

Analytical Evaluation of a Medium Access Control Priority Mechanism for Wireless Ad Hoc Networks

J. Alonso-Zárate*, D. Gregoratti*, L. Alonso[†], and Ch. Verikoukis*

*Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Castelldefels, Barcelona, Spain

{jalonso, dgregoratti, cveri}@cttc.es

[†]Dept. of Signal Theory and Communications

Universitat Politècnica de Catalunya (UPC-EPSC), Castelldefels, Barcelona, Spain

luisg@tsc.upc.edu

Abstract—We present in this paper the analytical evaluation of a simple mechanism to reduce the average transmission delay of master stations in a network based on the Distributed Queuing MAC protocol for Ad hoc Networks (DQMAN). When DQMAN is executed, the network is self-organized into dynamic and spontaneous master-slave clusters. Within each cluster, a high-performance MAC protocol based on a tree-splitting collision resolution algorithm which uses access minislots is executed. By allowing temporary master stations to avoid contention to get access to the channel, their average packet transmission delay can be effectively reduced compared to that of slaves. This technique provides thus master stations with higher access priority to the channel and, indeed, could be used in any MAC protocol based on access minislots to provide a subset of users with higher priority.

I. INTRODUCTION

Tree-splitting collision resolution algorithms constitute a powerful tool to manage collisions in Medium Access Control (MAC) protocols for wireless networks. The main idea behind these techniques consists in splitting collided users into smaller groups, and those groups into smaller subgroups and so on, forming a tree of resolutions until the collision can be solved [1], [2]. Some of the existing MAC protocols based on these techniques allocate dedicated resources for this purpose (typically referred to as the contention window) so that the costly collision of data packets can be avoided. As a consequence, the performance of the overall system can be boosted.

In this paper we focus on the the Distributed Queuing MAC protocol for Ad Hoc Networks (DQMAN). DQMAN uses a tree-splitting algorithm based on access minislots and attains very high throughput in a totally distributed manner. This protocol is the extension of the Distributed Queuing Collision Avoidance (DQCA) protocol [3], originally designed for the uplink of Wireless Local Area Networks (WLANs), to operate over networks without infrastructure.

When DQMAN is executed, all the stations with data to transmit get access to the channel following a distributed access method similar to the Distributed Coordination Function (DCF) of the IEEE 802.11 Standard [4]. Whenever a station

gets access to the channel, it transmits its data packets and it also becomes the master of a one-hop temporary cluster where DQCA can be applied. The one-hop neighbors become slaves upon the reception of a periodic beacon broadcast by the master. Despite this master-slave structure, communications are done by establishing peer-to-peer links between source and destination within the cluster. The master is simply an indirect coordinator of the channel access and has no control over it. More specifically, the main responsibility of masters is to periodically transmit a control packet in order to:

- 1) Define a MAC frame structure to which stations can get synchronized and become slaves. This frame is divided into three parts which contain the access minislots wherein any station with data to transmit can send an access request, the data transmission part, and a control part for the broadcast of control information, respectively.
- 2) Broadcast the minimum control information required for the execution of the protocol rules [3].

Since the responsibility of operating in master mode may entail extra energy consumption, it would be desirable to provide stations with some kind of incentive to get this role when they have data to transmit if there is no other station already operating in master mode. Otherwise, if no station becomes master when there is data to transmit, DQMAN cannot be executed. This is the motivation of the work presented in this paper.

The main contribution of this paper is the description and theoretical analysis of a simple mechanism to encourage stations to become master despite the associated extra power consumption. In short, the idea is that it is possible to reduce the average packet transmission delay of master stations by allowing them to avoid contention when attempting to get access to the channel. Since master stations are responsible for distributing the required control information for the execution of the DQCA rules, they can smartly handle this information to avoid contention (for their own data transmissions) without incurring in unfairness.

Numerical results show that the proposed technique reduces the average packet transmission delay of masters compared to that of slaves. Further, since the master does not take part in the

This work has been supported by NEWCOM++ (ICT-216715), LOOP (FIT-330215-2007-8), PERSEO (TEC2006-10459/TCM), COOLNESS (218163-FP7-PEOPLE-2007-3-1-IAPP), and CENTENO (TEC2008-06817-C02-02).

contention for the channel, the overall network performance is improved as the total number of contenting users is decreased by one.

