A Near-Optimum Cross-Layered Distributed Queuing Protocol for Wireless LAN

J. Alonso-Zárate♦, E. Kartsakli*, A. Cateura*, C. Verikoukis♦ and L. Alonso*

♦ Telecommunications Technological Center of Catalonia (CTTC)
Av. Del Canal Olímpic s/n, CTTC, 08860 Castelldefels (Spain)
E-mail: [jesus.alonso, cveri]@cttc.es
*Department of Signal Theory and Communications, Technical University of Catalonia (UPC)
Av. Canal Olímpic s/n, EPSC, Campus UPC 08860 Castelldefels (Spain)
E-mail: [ellik, cateura, luisg]@tsc.upc.edu

Indexing Terms: WLAN MAC, Cross-Layer techniques, Distributed Queues, IEEE 802.11, DQCA.

Abstract — **Distributed Queuing Collision Avoidance (DQCA) is a distributed MAC protocol for WLAN systems that offers near optimum performance. The protocol implements a reservation scheme that ensures collision-free data transmissions for high traffic load and switches smoothly and automatically to a random access mechanism when the traffic load is light, improving the delay performance for this situation. In this paper, the DQCA protocol operation is thoroughly described and its algorithm rules are given. Furthermore, an enhanced cross-layer scheduling mechanism is also proposed to be included into the protocol procedure. This mechanism employs a Virtual Priority Function to reschedule transmissions according to a cross-layer design. Two possible configurations are described in this paper by including a PHY-MAC dialogue involving channel state information and the waiting time of the packets in the system, offering a trade-off between throughput maximization and fairness. The performance in terms of throughput and mean delay of DQCA and the two cross-layer schemes has been evaluated through simulations and a significant enhancement with respect to the legacy IEEE 802.11 operation is achieved. The obtained results emphasize the advantages of the proposed schemes and the importance of cross-layer design in wireless communication systems.**

A Near-Optimum Cross-Layered Distributed Queuing Protocol for Wireless LAN

J. Alonso-Zárate[♦], E. Kartsakli*, A. Cateura*, C. Verikoukis[♦] and L. Alonso*

[♦]Telecommunications Technological Center of Catalonia (CTTC)
Av. Del Canal Olímpic s/n, CTTC, 08860 Castelldefels (Spain)
E-mail: [jesus.alonso, cveri]@cttc.es

*Department of Signal Theory and Communications, Technical University of Catalonia (UPC)
Av. Canal Olímpic s/n, EPSC, Campus UPC 08860 Castelldefels (Spain)
E-mail: [ellik, cateura, luisg]@tsc.upc.edu

Indexing Terms: WLAN MAC, Cross-Layer techniques, Distributed Queues, IEEE 802.11, DQCA.

Abstract — **Distributed Queuing Collision Avoidance (DQCA) is a distributed MAC protocol for WLAN systems that offers near optimum performance. The protocol implements a reservation scheme that ensures collision-free data transmissions for high traffic load and switches smoothly and automatically to a random access mechanism when the traffic load is light, improving the delay performance for this situation. In this paper, the DQCA protocol operation is thoroughly described and its algorithm rules are given. Furthermore, an enhanced cross-layer scheduling mechanism is also proposed to be included into the protocol procedure. This mechanism employs a Virtual Priority Function to reschedule transmissions according to a cross-layer design. Two possible configurations are described in this paper by including a PHY-MAC dialogue involving channel state information and the waiting time of the packets in the system, offering a trade-off between throughput maximization and fairness. The performance in terms of throughput and mean delay of DQCA and the two cross-layer schemes has been evaluated through simulations and a significant enhancement with respect to the legacy IEEE 802.11 operation is achieved. The obtained results emphasize the advantages of the proposed schemes and the importance of cross-layer design in wireless communication systems.**

I. INTRODUCTION

Wireless communications have witnessed an unprecedented growth during the last years. The standards IEEE 802.11b and g for Wireless Local Area Network (WLAN) are now widely deployed, while newer versions of the standard are expected to come up in a near future applying Orthogonal Frequency Division Multiplexing (OFDM) and multiple antenna techniques at the physical layer (PHY) to improve the overall performance. Therefore, considerable amount of research has been directed towards the improvement of the PHY with more sophisticated transmission techniques and coding algorithms, and the medium access control (MAC) protocol at the Data Link Layer (layer 2), in order to bring new live to wireless communications. Regarding the latter, most of the work already done has been focused on incrementally improving the MAC protocol in the IEEE 802.11 Standard [1], which is based on a Carrier Sensing Multiple Access protocol with Collision Avoidance (CSMA/CA), executing a Binary Exponential Backoff (BEB) collision resolution algorithm (CRA) to solve the unavoidable occurrence of collisions in such an unpredictable wireless medium. It has been demonstrated that the performance of this protocol suffers from inefficiency caused by idle periods due to back off and high presence of collision as the offered traffic load grows [2]. This last phenomenon becomes critic with the impending and tightly demanding applications involving video on-demand or multimedia traffic, among others. Therefore, there is a need to develop completely novel MAC protocols that can support high performance requirements in terms of throughput, delay, jitter, fairness and power consumption, and which can also be adaptive to the dynamically changing conditions of the system, such as the traffic load or the number of nodes.

On the other hand, layering has been the dominating design methodology for wireless communications systems. In particular, WLAN systems follow the traditional Open System Interconnection (OSI) layer-based architecture where independency is the main principle for the seven defined layers (referred to as physical, data link, network, transport, session, presentation, and application layers). Even though layering simplifies design and facilitates the intersystem compatibility, it seems to mismatch the dynamic wireless channel. Therefore, WLANs can be optimized by enabling some information exchange among layers of the protocol stack. This general concept is known as cross-layer design [3] and is likely to make a breakthrough in the performance of WLANs.

