure 6: Determination of the optimal $W$.

Figure 7: Determination of the optimal $p$.

Figure 8: Determination of the optimal $W$. 
In figure 6 we have studied the influence of the value $W$, in the performance of the basic access mechanism. We see that the value $W$=packet transmission time seems to be a good choice for load 0.6. This result in turns makes more interesting the analytical models, which is restricted to this choice of $W$. Moreover, we have repeated the experiment in a low-loaded system ($\rho = 0.1$) but we have obtained results (see figures 7 and 8) which lead to the same considerations. In the rest of this section we always assume $W$ to be equal to the packet transmission time.

In figure 9 we validate the analytical model by comparing it with the results obtained by simulation corresponding to the basic access method. We have chosen a value for the transmission probability per minislot of $p = 0.2$ and depicted the complementary PDF of the delay for different loads. We can observe that the model gives always similar results to those obtained by simulation. For lower values of $p$ we expect an impairment in the behavior of the analytical approach, as it despises the idle minislots introduced by the random backoff windows.

In figures 10 and 11 we show the $10^{-3}$ and $10^{-6}$ quantiles of the packet delays for different network loads for the basic access mechanism (without implementing the threshold policy). These quantiles are defined as the minimum packet delay which is exceeded with a probability lower than $10^{-3}$ and $10^{-6}$, respectively. From these quantiles one can easily find estimations of the Cell Delay Variation introduced by the access protocol (see [COST 242 Final Report 1996]). The graphics show a good behaviour of the access protocol, allowing the support of real-time services for loads close to 0.8.

Figure 12 shows the threshold influence on the traffic delays for a real-time load of 0.2. On the left figure the offered data traffic is 0.3 and on the right one 0.6. The data and real-time delay curves are depicted for different threshold values. The curves labeled as "Analytical Model" correspond to the analytical model for the system without priorities.
Figure 10: $10^{-3}$ quantile for the packet delay.

Figure 11: $10^{-6}$ quantile for the packet delay.

Figure 12: Threshold influence on traffic delays.
Figure 13: Delays dependence on threshold.

Figure 14: Delays dependence on offered data traffic.

Figure 13 depicts the data and real-time delays dependence on the threshold value. The offered data load is 0.3 and the real-time load is 0.5. The threshold values tested are 2, 5 and 9. In Figure 14 it is shown the data and real-time delays dependence on offered data traffic for a fixed threshold. The real-time load is 0.2 and the threshold is 5.

Figures 12 and 13 depict the threshold policy results. The higher the threshold, the more similar the real-time and data delays are to those of the basic protocol (no priorities implemented) with a load equal to the sum of real-time and data traffic. On the other hand, if the threshold is set low enough the data and real-time delays diverge, the real-time delay converging to the delay experienced by the basic protocol with a load equal to only the real-time traffic.

In Figure 15 we can observe the offered data traffic influence on real-time traffic delay for a fixed threshold. The threshold value is 5 for both figures while the real-time offered load is 0.5 on the left one and 0.7 on the right one.

Figures 14 and 15 show the dependence of the real-time delay on the offered data traffic. This dependence is negligible for \( Th = 2 \). Moreover, the delay impairment the data traffic may cause is bounded.

Tables 1, 2 and 3 are related to Figures 14 and 15. They show how the data throughput is roughly equal to the difference between the maximum throughput of the basic protocol (no priorities) and the real-time throughput (which is always guaranteed as long as it does not exceed the capacity of the system. With high offered data loads there are losses as the simulator was built with a constraint in the maximum data buffer size.
Figure 15: Offered Data traffic influence on Real-time traffic delay.

<table>
<thead>
<tr>
<th>Offered Real-time Traffic = 0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold = 5</td>
</tr>
<tr>
<td>Offered Data Traffic</td>
</tr>
<tr>
<td>Data Throughput</td>
</tr>
<tr>
<td>Total Throughput</td>
</tr>
</tbody>
</table>

Table 1: Data Throughput for threshold 5 and real-time traffic 0.2

<table>
<thead>
<tr>
<th>Offered Real-time Traffic = 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold = 5</td>
</tr>
<tr>
<td>Offered Data Traffic</td>
</tr>
<tr>
<td>Data Throughput</td>
</tr>
<tr>
<td>Total Throughput</td>
</tr>
</tbody>
</table>

Table 2: Data Throughput for threshold 5 and real-time traffic 0.5

<table>
<thead>
<tr>
<th>Offered Real-time Traffic = 0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold = 5</td>
</tr>
<tr>
<td>Offered Data Traffic</td>
</tr>
<tr>
<td>Data Throughput</td>
</tr>
<tr>
<td>Total Throughput</td>
</tr>
</tbody>
</table>

Table 3: Data Throughput for threshold 5 and real-time traffic 0.7
7 Conclusions

In this paper we have presented a MAC protocol oriented to ad-hoc Wireless LANs supporting two different traffic classes, one with QoS guarantees. The protocol is a Group Resolution Algorithm, as it gathers into Transmission Sets the packet arrivals taking place in grouping intervals of fixed length. Every Transmission Set is served via an adaptive-window random backoff CSMA, and the different TSs are handled with FIFO discipline.

We have developed a simplified analytical model which only considers the situation where the packet lengths equal the size of the grouping intervals. The results obtained show that the protocol can support real-time services even with high network loads, while non real-time services make use of whatever capacity is left unused by real-time traffic.

References


