

Analysis of the Distributed Queueing Algorithm in a Multichannel Network with Mixed-Type Traffic

Romeo Nibitanga^{1,*}, Elijah Mwangi² and Edward Ndung'u³

¹Pan African University, Kenya

²University of Nairobi, Kenya

³Jomo Kenyatta University of Agriculture and Technology, Kenya

Abstract

Networks such as LTE-M, NB-IoT, LoRa and SigFox are being deployed for massive IoT communications. However, the Aloha protocols used for the media access are inefficient when the network size is high. As a result, a Distributed Queueing (DQ) algorithm is being proposed to replace the conventional access schemes because of its superior network performance. In this paper, we study the algorithm with no prior consideration of any underlying technology at the physical layer. We perform a steady-state analysis of the algorithm in a network where the traffic is composed of periodic and urgent packets. It was found that for high traffic from the network all the nodes were taking part in each contention. Nevertheless, for better performance in terms of access delay and packet drop rate, it was preferable to operate the algorithm in the small traffic interval. Moreover, the downlink traffic has been observed to have a significant impact on the stability of the algorithm.

Keywords: Aloha, contention resolution, distributed queueing, massive communication, medium access control, wireless sensor network.

Received on 17 January 2021, accepted on 15 February 2021, published on 07 April 2021

Copyright © 2021 Romeo Nibitanga *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [Creative Commons Attribution license](#), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.7-4-2021.169176

1. Introduction

With the Internet of Things (IoT), the communication networks are expected to be dominated mostly by the traffic from thousands of connected devices [1]. Low Power Wide Area Networks (LPWAN) are one of the networks suggested for the support of the massive number of connections that will be generated by those devices [2], [3]. Most of the emerging solutions proposed for IoT networks such as the Long Term Evolution for Machine (LTE-M), Narrow Band Internet of Things (NB-IoT), LoRaWAN (Long Range Wide Area Networks), and SigFox share in common the use of the Aloha-based algorithms for channel access. Those protocols are easy to implement in IoT devices. However, as many studies [4]-[7] have pointed out, the Aloha-based algorithms are inefficient in terms of delay access, energy consumption and channel throughput when the number of nodes in the network is high. Therefore, there is a need for a new channel access

mechanism capable of handling the massive number of sessions between IoT devices.

Currently, the Distributed Queueing (DQ) algorithm is being proposed to replace the Aloha-based protocols used in most of the conventional IoT networks. In [8], it is even predicted as a solution to most channel access issues found in IoT networks. The basic principle of the DQ algorithm is to divide an initial number of colliding devices into smaller groups before trying to resolve another collision. Therefore, contending nodes are placed in a Collision Resolution Queue (CRQ) as long as they did not finish their contention. Otherwise, they are sent in a Data Transmission Queue (DTQ) when they have been granted access to the channel and they are waiting for data transmission. Several works [5]-[7], [9]-[11] analyzed the DQ algorithm and have found that it outperforms Aloha-based algorithms in terms of several network performance metrics.

However, most of the works evaluated the DQ algorithm under a specific physical technology, making the results only relevant to that technology. Moreover, only a given number

*Corresponding author. Email: nibitanga.romeo@students.jkuat.ac.ke

of contention slots is considered or only some performance metrics such as the access delay, the energy consumption and the throughput are of interest. Furthermore, the DQ algorithm is considered stable independent of the number of nodes contending in the network, which may not be the case.

In general, the IoT traffic is expected to be heterogeneous [12]-[17]. Some applications are characterized by packets that are generated at a regular interval of time, whereas for others they are dominantly triggered by external events. However, most of the time, both types of the packet characterize most of the IoT applications. In this paper, every node in the network can generate both types of the traffic in the form of periodic and urgent packets. Therefore, the total traffic load in an IoT network is a mix of many sources with different periods and rates. The superposition of traffic from the nodes is aggregated into a Poisson arrival process. Such an approximation has been found valid when the number of nodes in the network is sufficiently large and when the traffic is not purely homogeneous [12].

Briefly, there is a need of an independent analysis of the DQ algorithm without any prior consideration of any physical layer or a given number of contention slots. Such an analysis should include a traffic model that considers the real IoT data to allow the foundation for any further introduction of quality of service in the DQ system. That is why we propose in this paper an extended evaluation of the DQ algorithm that would serve as a basic model for any further analysis of the DQ algorithm in IoT setting.

The contribution of this paper lies in the fact that we are evaluating the DQ algorithm in a new network environment considering a traffic from multiple nodes from various applications in the network. Besides, new performance metrics not considered in previous works and their analytical models are also introduced. Those metrics are the number of packets contending per session, the packet drop rate, and the average distribution of contention slots. The impact of the downlink traffic on the stability of the DQ algorithm is also presented. Moreover, different traffic load intervals from the network given the number of the contention slots were also defined. As IoT nodes are expected to be deployed in networks with a large number of nodes, we also consider a network with multiple channels. Depending on the technology at the physical layer, those channels may be available in the form of sub-carriers or spreading factors. In addition, we present the conditions under which the DQ algorithm is stable in such environments.

The rest of the paper is organized as follows. In section II, a literature review on works related to the DQ algorithm is given. The system model together with a brief description of the DQ algorithm and its modification to a multichannel network are presented in Section III. The performance evaluation of the DQ algorithm in a multichannel network is given in Section IV. Section V concludes and gives recommendations for future extension of the work.