The remainder of the paper is organized as follows. An overview of DQMAN is provided in Section II. The proposed mechanism for masters is described and analyzed in Section III. Numerical evaluation of the efficiency of the priority mechanism is presented in Section IV. Finally, Section V concludes the paper.

II. DQMAN OVERVIEW

The key features of DQMAN are overviewed in the following subsections for the sake of the understanding of the herein proposed mechanism.

A. Clustering Overview

When DQMAN is applied, stations are spontaneously organized into temporary master-slave clusters whenever there is data to transmit in the network. There is no explicit clustering formation overhead and the cluster membership is soft-binding, i.e., there are no tight association and disassociation processes. A station belongs to a cluster as long as it can receive the control packets transmitted by the master. The clustering algorithm is based on the CSMA-based access of the Distributed Coordination Function of the IEEE 802.11 Standard [4].

Three modes of operation are defined, namely, i) *idle*, ii) *master*, and iii) *slave*. Any station should be able to operate in any of these three modes of operation and switch from one to another whenever required.

Whenever any idle station has data ready to transmit, it senses the radio channel for a randomized period of time seeking for an already established master (an active cluster). If a master is found, then the idle station becomes slave and gets synchronized with this cluster. Masters are responsible for the periodical broadcast of a control packet including a PHY preamble that allows for clock synchronization. Otherwise, if the channel is idle for the whole sensing interval, the idle station sets itself to master and starts to broadcast its own synchronizing control packet every T_{frame} seconds.

A master operates as such for as long as there is data activity in its cluster. Whenever its own data buffers are empty and there are no new access requests for the channel, the master reverts to idle mode. Therefore, the cluster topology varies as a function of the traffic load of the network. Under heavy traffic conditions, the lifetime of a cluster is bounded to a maximum time so that the responsibility of being master is shared among all the stations of a network.

B. MAC Overview

Within each cluster, a control packet, named Feedback Packet (FBP), is broadcast by the master every T_{frame} seconds. The FBP defines a time-frame MAC structure, as depicted in Fig. 1 and allows for clock synchronization within the cluster. This frame is divided into three main parts of different duration:

- 1) The first part, referred to as the Contention Window (CW), is further divided into m access minislots wherein those stations with data ready to be transmitted send an Access Request Sequence (ARS) in one of the minislots selected at random (with equal probability). The format of an ARS is described below.
- 2) The second part is devoted to the almost free-of-collision transmission of data packets.
- 3) The third part is devoted to the exchange of control information between the stations of the network. It consists of one slot reserved for the transmission of ACK packets from any destination upon the reception of a data packet and a slot reserved for the FBP.

Short Inter Frame Spaces (SIFS) are left between the different parts of the frame in order to tolerate non-negligible propagation delays, switching times from receiving to transmit mode of the RF receivers, and for processing time requirements.

The FBP contains ternary feedback information (success, idle, or collision) related to the status of all the m access minislots at the beginning of the frame. Without diving into the details of the protocol, the general idea is that active stations are organized into two distributed queues that manage both the data transmission and the resolution of collisions. They are the Collision Resolution Queue and the Data Transmission Queue, respectively. These queues are updated in a distributed manner at each station with the control information broadcasted within the FBP. Each of these queues are simply represented at each station with two integer numbers; one indicating the total number of stations in the queue, and another one indicating the position of the station in the queue.

When a station has data to transmit, it sends an ARS within one of the randomly selected access minislots at the beginning of the frame. Since an ARS does not need to contain any explicit information, it may consist of a short detectable sequence of chips. A method for the detection of the state of the minislots with pseudo-random sequences is the subject of a patent [5]. Therefore, the master is able to decide whether each minislot was idle (no ARS sent by any station), successful (only one ARS transmitted), or a collision occurred (more than one ARS, regardless of how many).