In this context, an innovative and efficient scheme for WLAN systems, called Distributed Queueing with Collision Avoidance Protocol (DQCA) is presented. DQCA obtains a near-optimum performance close to the theoretical maximum system capacity. Indeed, its throughput bounds, calculated in [4] without incorporating cross-layer design, demonstrate its potential to outperform current MAC protocols in 802.11 Standards. This is achieved by adaptively combining the strengths of random access protocols under low traffic load, thus attaining low transmission delays, with the capabilities of reservation schemes for heavy traffic loads. Collisions are handled by a blocked-access splitting algorithm that works in parallel with the data transmission process.

DQCA is based on the Distributed Queueing Random Access Protocol (DQRAP) presented in [5] by Xu and Campbell initially for the distribution of Cable TV, which shows a near-optimum performance independently of the amount of active terminals and the offered data traffic. Since then, and due to the outstandingly near-optimum performance of DQRAP, some efforts have been done to adapt this two-distributed queues concept to different environments. This is the case of the Extended DQRAP (XDQRAP) [6] and the Prioritized (DQRAP) [7] for wired centralized networks, the Interleaved DQRAP for satellite applications with long propagation delays [8], or the

DQRAP based on Code Division Multiple Access (DQRAP/CDMA) conceived for 3G cellular networks [9]. In this paper, the preliminary theoretical analysis presented in [4] has been extended to adapt the dual distributed queuing paradigm to the specific requirements of WLANs. A formal description of DQCA is presented, including a cross-layer approach for the management of the distributed queues that outperforms the rigid and suboptimal FIFO (First In First Out) discipline.

The outline of this work is as follows: Section II includes the detailed description of the DQCA protocol for WLANs. Section III is devoted to present the cross-layer approach of DQCA to reschedule the data transmissions. Section IV presents three different possible configurations of DQCA (with and without cross-layer) whose performance is compared through software simulations to the IEEE 802.11 MAC protocol. Finally, Section IV is devoted to the conclusions and description of the future ongoing research on the topic.

II. DQCA Description

A. Protocol Overview

DQCA is a distributed high-performance medium access protocol designed for WLAN environments that behaves as a random access mechanism under low traffic conditions and switches smoothly and automatically to a reservation scheme when traffic load grows. DQCA eliminates back-off periods and guarantees collision-free data transmissions for high traffic loads, and its performance is independent of the number of nodes transmitting in the system. In addition, this near-optimum performance is not degraded under high traffic conditions (like slotted Aloha), and a maximum throughput is achieved and maintained even when the traffic load exceeds the channel capacity.

The main idea of DQCA is that the nodes may ask for channel access in a reserved time interval, thus confining collisions almost exclusively to a part of the frame. Any collisions are resolved by a blocked access *m*-ary tree-splitting CRA in a FIFO order, managed by a distributed queue. Once having successfully sent an access request, a node enters in another distributed queue (devoted to the scheduling of collision-free data transmission) and waits for its turn to transmit. The next section contains a thorough description of the DQCA protocol.

B. Protocol Description

An infrastructure wireless network wherein N nodes share a wireless channel in order to communicate with an Access Point (AP) is considered. The time axis is divided into continuous DQCA frames which consist of three parts of different duration, as depicted in Figure 1. The first two parts are devoted to the uplink communication from the nodes to the AP while the third part is reserved for the downlink broadcast of control information by the AP.

The first part, also referred to as Contention Window (CW) is divided into m control minislots. During those, and as long as no previous collisions in the system are pending to be resolved (blocked-access CRA), nodes with data ready to transmit request access to the channel by sending an Access-Request Sequence (ARS). As discussed in detail in [5], with at least 3 minislots (m=3), the collision resolution works faster than the transmission of data and therefore it is sufficient to ensure near-optimum performance of the protocol for any traffic load. Higher values of m attain slightly lower delay at the cost of increased overhead.

The operation of DQCA is based on the use of m-ternary feedback information on the state of each of the m minislots. The AP must be able to distinguish between idle, success and collision state for each control minislot and must broadcast this information at the end of each frame. Therefore, an ARS is not required to contain any kind of actual information but must be properly selected in order to enable collision detection. Adopting the technique described in [10], each node may be assigned a unique bit pattern such that when two or more ARS collide, the pattern of the overlapping signal is distinguishable from the original pattern of any single ARS.

The second part of the frame is reserved for the almost collision-free transmission of data packets by one node at a time. This part is of fixed byte length (the time duration will depend on the actual transmission rate, allowing therefore the implementation of bit rate adaptation techniques) and large messages are fragmented into smaller packets. In any case, the frame could be easily adapted to support variable length data packets.

In the last part of the frame the AP broadcasts a Feedback Packet (FBP) that contains:

i) Ternary feedback information on the state of all access minislots.

ii) An acknowledgement (ACK) to verify the correct reception of data packets.

iii) A final-message bit which is set to 1 when the last packet of a message has been received and to 0 when more packets of the same message are expected to follow.

In certain configurations, the FBP may contain additional information, as it will be explained later. Moreover, any possible ARQ strategy may be used regarding the ACK part of the FPB.

In order to compensate for propagation delays, turn around times (to switch from receiving to transmit mode), and for processing purposes, the uplink (CW and Data Slot) and downlink (FBP) transmissions are separated by a SIFS (Short Inter Frame Space).

The protocol uses two logical distributed queues: the Collision Resolution Queue (CRQ) and the Data Transmission Queue (DTQ) that handle the collision resolution and the data transmission scheduling, respectively. These two queues are represented at each station by simple integer values indicating the total length of the queues and the individual position of the station within the queues. Although each ARS is sent for a specific message, the position in the logical queue for that message is maintained by the requesting station as it was the station itself who were within the queue, having into account that this is only a logical allocation.