2. Related works

At its introduction as the Distributed Queuing Algorithm Random Access Protocol (DQRAP) in [18] by Xu and Campbell, the DQ algorithm was used in the cable TV

channel for the transmission of digital data. Then, it was adapted to various systems such as the Prioritized Distributed Queuing Random Access Protocol (PDQRAP) [19] for priority-based traffic and the Extended Distributed Queuing Random Access Protocol (XDQRAP) [20] for the support of packets with variable length. Before the IoT era, the DQ algorithm was also proposed for several types of networks such as wireless area networks [21], cellular third-generation networks [22], body area networks [23], mobile ad hoc networks [24] and cooperative networks [25].

In [18], [26], an evaluation of the stability condition for the DQ algorithm in a single channel network with a Poisson distributed input traffic is presented. The CRQ was modelled using the Markov chain theory, whereas the DTQ was modelled as G/D/1 queue. The authors found that the stability of the DQ system was only defined by the DTQ. As for the CRQ, it was stable for any input rate.

Since recently, the DQ algorithm has been suggested as a solution to several of the challenges encountered in IoT networks in both the licensed and unlicensed frequency spectrum.

In [27], the authors found the DQ algorithm to reduce the energy consumption compared to the Frame Slotted Aloha (FSA) protocol in networks with 802.15.4 devices. Later, they proposed two energy-efficient DQ-based protocols: these are the Low Power Distributed Queuing (LPDQ) [28] and the Energy Harvesting Distributed queuing (EHDQ) [4]. The first protocol was developed for low-power wireless networks with bursty traffic, whereas the second protocol was designed for data collection networks with devices equipped with energy harvester. Both protocols were evaluated in Radio Frequency Identification (RFID) networks.

Several works evaluated the DQ algorithm in LTE networks [5], [9], [10], [29]-[32]. In [5], the authors proposed the DQ algorithm as a replacement to the Aloha-based protocols used in LTE networks for the support of the massive number of devices in the network. The results showed that applying the DQ algorithm in the media access layer of the LTE networks would achieve, under specific network configuration, a reduction of 85% and 40% with respect to the delay and the energy consumption.

The Distributed Queuing Algorithm Access for LTE (DQAL) [29] allows different contending groups with a collision to re-transmit their preambles in the same access slot. Here again, the algorithm was found superior to the Extended Barring Access (EAB) conventionally used in LTE networks in terms of delay. In [30], a DQ algorithm that resolves the contention groups from an initial collision in parallel was found to decrease the average number of access compared to the DQ sequential method.

To reduce the significant access delay introduced by the conventional DQ algorithm in LTE networks, the authors in [9] presented the Free Access Distributed Protocol (FADQ). It divides the colliding devices into smaller groups where the DQ algorithm is applied. Besides, each new node may contend in the next random access opportunity without waiting for the collision to end. Moreover, the same authors in [10] gave a priority-based DQ algorithm that uses the length of the collision resolution queue to get information about the channel congestion. Another DQ-based algorithm for LTE networks that shares the concept of dividing the

colliding devices into smaller groups is also presented in [31]. However, the authors proposed that newly arrived devices could only contend in the groups without collisions.

The DQ algorithm is also being proposed as a candidate at the media access level in low-power wide area networks [6], [7], [33]. In [6], the authors presented the Resource Grouping Distributed Queuing algorithm (RGDQ) to resolve the massive connectivity issue present in Narrow Band Internet of Things (NB-IoT) networks. Aware of the inherent significant delay caused by the conventional DQ scheme, the devices in the network are divided into groups depending on their coverage area. Then, the DQ algorithm is applied to each group. The sub-carriers are shared evenly between groups. In [7], the DQ-Lora, where the Distributed Queuing protocol was used at the media access layer and the LoRa technology was applied at the physical layer, is proposed for the LoRa networks as a replacement of the pure Aloha conventionally used. The DQ-LoRa was found to outperform the pure Aloha in terms of energy consumption, access delay and channel throughput. Lastly, a DQ algorithm with multiple data slots DQ-N was also proposed for low-power crowd-sourced networks in [33]. Unlike the conventional DQ algorithm, DQ-N divides the contention-free period into multiple data slots instead of one to support a massive number of nodes with a low data rate.

The DQ algorithm has also been used together with the Massive Input Massive Output (MIMO) technology. In [11], the authors proposed the Distributed Queuing Random Access Massive Input Massive Output (DQRA-MIMO) system to get higher throughput under delay constraint and limited time-frequency resources.

In general, in most of the reviewed works, we have noticed that the performance evaluation of the DQ algorithm was done under a given physical layer, making the results only valid for that specific layer. Besides the utilization of the contention period is systematically missed in all the research investigations. It has also been noted that most of the works, although claiming to be IoT-oriented, were only considering one type of the traffic in the network. Periodic packets and the performance metrics related to them are ignored. Furthermore, the performance analysis of the the algorithm was found to be focused on the CRQ, whereas the analysis of the other queue, namely the DTQ, is absent. It has also been found that the DQ algorithm was considered to be stable for any number of nodes in the network or no information was given on how the packets are processed after leaving the CRQ. The research investigation presented in this paper is an attempt to bridge the above research gaps.

Finally, in our previous work [34], a performance evaluation of the DQ algorithm in a single-channel network was conducted. It was shown that the DQ algorithm, when it was stable, did not use efficiently the contention slot. In addition, the algorithm was found to outperform Aloha-based protocols only in idle-to-saturation scenario. In contrast to the evaluation performed in [34], in this paper, we propose an analysis of the DQ algorithm in a multichannel network instead of a single-channel setting. Moreover, a statistical analysis of the DQ algorithm in a single channel network is also proposed in our work presented in [35]. However, in this

paper, the steady-state values of the metrics in network with nodes from various applications are of interest, whereas in [35] only one contention is considered.