Using the ternary feedback information attached to the FBP, the requesting station learns whether its ARS was successful or not. In the case of success, it is queued in the Data Transmission Queue and waits until it reaches the first position in the queue to transmit its data. In the case of collision, the involved stations are queued in the Collision Resolution Queue and try to solve their collisions orderly in time following a blocked access tree-collision resolution algorithm [3].

III. ACCESS PRIORITY MECHANISM

A. Motivation

Since the fundamentals of the MAC protocol executed within each cluster of DQMAN are based on DQCA, the average transmission delay calculated in [6] for $K = 1$ is

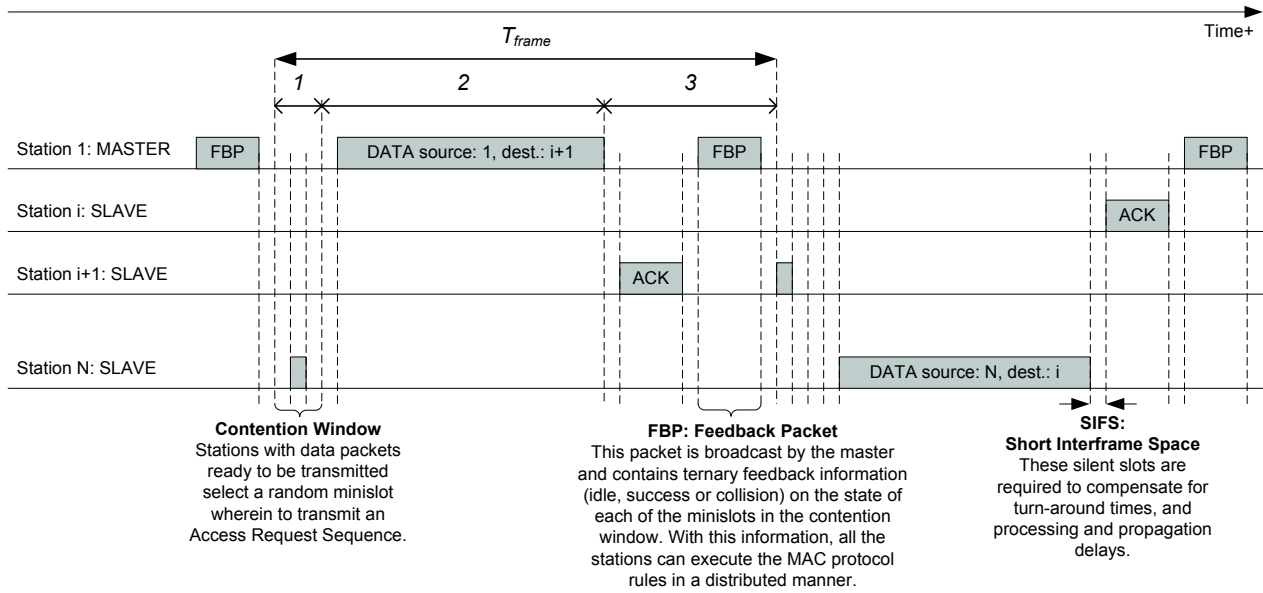


Fig. 1. DQMAN Frame Structure

also valid for DQMAN (once a cluster is set). This model consists of two M/M/1 queuing systems set in tandem; the first one models the collision resolution subsystem of the protocol, and the second one models the data transmission subsystem. According to this model, the average message transmission delay can be computed as

$$E[t] = E[t_{ETI}] + E[t_{RQ}] + E[t_{TQ}]. \quad (1)$$

where,

- $E[t_{ETI}]$ is average duration of the Enable Transmission Interval and it is equal to half the duration of a MAC frame. This time corresponds to the average time a newly arrived message waits until the contention process can be initiated at the beginning of the next frame.
- $E[t_{RQ}]$ is the average contention time.
- $E[t_{TQ}]$ is the average delay caused by the data transmission subsystem.

The exact expression of these terms can be found in [6] for $K = 1$ and have not been included in this paper due to space constraints. Analyzing this model, it is worth emphasizing that within the context of DQMAN, master stations incur in the same average message transmission delay as the associated slaves. Recall that masters only provide feedback information on the state of the access minislots, but they have no control on the access to the channel.