Therefore, the two queues are represented at every node by four integer counters denoted by TQ, RQ, pTQ and pRQ. TQ is the number of messages (or nodes) waiting for transmission in the DTQ, while RQ is the number of collisions waiting for resolution in the CRQ. All nodes should have the same TQ an RQ values (i.e. they represent distributed queues). The other two integers, pTQ and pRQ, represent the position of each node within the DTQ and the CRQ respectively and therefore may have different values for each node. They are set to zero whenever a node is in neither of the queues. Otherwise, their values range from 1 to TQ or RQ respectively, with the value 1 indicating the head of the queues. In principle, every node, including those that do not have a message to transmit, must update these counters upon the reception of the FBP at the end of each frame by executing the set of rules described below. However, it is also convenient to periodically send the TQ and RQ values within the FBP to allow inactive nodes to leave and reenter the system, which in addition will increase robustness against possible miscounting errors for all nodes.

Three sets of rules are defined and executed at each node upon decoding of the FBP. They are, in order of execution, the data transmission rules (DTR), indicating who can transmit data in the following frame, the request transmission rules (RTR), implementing the collision resolution algorithm and the queuing discipline rules (QDR), managing the update of the queues. The DQCA operation flow chart is illustrated in Figure 2 according to the previous discussion and the following rules:

a) *DTR (Data Transmission Rules)*

1) If there are no nodes waiting to transmit (TQ=0) and no collisions to be resolved (RQ=0), every node with a message ready to be sent transmits an ARS in a randomly selected control minislot and the first packet of the message in the data part. This rule is referred to as the immediate access rule, and constitutes the only possible cause of a collision in the data part. However, it avoids an empty data frame and improves the packet transmission delay when the traffic load is low, when the probability of having a collision is also low.
2) If a node is at the head of the data transmission queue (pTQ=1), it is enabled to transmit a packet in the following frame. If this packet is the last of a message, the node sets the final-message-bit to 1, otherwise it sets it to 0.

b) *RTR (Request Transmission Rules)*

1) If there are no collisions pending to be resolved (RQ=0) and there are transmissions scheduled (TQ>0), every node that is in neither queue (pTQ=0 and pRQ=0) and which has a message ready to be sent, randomly selects one of the *m* control minislots and transmits an ARS in the following frame.
2) If a node is at the head of the collision resolution queue (pRQ=1), it randomly selects one of the *m* control minislots and transmits an ARS in order to try to resolve the collision in the following frame.

c) *QDR (Queuing Discipline Rules)*

1) Each node increases the value of TQ by one unit for each control minislot with a successful state (indicating that a new node has entered the DTQ).
2) The value of TQ is reduced by one unit if a data packet has been successfully transmitted and the final-message-bit was set to 1, since the transmitting node will leave the queue.
3) If there are collisions pending to be resolved (RQ>0) the value of RQ is reduced by one unit, since those nodes at the head of CRQ will try to resolve their collision within the following frame.
4) The value of RQ is incremented by one unit for each control minislot where an ARS collision occurred.
5) Each node calculates its own position in the queues (values for pTQ and pRQ). If the node has transmitted an ARS in a particular control minislot and the state of this control minislot was “success” then the node sets its pTQ to the corresponding value at the end of TQ. It should be mentioned that the events of the control minislots are sorted by a time arrival criterion, meaning that a node with successful request at the first control

minislot enters the data queue before a node whose request was sent at the second control minislot. If the ARS has collided, the node calculates its position among all the present collisions and sets pRQ to the corresponding value at the end of RQ. If the node has not sent any request, then pTQ and pRQ follow the same update rules as TQ and RQ, respectively, as long as their initial values are non-zero.

III. DQCA Operational Example

Figure 3 gives an example of the DQCA operation for $m = 3$ control minislots, by illustrating two successive frames in time. Nodes are denoted by n_i and their respective pointer values by pTQ_{ni} and pRQ_{ni}. It is assumed that in the exactly previous (i.e. the $(k–1)^{th}$) frame the CRQ is empty (RQ = 0), the DTQ contains two nodes (n_1, n_2, with TQ = 2), and n_1 has transmitted a packet ($pTQ_{n1} = 1$). Three new messages have also arrived at nodes n_3, n_4 and n_5.

At the first (i.e the k^{th}) frame under examination, n_4 randomly selects the first control minislot and n_3 and n_5 the third one and transmit an ARS. In the data part of the same frame, n_1 continues and completes its message transmission. The AP detects the state of all the control minislots (Success – Idle – Collision respectively) and incorporates this information into the FBP. It also acknowledges the correct reception of the data packet and sets the final-message-bit to 1, indicating the exit of n_1 from the DTQ. The successful state of the first control minislot, causes all the nodes to increase TQ by one unit while the collision of the third minislot leads to an increase of RQ (RQ =1). TQ is then reduced by one for the successful data transmission. The nodes set their pTQ and pRQ values appropriately, with n_4 entering the DTQ in the second position ($pTQ_{n4} = 2$) and n_3 and n_5 entering the CRQ (pRQ = 1 for both).

In the CW of the second (i.e the $(k+1)^{th}$) frame the CRA is initiated and the two nodes in the head of the CRQ send an ARS to a randomly selected control minislot. In the meantime, n_2, who is at the head of DTQ, transmits the first packet of its message (with more to follow). The FBP reveals that the ARS and the data transmission have been successful and that the final-message-bit is 0. All nodes decrease RQ by one unit, since one collision has been resolved, and increase TQ by 2 (TQ = 4, RQ = 0) for the two successful ARS. The nodes n_3 and n_5 enter the DTQ by the order in which the ARS had been sent within the CW (first n_5 and then n_3). This process will continue for the next frames.