3. System model and algorithm description

The system model is a star network topology comprising of a base station and n nodes from N applications. It is assumed that all the nodes in the network can communicate with the base station independently of their location. It is also assumed that the propagation delay is negligible compared to the frame duration within the cell. Therefore, the distance between the nodes and the base station does not influence the performance metrics, specifically for the nodes that are at the edge of the cell.

A node can belong to only one application. Therefore, all the nodes from the same application have similar traffic characteristics: periodic packets are generated at regular time interval T_i and urgent packets are generated following a Poisson arrival process with a parameter λ_i with $i = 1, 2, \dots, N$. For a given j th node (with $j = 1, 2, \dots, n$) from an i th application, it is assumed that urgent packets have a high priority for contention over the periodic ones.

The nodes contend for access to the wireless channel following the rules of the DQ algorithm. The DQ protocol is a tree-splitting algorithm used for channel access. It divides contending nodes into two different subsystems: the collision resolution subsystem (CRS) and the data transmission subsystem (DTS). The CRS contains groups of nodes that are contending for the channel access, and the DTS is formed of nodes that have secured a position in the DTQ and are waiting their turn to transmit their data. In a DQ multichannel setting, the first subsystem is formed of one queue used for collision resolution, whereas the second subsystem is composed of a number c (with $c = 1, 2, \dots$) of queues (channels) used for data transmission.

A DQ frame is divided into three parts as shown in Figure 1: the contention slots used by the nodes to send the access request signals, the data slots used to send the packets from both the nodes and the base station and, lastly, the feedback slots used to send DQ related information to the nodes by the base station. A detailed description of the classical DQ algorithm can be found in [18].

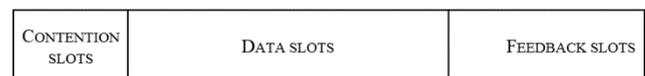


Figure 1. DQ frame structure

After leaving the CRQ, the packets are allocated in the DTQ channels randomly by the base station. It is assumed that after each DQ frame, the feedback slots are used to transmit information about the DTQ channels allocated to successful contending packets. A successful contention slot is mapped to a given DTQ channel by the gateway. Therefore, a node can

identify which DTQ channel has been allocated to its packet. A collision in the contention slots is handled following the classical rules of the DQ algorithm. Despite leading to a larger signalling frame compared to a single-channel DQ setting, the gateway allocation mechanism decreases the number of signalling messages that would be required if nodes were to choose individually the DTQ channels for data transmission.

Besides other DQ control and signalling information, the feedback message contains:

- (i) RQ , an integer that represents the length of the CRQ queue;
- (ii) TQ_i , $i = 1, 2, \dots, c$, each integer represents the length of each DTQ channel, respectively.

Together with the integers RQ and TQ_i , the nodes have to keep three other integers:

- (i) pRQ , the position of the contending packet (either periodic or urgent) in the CRQ;
- (ii) pTQ_p , the position of the periodic packet in the allocated DTQ channel;
- (iii) pTQ_u , the position of the urgent packet in the allocated DTQ channel.

The integers pRQ , pTQ_p and pTQ_u are computed by each node following the DQ algorithm queueing rules as described in [18]. The only difference to the classical DQ algorithm is that pTQ_p and pTQ_u are computed separately for each DTQ allocated channel.

At the beginning of the contention process, the CRQ and the DTQs are empty. Besides, there are no nodes scheduled for channel access or data transmission. We only consider the traffic in the contention slots and the data slots our analysis, as the feedback slots are reserved only for the base station and are contention-free. The nodes have a perfect DQ frame and contention slot synchronization. Thus, they can contend either in a synchronized or unsynchronized manner per application. In the synchronized scenario, the nodes from the same application are scheduled to contend during the same DQ frame at a regular time interval. However, in a massive communication network, a synchronized scenario is difficult to achieve across many nodes and could lead to a complex node. Therefore, we assume that the nodes are contending asynchronously.

On its incorporation into the network, a node from an i th application chooses randomly a DQ frame j for $j = 1, 2, \dots, T_i$ and regularly generates periodic packets at $j + T_i.k$ DQ frames for $k = 0, 1, 2, \dots$ Moreover, each packet is assumed to have a timeout period, after which it is dropped. However, an urgent packet can also be dropped if a new packet is generated before the former has not entered the CRQ. The timeout period is defined by the type of the packet. We assume that a node requires only one packet for its data transmission. It should also be noted that (i) new arrival nodes are not allowed to enter the CRQ before it is emptied, and (ii) a node can only contend for channel access for one type of the packet simultaneously.

In the long run, following the fundamental Palm-Khintchine theorem [12], the aggregated traffic generated from n nodes in a network with N applications can be

modelled as a Poisson arrival process with an overall parameter λ_{tot} .

Theorem 1 (Palm-Khintchine Theorem). Let us consider n independent renewal processes with identically and independent distributed inter-arrival times T_j with $j = 1, 2, \dots, n$. The expected inter-arrival time for each renewal process is $E[T_j] = 1/\lambda_j$ where λ_j is the arrival intensity. Then the superposition is asymptotically a Poisson process for $n \rightarrow \infty$, if the following assumptions hold:

- (i) The intensity λ of the superposition process is finite, $\sum_{j=1}^n \lambda_j = \lambda < \infty$ when $n \rightarrow \infty$
- (ii) No single process dominates the superposition process, $\lambda_j \ll \lambda/n; \forall j$.