However, since master stations are responsible for broadcasting the state of each of the access minislots at the end of a frame, they could avoid contention with other stations when accessing to the channel. Therefore, their average message transmission delay can be effectively reduced. This mechanism is presented and analyzed in the next subsections.

B. Description

Whenever a master has a data message to transmit, it reports a successful request in an empty minislot whenever it has data to transmit without actually sending any ARS.

Since the master senses the state of each of the access minislots within a MAC frame, it can wait until at least one minislot is empty in a given frame where to report a successful ARS. Therefore, contention is completely avoided. Recalling (1), in order to compute the average message transmission delay of a master, the term $E[t_{RQ}]$ can be substituted by the average time required to find an empty minislot wherein a successful ARS can be reported.

This calculation is presented in the next subsection and allows comparing the average message transmission delay of master and slave stations and evaluating, thus the efficiency of this simple mechanism to prioritize the access of masters.

C. Analysis

Due to the framed nature of the protocol, the interest is on computing the average number of frames that a master has to wait (in average) to find at least one minislot empty.

Recall that each MAC frame has m access minislots. In each one of these minislots an ARS may be sent by any station needing to transmit a packet. Then, the average number of frames needed to find at least one free minislot (without including the frame with the empty one), denoted by $E[T_{fm}]$, is calculated as

$$E[T_{fm}] = \left[\sum_{k=0}^{\infty} k (1 - P_f)^{k-1} P_f \right] - 1 = \frac{1}{P_f} - 1. \quad (2)$$

P_f is defined as the probability of finding at least one free access minislot in a given frame, and, according to the total probability theorem, it can be computed as

$$P_f = \sum_{k=0}^{\infty} P_{f|k} P(k). \quad (3)$$

$P(k)$ is the probability of having exactly k arrivals (access requests) into the system in a given MAC frame. This value

depends on the specific arrival distribution from all active stations in the network.

On the other hand, $P_{f|k}$ is defined as the probability of having at least one free access minislot given that there are k arrivals into the system. This probability is 1 for $k < m$. Otherwise, it can be computed turning to combinatorics. Indeed, the k arrivals can be parceled out into any non-empty subset of minislots, being the subsets of interest all those that leave out, at least, one empty access minislots. Therefore, for any $k \geq m$, the probability $P_{f|k}$ can be computed as

$$P_{f|k} = \frac{\sum_{j=1}^{m-1} j! \binom{m}{j} S(k, j)}{\sum_{j=1}^m j! \binom{m}{j} S(k, j)}. \quad (4)$$

The terms $S(k, j)$ in (4) are the Stirling Numbers of the Second Kind [7]. They are defined, for any $k \geq j$, as the number of ways of partitioning a set of k elements into j non-empty subsets and can be calculated as follows:

$$S(k, j) = \frac{1}{j!} \sum_{i=0}^j (-1)^i \binom{j}{i} (j-i)^k. \quad (5)$$

This formula considers neither the order of the access minislots in which the arrivals are partitioned nor the order of the arrivals within each minislot. However, since the minislots are ordered in time, ordering must be taken into account and a multiplying factor $j!$ has to be added to (5). In addition, the factor $\binom{m}{j}$ is necessary to consider all the possible combinations of selecting j minislots out of m . These two additions have been already included in (4). The accuracy of (4) is shown in Fig. 2 where its value has been compared to that obtained by computer simulation as a function of the number of arrivals k and for different number of access minislots.

Finally, using (4) and (5) into (3), and recalling (2), the average number of frames needed to find at least one free

minislot (without including the frame with the empty one) rewrites as

$$E[T_{fm}] = \frac{1}{P_f} - 1 = \left[\sum_{k=0}^{m-1} P(k) + \sum_{k=m}^{\infty} \frac{\sum_{j=1}^{m-1} \binom{m}{j} \sum_{i=0}^j (-1)^i \binom{j}{i} (j-i)^k}{\sum_{j=1}^m \binom{m}{j} \sum_{i=0}^j (-1)^i \binom{j}{i} (j-i)^k} P(k) \right]^{-1} - 1. \quad (6)$$

Using this expression in (1) it is possible to evaluate the efficiency of the prioritization mechanism presented in this paper. This evaluation is presented in the next section.