IV. DQCA Cross-Layer Design

The performance of WLAN can benefit from the exchange of information between different layers of the protocol stack. In this section a cross-layer mechanism for enhanced data transmission scheduling is proposed. In particular, instead of the FIFO discipline, the transmission order is determined by a Virtual Priority Function (VPF).

The FBP contains the required information to enable all nodes to calculate the VPF associated to every node in the data transmission queue (DTQ). Then, these values are sorted in descending order of VPF and the node with the higher VPF is enabled to transmit. In the case that more than one node have the same VPF value, priority is given to the one with the longest waiting time in DTQ, i.e., smallest pTQ value.

The VPF may allow the establishment of a cross-layer dialogue between different entities at different layers of the protocol stack striving to improve any figure of merit. On the other hand, by defining the VPF as a constant value, the operation without cross-layer dialogue may be considered. It is also worth mentioning that the exchange of information between stations and the AP may be done by attaching the cross-layer information to the downlink FBP.

V. Study Cases

The performance of DQCA has been evaluated through computer simulations in an infrastructure WLAN system where a number of data nodes communicate with an Access Point (AP) through a shared radio channel. The performance of three different configurations of DQCA with different VPFs is compared to the one attained by the standard IEEE 802.11b for WLAN [1]. The High Rate Direct Sequence Spread Spectrum (HR-DSSS) PHY defined in the IEEE 802.11b with four available rates (1, 2, 5.5 and 11 Mbps) has been considered, although the DQCA could also be applied above any other PHY specification. The considered system model and the selection of VPF are described next.

A. Channel Model

An error-free slow varying channel has been considered. The channel model calculates the probability of each node transmitting a certain available maximum bit rate. The model implements a four-state discrete Markov chain similar to one in [11] where each state represents one of the four available bit rates. The idea is based on the fact that although wireless channels are characterized by fast-fading, some correlation exists between the current transmission state and the immediate previous state. This means that, for instance, an 802.11b node able to transmit at 1 Mbps, will probably not be able to transmit at 11 Mbps after a period of time smaller or similar to the coherence time. The Markov chain is represented by a transition matrix *P*. This matrix is used by each node in the system to select the new transmission rate every time the coherence time has elapsed. Without loss of generality, the actual matrix *P* used in the scenario is:

$$P = \overbrace{\begin{bmatrix} 0.5 & 0.4 & 0.1 & 0 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.5 & 0.3 \\ 0 & 0.2 & 0.3 & 0.5 \end{bmatrix}}^{\text{future state}\ (1,\ 2,\ 5.5,\ 11)} \begin{matrix} 1 \\ 2 \\ 5.5 \\ 11 \end{matrix} \left.\vphantom{\begin{matrix} 1 \\ 2 \\ 5.5 \\ 11 \end{matrix}}\right\} \text{current state} \quad (1)$$

The rows in *P* represent the current state, i.e., the current available transmission rate index. Columns represent the future state (available transmission rate) once the coherence time is elapsed. The selected values for *P* are only an example, and they represent a relatively hostile channel in which nodes are more likely to transmit at the rates of 2 or 5 Mbps. Of course, any other values for *P* could be tested to model different channel scenarios.

B. System Parameters

A system with 20 mobile nodes which generate variable data traffic load has been considered. Data traffic generation is Poisson distributed and the size of the messages follows an exponential distribution with mean of $10 \cdot L_d$ bytes, where L_d is the size of the bytes per packet, transmitted in each DQCA frame or in each IEEE 802.11b data frame, respectively. In the simulations a packet size of L_d=2312 bytes has been considered. The data service does not accept any loss of packets but can tolerate reasonably large delays. An upper limit of 1 s to for data mean delay has been defined.

The MAC-specific parameters are summarized in Table 1. All control packets (FBP, RTS, CTS and ACK) are sent at the minimum rate of 1Mbps in order to ensure reliable transmission reliable transmission and backwards compatibility. As the ARS packets have no information bits [9], their size is therefore expressed in terms of transmission time. Large messages are fragmented to packets of size L_d and are transmitted in consecutive frames. The FBP consists of 2 bytes for the Frame Control (FC) field, 6 bytes for the feedback information, 1 byte for the ACK and 4 bytes for the FCS (Frame Control Sequence).

C. DQCA Configuration

Three different DQCA configurations have been considered by defining different VPFs:

1) DQCA: The VPF is defined as a constant value for all nodes. In this case the DTQ operation behaves as a FIFO scheduler, and therefore, no cross-layer mechanism is applied.
2) DQCA-VPF1: The VPF value for each node is defined as the value of its available transmission rate. An opportunistic scheduling scheme is obtained by using this VPF. Therefore, those nodes with better channel will attain access to the channel, thus maximizing the throughput of the network. However, this mechanism reduces fairness among nodes, since those with bad channel conditions may be forced to wait for a long period of time.
3) DQCA-VPF2: The VPF value for each node is defined as the ratio of the available bit rate of the node to its position in the DTQ. This scheme increases fairness as nodes with a bad channel are given a greater chance to transmit, if they have been queuing in the DTQ for a long time.

In the two latter cases, the FBP must include the available transmission rates of all the nodes in DTQ, required for the calculation of the respective VPF values. Since a set of four rates is supported by the considered PHY (802.11b), 2 bits suffice for their representation. This number of control bits should be adjusted accordingly for different PHY settings. As the number of nodes in the DTQ is expressed by the TQ value, an overhead of 2×TQ bits is required, rounded up to the closest number of bytes (this rounding up has been considered for feasibility reasons). It is worth mentioning that since this information might become stale as time goes by, it could be possible and convenient to periodically transmit some refresh channel state information frames to update the available rates of each node in the DTQ, as described in [12].