Let assume that the application time periods are $T_1 < T_2 < \dots < T_{N-1} < T_N$. Therefore, in the long run for a time interval T_N , the total traffic load λ_{tot} from n nodes in a network with N applications is defined by the traffic from both the periodic and urgent packets generated during that interval of time:

$$\lambda_{tot} = \sum_{j=1}^n \frac{T_N}{T_j} + \sum_{j=1}^n \lambda_j \quad (1)$$

where T_j and λ_j are the application period and the rate parameter for the j th node respectively. For all $j = 1, 2, \dots, n$, we have $T_j = T_i$ and $\lambda_j = \lambda_i$ if a node j belongs to an application i .

The Eq. (1) is also equivalent to

$$\lambda_{tot} = \sum_{i=1}^N v_i n_i (1 + \lambda_i T_i) \quad (2)$$

where $v_i = T_N/T_i$ are constants, T_i is the i th application period, n_i is the number of nodes in the network from the i th application and λ_i is the i th application rate parameter.

Let us define n_x as the average number of packets contending at the same time in a DQ frame during the period T_N , i.e., the average traffic load. The probability that a node chooses a given DQ frame for its periodic or urgent packet during the period T_N follows a discrete uniform distribution with the probability mass function $p_n = 1/T_N$.

The probability of having x nodes contending in a DQ frame follows a binomial distribution with parameters p_n and λ_{tot} :

$$p_x(X = x) = \binom{\lambda_{tot}}{x} p_n^x q_n^{\lambda_{tot}-x} \quad (3)$$

where X is the corresponding random variable and $q_n = 1 - p_n$.

The average number of packets n_x contending at the same time in a DQ frame during the period T_N in the long run is:

$$n_x = \sum_{x=0}^{\lambda_{tot}} x p_x = \frac{\lambda_{tot}}{T_N} \quad (4)$$

To validate our analytical results, a steady-state discrete-event simulation model of the DQ algorithm as described above has been developed and executed in Matlab for the evaluation of the performance metrics of the DQ algorithm. The sample averages of the considered metrics are obtained over a single replication. We use the method of batch means for the estimation of the throughput because it is less susceptible to the initial effects of the simulation [36], [37]. The batch size corresponds to the smallest application period. Furthermore, we considered a number of 1500 batches to guarantee the independence between them. As for the other metrics, we consider each contention as a terminating simulation. Therefore, the estimate of the mean is averaged

over the number of contentions occurring during the observation time. Therefore, an arbitrary network with nodes from three applications is considered. It is assumed that the nodes are distributed equally among the applications. The application periods in frames are: $T_1 = 100$, $T_2 = 200$ and $T_3 = 300$.

4. Results and discussion

4.1. Average number of packets contending per CRQ session

Let n_{crq} be the steady-state average number of packets (both periodic and urgent) contending during each CRQ session. It is defined by the average number n_x of packets generated per DQ frame and the average collision resolution time t_{crq} . The average collision resolution time t_{crq} was observed, through numerical simulations, to vary with the number of packets n contending at the beginning of a CRQ session as follows:

$$t_{crq} = n c_{crq} \quad (5)$$

where c_{crq} is a constant that depends on the number of contention slots m in the DQ frame, and it follows a normal distribution with parameters μ and σ^2 as presented in Table 1. In general, the constant c_{crq} has been found to be equal to $1/\log(m)$ and it confirms the results from [38]. It should be noted that the metric t_{crq} is measured in frames. The average collision resolution time t_{crq} was observed to decrease logarithmically with m and to grow with the number of packets n contending at the beginning of the session.

Table 1. Constant c_{crq} and the parameters $n_x^{(l)}$ and $n_x^{(u)}$

Number of contention slots	c_{crq}		$n_x^{(l)}$	$n_x^{(u)}$
	μ	σ^2		
$m = 3$	0.906	2.89×10^{-4}	1.10	2.20
$m = 4$	0.719	1.69×10^{-4}	1.39	2.78
$m = 8$	0.476	9.80×10^{-5}	2.10	4.20
$m = 12$	0.410	7.06×10^{-5}	2.44	4.88
$m = 16$	0.353	3.76×10^{-4}	2.83	5.61

From (5), the collision resolution time for the j th CRQ session can be defined as follows:

$$t_{crq}(j) = n_{crq}(j-1) c_{crq} \quad (6)$$

where $n_{crq}(j-1)$ is the average number of packets generated during the $(j-1)$ th CRQ session for $j = 1, 2, \dots$ and we can consider that $n_{crq}(0) = n_x$.

The average number of packets generated during the j th collision resolution time from an i th application is defined by the application period T_i for periodic packets and by the application rate parameter λ_i for urgent packets. Let us define $n_{pcrq}^{(i)}(j)$ and $n_{ucrq}^{(i)}(j)$ as the average number of periodic and

urgent packets generated during the time interval $t_{crq}(j)$ from an i th application, respectively. Therefore, we have:

$$\begin{cases} n_{pcrq}^{(i)}(j) = t_{crq}(j) \frac{n_i}{T_i} \\ n_{ucrq}^{(i)}(j) = t_{crq}(j) n_i \lambda_i \end{cases} \quad (7)$$

where n_i is the number of nodes in the network from an i th application.

Let $p_u^{(i)}[U(j)]$ be the probability that u (with $u = 1, 2, \dots$) urgent packets from an i th application are generated during the j th collision resolution time:

$$p_u^{(i)}[U(j) = u] = \frac{[\lambda_i t_{crq}(j)]^u e^{-\lambda_i t_{crq}(j)}}{u!} \quad (7)$$

where U is the corresponding random variable.

A periodic packet from an i th application takes part in a contention if no urgent packet is generated during the waiting time before the contention, i.e. $p_u^{(i)}[U(j) = 0]$. However, an urgent packet will participate in a contention:

- (i) if no other urgent packet is generated during the waiting time before contention, i.e. $p_u^{(i)}[U(j) \leq 1]$ or;
- (ii) if the waiting time before the contention is not greater than its application timeout.