IV. NUMERICAL EVALUATION

The numerical evaluation of the reduction of the average message transmission delay of masters is presented in this section considering a Poisson arrival distribution. This means that

$$P(k) = \frac{\lambda_t^k}{k!} e^{-\lambda_t}, \quad (7)$$

where λ_t is the total aggregate arrival rate.

Recall the expression of the average message transmission delay introduced in (1) and fully described in [6]. By comparing this expression with the same equation but substituting the term $E[t_{RQ}]$ by (6) it is possible to evaluate the efficiency of the prioritization mechanism in terms of reduced average message transmission delay. Considering the Poisson arrival distribution, the term $E[t_{RQ}]$ can be derived from [6] (for $K = 1$) and expressed as

$$E[t_{RQ}] = \frac{1}{\mu_{RQ}} \left(1 + \frac{(\lambda/\mu_{RQ})}{(1 - (\lambda/\mu_{RQ}))} \right), \quad (8)$$

where

$$\frac{1}{\mu_{RQ}} = \frac{1}{\ln \left(\frac{1}{1 - e^{-\lambda/m}} \right)}. \quad (9)$$

Note that the term λ/m in (9) is the total traffic load offered to each of the m access minislots. Considering the priority access mechanism described in this paper, the input rate that has to be considered in the computation of (8) and (9) is only the contribution of the slaves.

A. System Model

A network composed of $n = 10$ stations, all of them within the transmission range of each other, is considered.

Time is normalized to the duration of a MAC frame. Each of the n stations generate the same amount (in average) of data traffic according to a Poisson arrival distribution with rate λ_i messages/frame. The average transmission time of a message has an exponential distribution with mean $1/\mu$, i.e., $1/\mu$ is

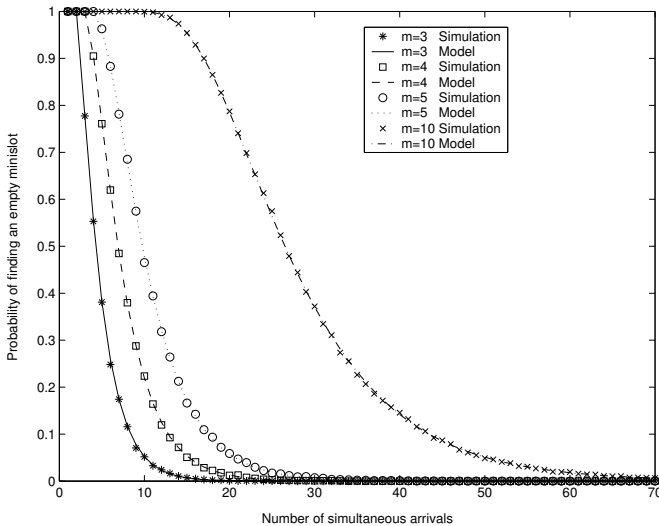


Fig. 2. Probability of finding an empty minislot in a given frame

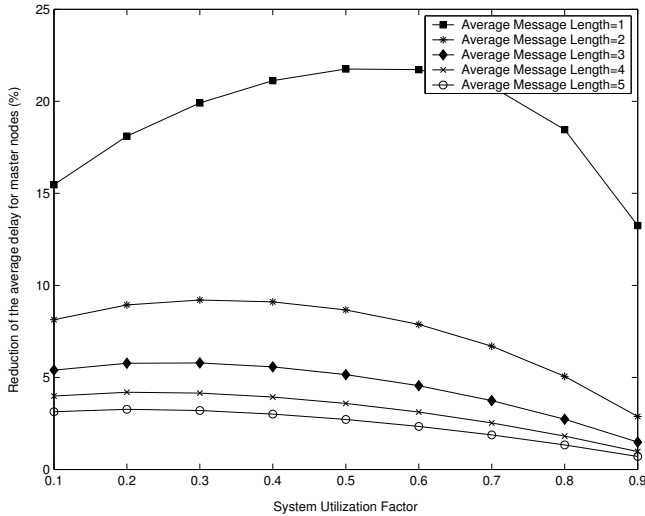


Fig. 3. Reduction of the average message transmission delay for masters compared to slaves

the average number of frames required to transmit an entire message. Therefore, the total offered load to the network is a Poisson process of total input rate

$$\lambda_t = \sum_{i=1}^n \lambda_i = n\lambda_i. \quad (10)$$

The total offered traffic input rate can be split into two components as $\lambda_t = \lambda_m + \lambda_s$, with λ_m being the traffic offered by the master station and λ_s being the traffic rate offered to the network by the slave stations. Therefore, λ_s is the input rate to be considered for the computation of (8) and (9).