D. Simulation Results

A C++ based simulator has been developed for the evaluation of the proposed MAC mechanism. The performance in terms of throughput and mean delay versus the offered traffic load has been plotted for the three aforementioned DQCA configurations. Throughput is defined as the correctly received data bits per time unit and mean packet transmission delay is the average waiting time of the packets from their generation until their complete transmission. As a reference and to emphasize the enhancement achieved by DQCA, the performance metrics of 802.11b with the RTS/CTS mechanism enabled are also given.

Figure 4 depicts the throughput versus the offered load. For lighter traffic conditions the throughput increases linearly with the offered load, since all the generated packets are transmitted. At some point, the curves stabilize, indicating the maximum achievable throughput for each MAC algorithm. As a first observation, it is clear that the legacy 802.11 MAC has a fairly poor performance. Although the maximum PHY supported rate is 11 Mbps, the actual achieved throughput does not exceed 1.3 Mbps. This inefficiency is caused by the MAC layer overhead (including RTS, CTS, and ACK), the presence of data collisions, the channel hostility, and the time wasted due to backoff idle periods of time. On the other hand, when DQCA is applied, a significant enhancement is achieved, attributed to the reduced protocol overhead (i.e. ARS, FBP) and the almost collision-free data transmissions. For the most basic DQCA configuration the maximum achieved throughput value is above 2.5 Mbps.

The use of cross-layer-based scheduling further boosts the performance. Both proposed configurations (DQCA-VPF1, DQCA-VPF2) attain higher maximum throughput values with respect to both 802.11 and DQCA. In particular, DQCA-VPF1 provides the most significant throughput gain since at any time the node with the highest available rate is enabled to transmit. Consequently, the higher PHY-supported rates (5.5, 11 Mbps) are employed most of the time. DQCA-VPF2 throughput performance is bounded between DQCA-VPF1 and DQCA, thus achieving a compromise between efficiency (opportunistic scheduling) and fairness among nodes (FIFO transmission discipline).

The mean delay versus the offered load is illustrated in Figure 5. Again, 802.11b suffers from longer delays (no lower than 100 ms), mainly caused by the back-off mechanism. DQCA, with the parallel operation of the channel access and the data transmission mechanism manages to decrease the delay values at a scale of 50-60ms. The mean delay is even lower for the two cross-layer schemes, in which higher transmission rates are generally employed, thus reducing the transmission duration of data packets. Comparing the latter schemes, DQCA-VPF1 offers higher packet transmission delay compared to DQCA-VPF2 due to the longer periods of time that nodes with bad channel conditions have to wait to get access to the channel. However, since the saturation throughput of DQCA-VPF2 is higher, the delay remains at low values for larger traffic loads.

VI. Conclusions

The DQCA MAC protocol is presented in this paper as an innovative step towards the definition of novel MAC protocols for WLAN capable of dealing with the demanding and dynamic nature of new multimedia applications. By separating the resolution of collisions and the schedule of the data transmission into two distributed logical queues, throughput close to the maximum capacity is achieved, and what is more important, the maximum throughput is ensured regardless of the number of active nodes or the traffic load offered to the network. Both simulation and analytical results show improved performance compared to the current most extended standard for WLAN. In the light of actual implementation of the protocol, it is worth mentioning that DQCA could operate on top of any PHY layer. The possible backwards compatibility of the MAC protocol with the Standard IEEE 802.11 MAC protocol is being currently studied. In addition, current ongoing research is being focused on adapting the ideas of DQCA to distributed ad hoc networks and cooperative communications, as well as on supporting heterogeneous traffic with Quality of Service and exploiting the benefits of spatial diversity by using MIMO schemes.

ACKNOWLEDGEMENTS

This work has been partially funded by the Research Projects PlaNetS (FIT 330220-2005-111), LOOP (FIT-330215-2007-8), ETECLAS (TEC2005-07326-C02-01/TCM) and PERSEO (TEC2006-10459/TCM).

REFERENCES

[1] IEEE Std. 802.11 WG, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1999, standard.

[2] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", In *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 535-547, March 2000.

[3] G S. Shakkottai, T. S. Rappaport, P. C. Karlsson, "Cross-Layer Design for Wireless Networks", In *IEEE Communications Magazine*, Vol. 41, No. 10, Oct. 2003, pp.74 – 80.

[4] L. Alonso, R. Ferrús and R. Agustí, "WLAN Throughput Improvement via Distributed Queuing MAC", In *IEEE Communications Letters*, Vol. 9, No. 4, pp. 310 - 312, April 2005.

[5] W. Xu and G. Campbell, "A Near Perfect Stable Random Access Protocol for a Broadcast Channel", In *Proc. of the IEEE International Conference on Communications* 1992 (ICC'92), Vol. 1, pp. 370-374.

[6] C-T. Wu and G. Campbell, "Extended DQRAP (XDQRAP) A Cable TV Protocol Functioning as a Distributed Switch", In *Proc. of the IEEE 1st International Workshop on Community Networking Integrated Multimedia Services to The Home*, pp. 191-198, 1994.

[7] H-J. Lin and G. Campbell, "PDQRAP – Prioritized Distributed Queueing Random Access Protocol", In *Proc. of the IEEE Conference on Local Computer Networks*, Minneapolis, MN, USA, pp. 82-91, 1994.

[8] C-T. Wu and G. Campbell, "Interleaved DQRAP with Global TQ", Internal Report, Department of Computer Science, Illinois Institute of Technology, 1994.