Let $p_w^{(i)}[W(j) \leq 1/\lambda_i]$ be the probability for an urgent packet from an i th application to wait before the contention an interval of time not greater than the average inter-arrival time $1/\lambda_i$ during the collision resolution time $t_{crq}(j)$. The waiting time before the contention for an urgent packet is distributed following a uniform discrete distribution $U \sim (0, t_{crq}[j])$. Therefore, we have:

$$p_w^{(i)}[W(j) \leq \frac{1}{\lambda_i}] = \sum_{i=0}^{\lambda_i} \frac{1}{t_{crq}(j)+1} \quad (9)$$

Consequently, the probability for an urgent packet from an i th application to take part in a contention during the j th collision resolution time is:

$$p_u^{(i)}(j) = p_u^{(i)}[U(j) \leq 1] + p_w^{(i)}[W(j) \leq \frac{1}{\lambda_i}] - p_u^{(i)}[U(j) \leq 1] p_w^{(i)}[W(j) \leq \frac{1}{\lambda_i}] \quad (10)$$

The average number of packets generated during the j th collision resolution time from an i th application that may contend in the next CRQ session can be defined as follows:

$$n_{crq}^{(i)}(j) = n_{pcrq}^{(i)}(j) p_u^{(i)}[U(j) = 0] + n_{ucrq}^{(i)}(j) p_u^{(i)}(j) \quad (11)$$

The number of packets generated during the $(j-1)$ th collision resolution time corresponds to the number of packets that are scheduled to contend during the j th contention CRQ session. It is assumed that a node can only contend for one type of packet during a CRQ session. Thus, the average number of packets contending per j th CRQ session from an i th application cannot be greater than the number of nodes from that application present in the network:

$$n_{crq}^{(i)}(j) \leq n_i \quad (12)$$

Therefore, the average number of contending packets (both periodic and urgent) per CRQ session is:

$$n_{crq} = \frac{1}{j} \sum_{j=1}^{\infty} \sum_{i=1}^N n_{crq}^{(i)}(j) \quad (13)$$

In Figure 2, we present both the analytical and numerical results of the variation of the average number of packets contending per CRQ session n_{crq} over the average traffic n_x for an arbitrary network with nodes from three applications. From Figure 2, it can be observed that three intervals of traffic from the network can be distinguished for each value of m . For $n_x \leq n_x^{(l)}$, we have $n_{crq} \leq n_x$ because all the contending packets during a session resolve their contention in the same DQ frame as they were generated. The parameters $n_x^{(l)}$ and $n_x^{(u)}$ corresponds to the average number of packets generated during a DQ frame n_x for which $t_{crq} = 1$ and $t_{crq} = 2$ respectively as presented in Table 1. However, for $n_x > n_x^{(u)}$, all the nodes in the network are taking part in each contention due to significant collision resolution time. Thus, we have $n_{crq} = n$. As for $n_x^{(l)} < n_x \leq n_x^{(u)}$, the number of packets contending per CRQ session varies between a value corresponding to the network traffic and the total number of nodes in the network, i.e., $n_x \leq n_{crq} \leq n$.

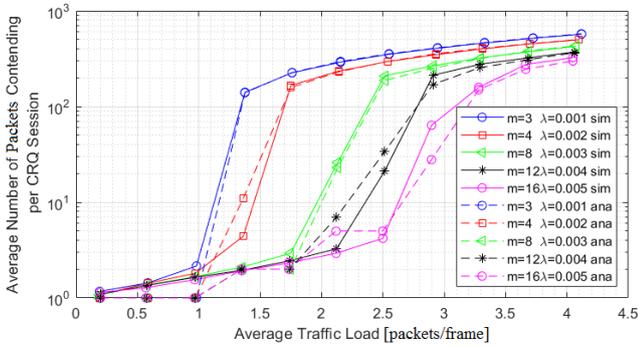


Figure 2. Average number of packets contending per CRQ session

Let us consider the three intervals to be respectively the small, the medium and the high traffic intervals from the network. From Table 1, it can be noticed that the lengths of the intervals increase logarithmically with the number of contention slots.

It should be noted that we use the abbreviations *ana* and *sim* for analytical and simulation results in all the figures, respectively.

4.2. Average access delay in the CRQ per packet

Let $t_{w/packet}$ and $t_{crq/packet}$ be the average waiting time before entry into the CRQ for each packet and the average collision resolution time per packet, respectively. Both metrics are measured in frames. It is assumed that because a node can only take part in a CRQ session for one type of the packet at a time, the urgent packets have a high priority over the periodic ones.

Once a packet has entered the CRQ, the time spent in the queue is independent of its type because packets are treated equally during the collision resolution. In the long run,

depending on the network traffic load, these cases can be observed:

- (i) if $n_x \leq n_x^{(l)}$, then any contending packet leaves the CRQ in the same DQ frame as it was generated. The small traffic load from the network guarantees instantaneous participation in the CRQ and an immediate contention resolution for all contending packets.
- (ii) if $n_x > n_x^{(l)}$, then we have:
 - a) if $t_{crq} < T_i$, then the waiting time before the contention for urgent packets follows a discrete uniform distribution $U \sim (0, t_{crq})$ because the urgent packets have a high priority over the periodic ones. The waiting time for periodic packets is defined by the average collision resolution time t_{crq} , the application period T_i and the application rate parameter λ_i . Let us define $p_p^{(i)}$ and $p_u^{(i)}$ as the probabilities of having respectively a periodic packet and an urgent packet from an i th application during a DQ frame. From (1), the total traffic load from an i th application during the time interval T_N is:

$$\lambda_{tot}^{(i)} = v_i n_p(i) (1 + \lambda_i T_i) \quad (14)$$

which leads to:

$$\begin{cases} p_p^{(i)} = \frac{1}{1 + \lambda_i T_i} \\ p_u^{(i)} = \lambda_i \frac{T_i}{1 + \lambda_i T_i} \end{cases} \quad (15)$$