Accordingly, the system utilization factor, i.e., the probability that the system is busy or, in other words, the probability that at least one station has data to transmit or is already transmitting, is denoted by ρ and defined as

$$\rho = \lambda_t / \mu. \quad (11)$$

Numerical results are presented in the next subsection to show the efficiency of the access priority mechanism presented in this paper.

B. Results

The ratio (in percentage) of the average message transmission delay of a station operating in master mode with respect to the average contention delay perceived by the slaves is illustrated in Fig. 3. Different curves have been plotted as a function of the system utilization factor and for different average message lengths (number of data packets per message). It is worth mentioning that the results show the existence of an optimal system utilization factor at which the delay reduction benefit is maximized.

In any case, the average packet transmission delay transmitted by the master is lower than that of slaves in all cases. In addition, as it could be expected, the average message length, which corresponds to the average number of packets transmitted per successful access request, has a remarkable impact on the efficiency of the proposed prioritization technique.

For short message lengths (1 or 2 packets per message), the considered mechanism is very efficient in giving priority to the master in a cluster. For example, for an average length of one packet per message a master station experiences an average transmission delay reduction of up to approximately 22%. On the contrary, as the average message length increases, the contention time becomes smaller compared to the actual data transmission time, and thus, the benefits of the prioritization technique become less significant. Recalling (1), if $E[t_{TQ}] \gg E[t_{RQ}]$, a reduction of the contention time does not have a great impact on the overall message transmission delay.

Since messages transmitted over wireless channels tend to be short to combat the wireless channel impairments, in most practical cases, the potential benefits that a master may achieve in terms of average packet transmission delay may payoff its extra energy consumption in a DQMAN network.

V. CONCLUSIONS

We have presented in this paper the analysis of a simple technique to encourage users to become master in a DQMAN-based network. The idea is that master stations can avoid contention to get access to the channel by simply notifying of a successful access request without actually sending it. The only requirement is that they have to wait until there is an empty access minislot in a give MAC frame to notify the success. We have evaluated in this paper the delay caused by this waiting time and numerical results show that their average packet transmission delay can be effectively reduced in comparison to that of slaves. Therefore, stations can perceive the role of master as a privilege and may will to become master despite the associated extra power consumption.

Since the mechanism does not deal with the rules of DQMAN, but with the smart use of the feedback information regarding the state of the access minislots, this technique could be applied to any MAC protocol based on access minislots in order to provide a subset of users with better performance or even prioritized access.

REFERENCES

- [1] D. Bertsekas and R. Gallager, *Data Networks*. NJ: Prentice Hall, 1992.
- [2] R. Rom and M. Sidi, *Multiple Access Protocols*. Springer-Verlag NY, 1990.
- [3] J. Alonso-Zárate, E. Kartsakli, A. Cateura, C. Verikoukis, and L. Alonso, "A near-optimum cross-layered distributed queuing protocol for wireless LAN," *IEEE Wireless Communications Magazine*, vol. 15, no. 1, pp. 48–55, January 2008.
- [4] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2007.
- [5] G. Campbell *et al.*, "Method and apparatus for detecting collisions and controlling access to a communications channel," U.S. Patent US6408009 B1, 2002.
- [6] L. Alonso, R. Agustí, and O. Sallent, "A near-optimum MAC protocol based on the distributed queueing random access protocol (DQRAP) for a CDMA mobile communication system," *IEEE JSAC*, vol. 18, pp. 1701–1718, September 2000.
- [7] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th printing. Dover NY, 1972, ch. Stirling Numbers of the Second Kind, section 24.1.4, pp. 824–825.