[9] 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", In *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 9, pp. 1701-1718, September 2000.

[10] G. Campbell et al. “Method and apparatus for detecting collisions and controlling access to a communications channel”, US Patent no. US6408009 B1, June 2002.

[11] A. Konrad et al., “A Markov-Based Channel Model Algorithm for Wireless Networks”, In *Proc. 4th ACM Int'l. Workshop Modelling, Analysis and Simulation of Wireless and Mobile System*, Rome, Italy, pp. 28-36, July 2001.

[12] E. Kartsakli, A. Cateura, L. Alonso, J. Alonso-Zárate and Ch. V. Verikoukis, “Opportunistic Scheduling using and Enhanced Channel State Information Update Scheme for WLAN Systems with DQCA”, In Proc. of the IEEE Vehicular Technology Conference 2007, Dublin.

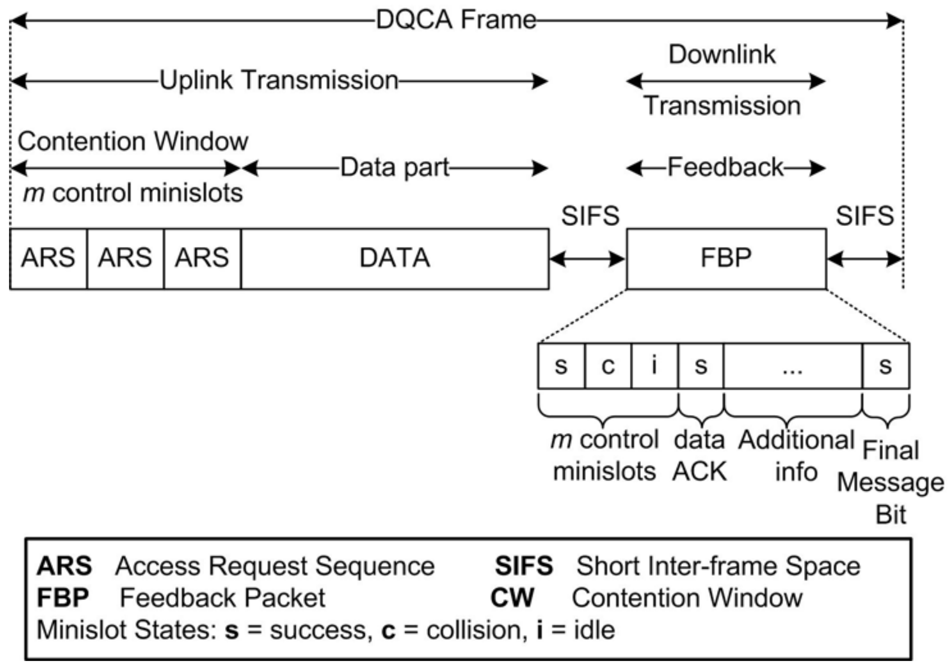

Figure 1 DQCA Frame Structure

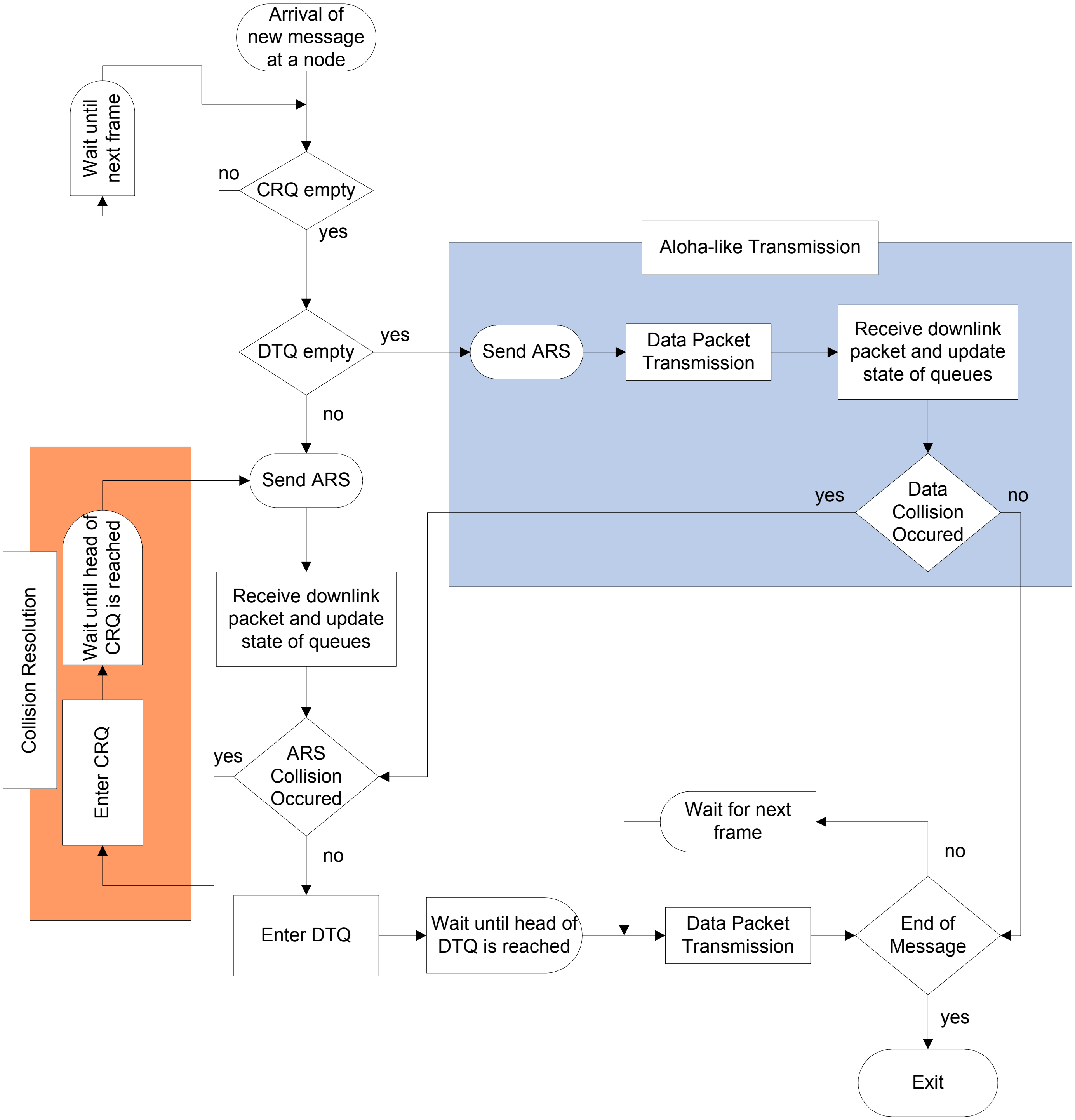

Figure 2 Flow Chart of the DQCA algorithm

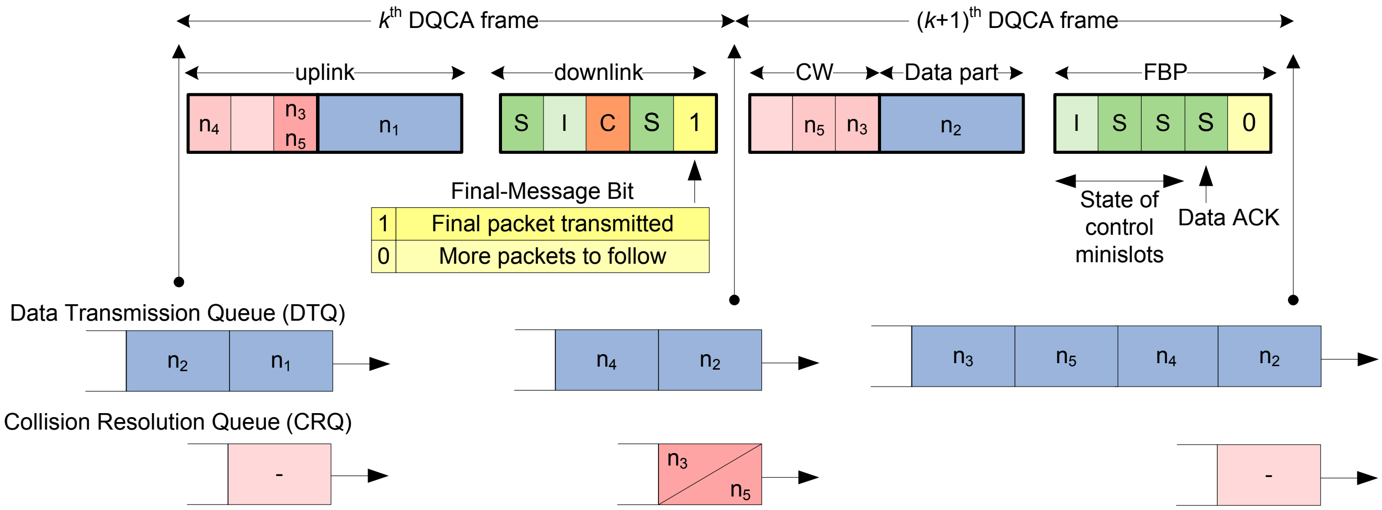

Figure 3 Example of the DQCA operation

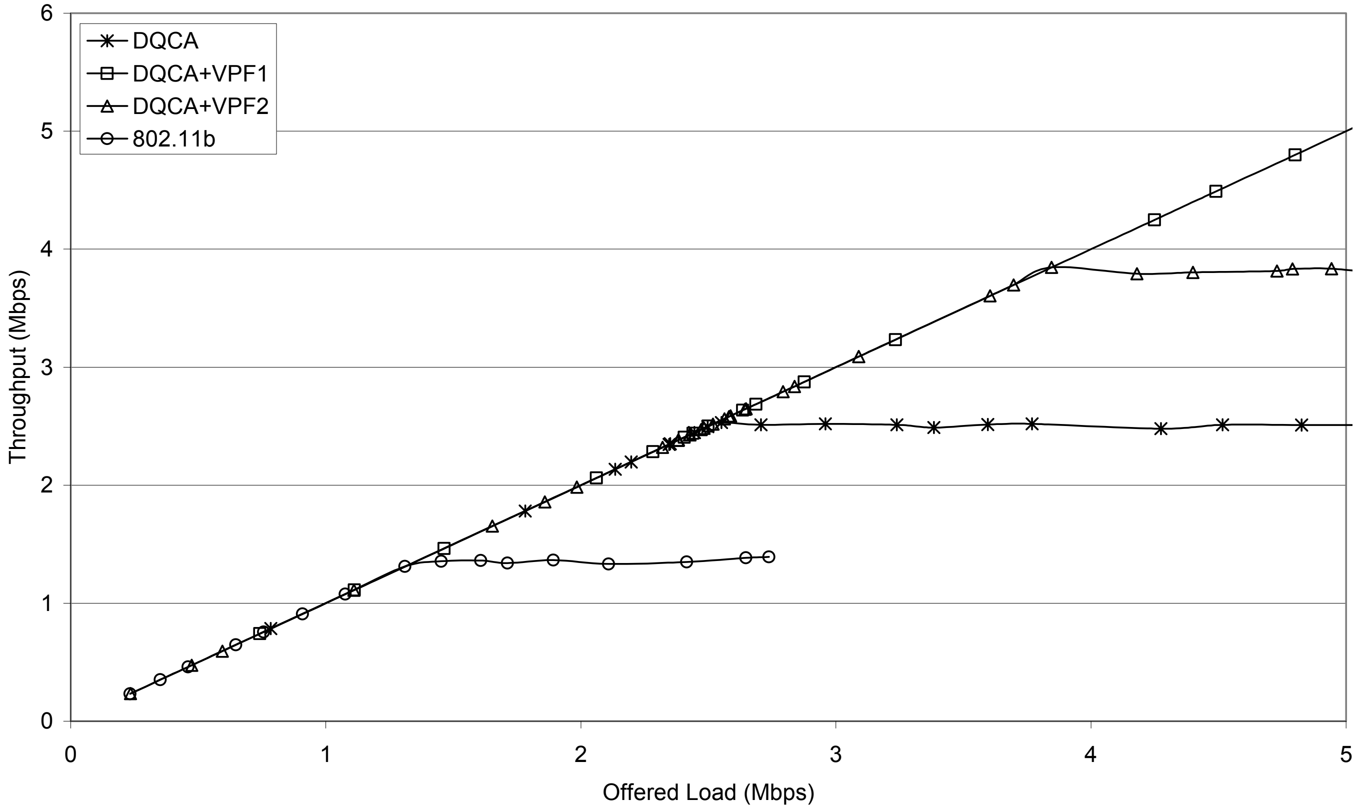