Thus, the average waiting time per packet before the contention can be defined as follows:

$$t_{w/packet} = \begin{cases} \frac{t_{crq} p_p^{(i)} + T_i p_u^{(i)}}{2} & (*) \\ \frac{t_{crq}}{2} & (**) \end{cases} \quad (16)$$

where the first expression (*) is for periodic packets and the second expression (**) is for urgent packets.

- b) if $t_{crq} \geq T_i$, then the waiting times before the contention for periodic and urgent packets from an i th application follow a discrete uniform distribution $U \sim (a, b)$ with $a = 0$ and $b = T_i$ for the former and with $a = 0$ and $b = t_{crq}$ for the latter. Therefore, the average waiting time before contention per packet is:

$$t_{w/packet} = \begin{cases} \frac{T_i}{2} & \text{for periodic packets} \\ \frac{t_{crq}}{2} & \text{for urgent packets} \end{cases} \quad (17)$$

For $n_x > n_x^{(l)}$, the average collision resolution time $t_{crq/packet}$ for both types of the packet is the same because once the contention process has begun, no type of packet is preferred over the other. The average waiting time per packet in the CRQ was observed through numerical simulations to increase on average with the number n_{crq} of packets contending during a CRQ session for a given number of contention slots. However, it decreases with the number of

contention slots. The average collision resolution time per packet $t_{crq/packet}$ can be defined as follows:

$$t_{crq/packet} = n_{crq} c_{crq/packet} \quad (18)$$

where $c_{crq/packet}$ is a constant depending on the number of contention slots m in the DQ frame. It follows a normal distribution $N \sim (\mu, \sigma^2)$ as presented in Table 2.

Table 2. Constant $c_{crq/packet}$

Number of contention slots	$c_{crq/packet}$	
	μ	σ^2
$m = 3$	0.522	9.22×10^{-5}
$m = 4$	0.393	6.24×10^{-5}
$m = 8$	0.228	8.46×10^{-5}
$m = 12$	0.188	2.16×10^{-4}
$m = 16$	0.148	2.16×10^{-4}

We present the average waiting time per packet before the collision resolution $t_{w/frame}$ in Figure 3(a) for periodic packets in a network with three applications and in Figure 3(b) for urgent packets for various networks with different scenarios. In addition, the average collision resolution time $t_{crq/packet}$ over the network traffic n_x is given in Figure 3(c). It is assumed that urgent packets from different applications have the same application rates in Figure 3(b). Therefore, the results are identical for all the applications.

From Figure 3, it can be observed that for both metrics $t_{w/packet}$ and $t_{crq/packet}$ when the traffic from the network is small, i.e., $n_x \leq n_x^{(l)}$, a node has immediate access to the channel. However, as the network traffic increases to the medium-to-high interval, both the waiting time before entry in the CRQ and the time spent in the CRQ are defined by the overall collision resolution time. The first metric is limited by the timeout period defined by the type of the packet, whereas the second metric increases linearly with the traffic load. Therefore, in terms of access delay, it would be preferable to operate the DQ algorithm in the small traffic interval to keep the values of $t_{w/packet}$ and $t_{crq/packet}$ at their minimum.

4.3. Average number of random access attempts per packet

During the collision resolution time, each node with a packet ready to be sent tries to access the wireless channel. It sends request signals in the chosen contention slot every time its CRQ group is scheduled to contend. That process is repeated until the packet leaves the CRQ.

Let $RA_{attempts}$ be the average number of random access attempts required by a node to access the wireless channel. It is the same for both periodic and urgent packets because they are both treated with equal priority once they have entered a collision resolution process. The average number of attempts per packet is defined by the average number n_{crq} of packets taking part in a CRQ session. Thus, we have:

- (i) if $n_x \leq n_x^{(l)}$, then all the packets contending during the CRQ session have access to the channel immediately after they are generated, meaning that only one attempt is required for any node to leave the CRQ:

$$RA_{attempts} = 1$$

(19)

- (ii) if $n_x > n_x^{(l)}$, then the average number $RA_{attempts}$ of attempts per packet has been found to increase with the average number n_{crq} of packets contending during a CRQ session as presented in Figure 4.