Figure 4 Throughput versus offered load

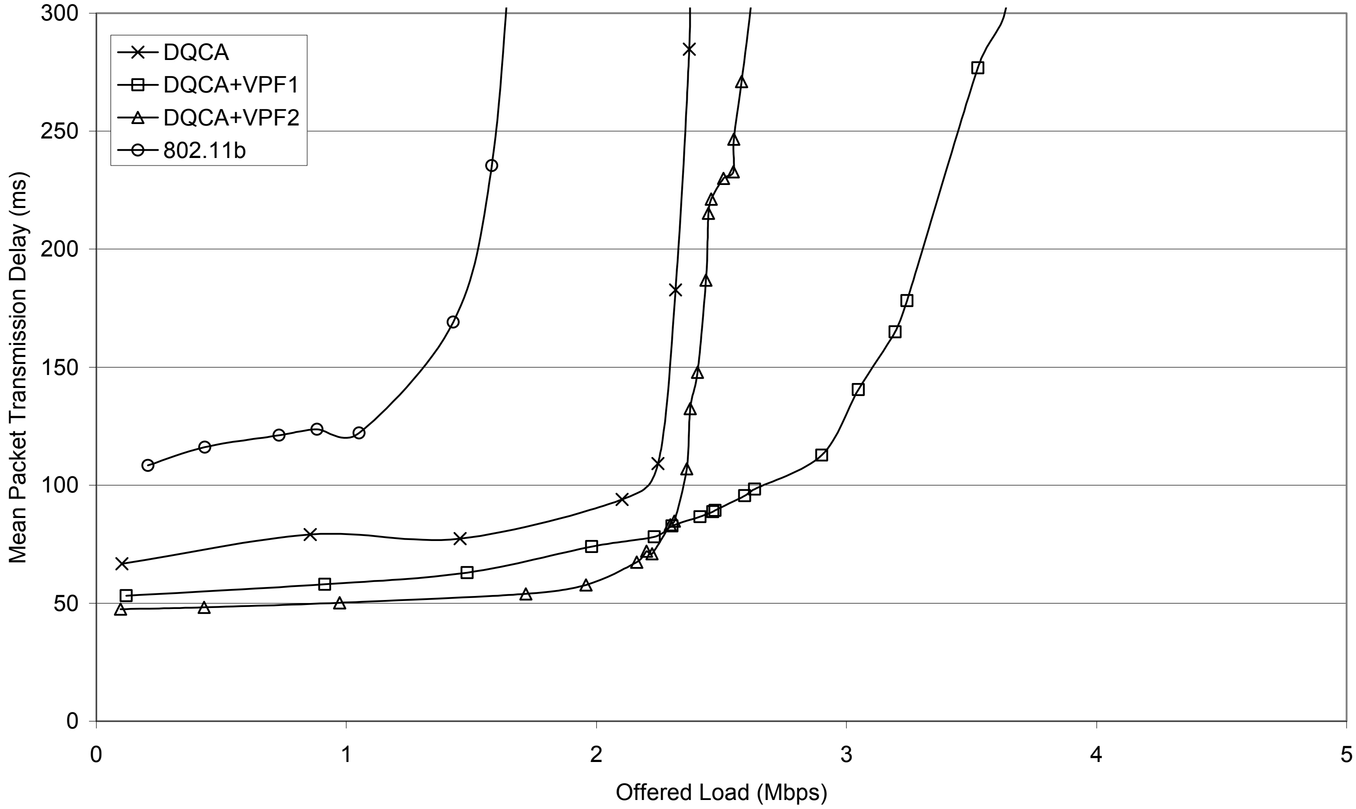

Figure 5 Mean Packet Transmission Delay versus offered load

Table 1. MAC layer parameters

Common MAC parameters	
MAC header length	34 bytes
PHY header duration	96 µs
SIFS	10 µs
Data Packet Size L_d	2312 bytes
IEEE 802.11b	
RTS length	20 bytes
CTS length	14 bytes
ACK length	14 bytes
SlotTime	20 µs
DIFS	50 µs
CWmin	31
CWmax	1023
DQCA	
Number of control slots m	3
ARS duration	10 µs
FBP length	13 bytes
FBP overhead for DQCA VPF1 and 2	(2×TQ) bits