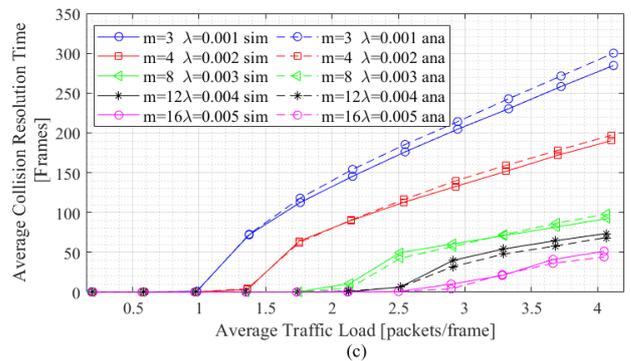
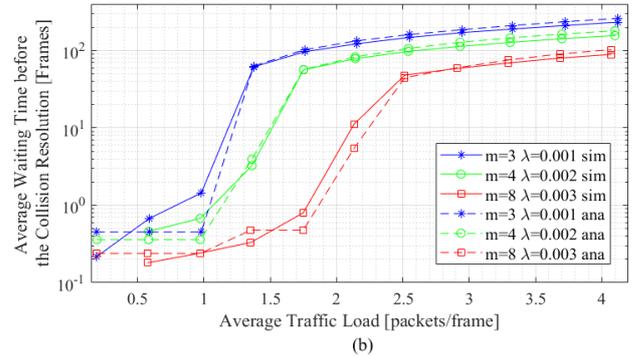
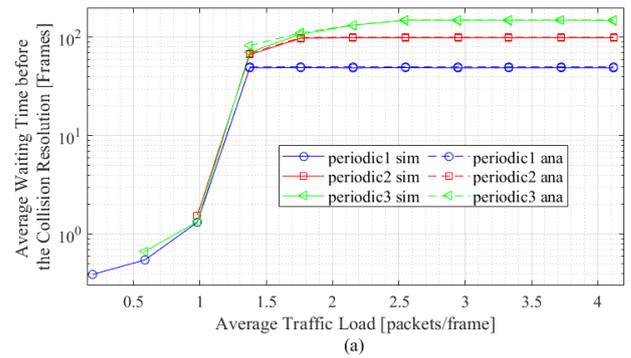


Figure 3. Average waiting time before entry in the CRQ (a) for periodic packets for a network with three applications and $m=3$ (b) for urgent packets for different scenarios. (c) Average collision resolution time per packet

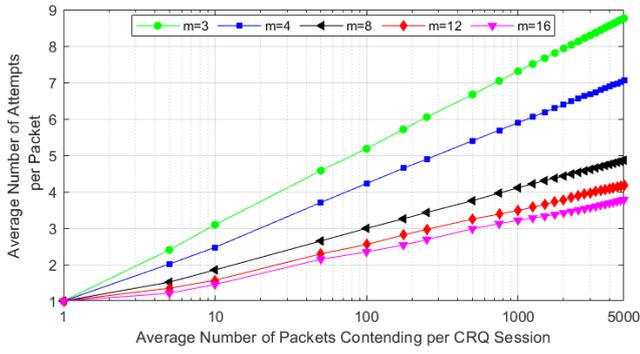


Figure 4. Average number of attempts per packet (both periodic and urgent) during the contention resolution vs. average number of contending packets during a CRQ session

It can be observed from Figure 4, that on average the number of attempts per packet grows logarithmically with the number of packets contending at the beginning of the contention. Besides, the metric $RA_{attempts}$ decreases logarithmically with the number of contention slots in the DQ frame on average. A DQ frame with more than eight contention slots has a slight impact on the number of random access attempts in general.

The energy spent by a node during the channel contention for a packet increases proportionally with the number of attempts required for that packet to leave the CRQ. However, an increase in the number of contention slots would imply a more sophisticated node in terms of synchronization, leading to an expensive and large node. Therefore, a trade-off has to be made depending on the importance of the performance metric under consideration.

4.4. Average distribution of the contention slots during the collision resolution time

A contention slot may be in one of these states: success, idle (or empty) and with collision(s). Let p_s , p_e and p_c represent respectively the allocation of the contention slots into successful, empty and those with collisions during a contention process. The parameters p_s , p_e and p_c are measured in terms of proportions as follows:

$$\begin{cases} p_s = \frac{\text{Number of successful contention slots}}{m} \\ p_e = \frac{\text{Number of empty contention slots}}{m} \\ p_c = \frac{\text{Number of contention slots with collisions}}{m} \end{cases} \quad (20)$$

with $p_s + p_e + p_c = 1$.

It has been found through numerical simulations that the average distribution of the contention slots into p_s , p_e and p_c is independent of the number of nodes at the beginning of the contention process. It varies only with the number of contention slots m as presented in Table 3.

An optimal ratio between the three metrics is observed in the case when the number of contention slots in a frame is three, i.e., $m = 3$. A further increase in the number of

contention slots leads to their inefficient use because the number of empty contention slots will be larger than the remaining two metrics.

Table 3. Average distribution of the contention slots during the collision resolution time

Number of contention slots	Ratios		
	p_s	p_e	p_c
$m = 3$	0.36	0.31	0.33
$m = 4$	0.34	0.40	0.26
$m = 8$	0.27	0.61	0.12
$m = 12$	0.20	0.71	0.09
$m = 16$	0.18	0.76	0.06

4.5. Average drop rate per application and per node

Any packet, independent of its type, is dropped once it has reached its timeout period or if another packet is generated before it has entered the CRQ. We consider that for periodic packets the timeout corresponds to the application period T_i . As for urgent packets, it is defined by the average inter-arrival time between two packets from the same application. The inter-arrival time between packets follows an exponential distribution with a parameter λ_i . Therefore, the expected timeout for an urgent packet from an i th application is the expected value of the exponential distribution $1/\lambda_i$.

Let us define $\delta_p^{(i)}$ and $\delta_u^{(i)}$ as the average drop rate of periodic and urgent packets for a node from an i th application, respectively. Following the network traffic load, we have:

- (i) if $n_x \leq n_x^{(l)}$, then a packet (either periodic or urgent) resolves its contention instantaneously in the same DQ frame as it was generated. No packets are dropped in the network. Therefore, we have $\delta_p^{(i)} = 0$ and $\delta_u^{(i)} = 0$.
- (ii) if $n_x > n_x^{(l)}$, then several cases are observed depending on the type of the packet and its priority.

Let us define the average drop rate for each type of the packet separately. For periodic packet, when there are no urgent packets in the network, the average drop rate $\delta_p^{(i)}$ of periodic packets from an i th application is defined by the average collision resolution time t_{crq} and the application period T_i . A periodic packet from an i th application is dropped only if the average collision resolution time t_{crq} is larger than the application period T_i . Therefore, the waiting time for a periodic packet before the contention follows a discrete uniform distribution $U \sim (0, t_{crq})$. The probability for a periodic packet to be dropped, i.e., the probability for a periodic packet to wait before the contention an interval of time greater than its period T_i , is:

$$p_w^{(i)}(W > T_i) = \sum_{T_i+1}^{t_{crq}} \frac{1}{t_{crq}+1} \quad (21)$$

Therefore, in the long run, the average drop rate $\delta_p^{(i)}$ of periodic packets from an i th application, considering that no urgent packets are generated in the network, is:

$$\delta_p^{(i)} = \begin{cases} 0 & \text{if } T_i \geq t_{crq} \\ p_w^{(i)}(W > T_i) & \text{if } T_i < t_{crq} \end{cases} \quad (22)$$

In a network with urgent packets, a periodic packet is dropped if: (i) another periodic packet is generated before the former is sent, i.e., if it has reached its application timeout (T_i) or (ii) if at least one urgent packet is generated before it is sent and its timeout is reached. The periodic and urgent packets are generated independent of each other. Therefore, the average drop rate $\delta_p^{(i)}$ of periodic packets for a node from an i th application can be defined as follows:

(i) if $T_i \geq t_{crq}$, then we have:

$$\delta_p^{(i)} = p_u^{(i)}(U \geq 1)p_p^{(i)}(P = 1) = \frac{(1 - e^{-\lambda_i t_{crq}})t_{crq}}{T_i} \quad (23)$$

where $p_u^{(i)}(U \geq 1)$ is the probability of having at least an urgent packet generated during the collision resolution time t_{crq} for a node from an i th application.

It is defined following (8). The parameter $p_p^{(i)}(P = 1) = t_{crq}/T_i$ is the probability of having a periodic packet generated during the collision resolution time t_{crq} for a node from an i th application.

(ii) if $T_i < t_{crq}$, then we have:

$$\delta_p^{(i)} = p_w^{(i)}(W > T_i) + p_u^{(i)}(U \geq 1) - p_w^{(i)}(W > T_i)p_u^{(i)}(U \geq 1) \quad (24)$$

which leads to :

$$\delta_p^{(i)} = \frac{(t_{crq} - T_i + 1)}{t_{crq} + 1} + (1 - e^{-\lambda_i t_{crq}}) - \frac{(t_{crq} - T_i + 1)(1 - e^{-\lambda_i t_{crq}})}{t_{crq} + 1} \quad (25)$$

The average drop rate $\delta_u^{(i)}$ of urgent packets for a node from an i th application is defined by the average collision resolution time t_{crq} and by the application rate parameter λ_i . An urgent packet is dropped if another urgent packet is generated before the older packet has entered the CRQ, or if the older packet has reached its timeout period. Therefore, assuming that j urgent packets are generated during the collision resolution time, $(j - 1)$ urgent packets are dropped. Thus, we have:

$$\delta_u^{(i)} = \frac{\sum_{j=2}^{\infty} (\lambda_i t_{crq})^j e^{-\lambda_i t_{crq}} (j-1)}{j! \lambda_i t_{crq}} = \frac{(\lambda_i t_{crq} + e^{-\lambda_i t_{crq}} - 1)}{(\lambda_i t_{crq})} \quad (26)$$

In Figure 5, we present the analytical and simulation results for the average drop rate of both types of the packet over the average traffic load in network with three applications and for different scenarios. The application rates for urgent packets are the same for all the applications.

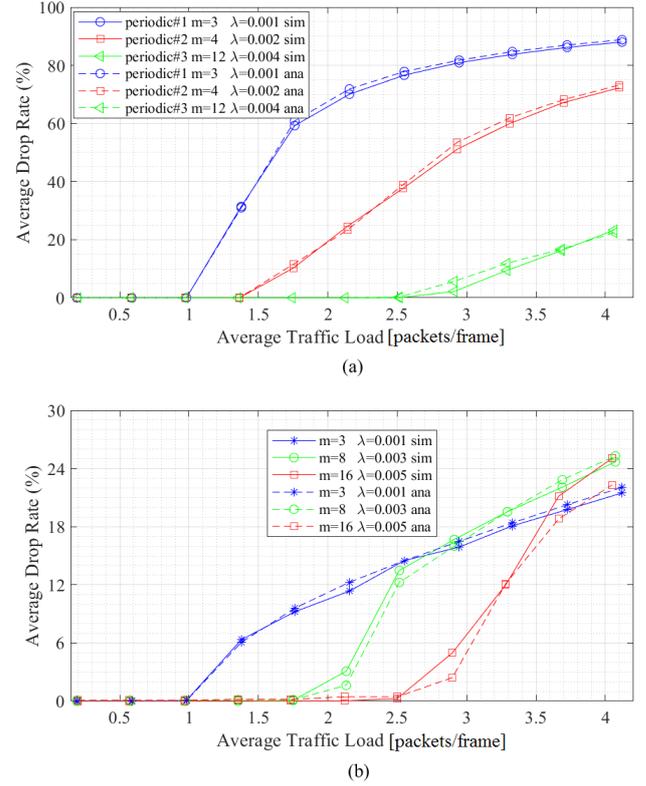


Figure 5. Average drop rate vs. average traffic load for (a) periodic and (b) urgent packets for different scenarios

It can be observed that when the traffic load from the network is small, the packets are not dropped for any of the applications. The nodes are getting immediate access to the channel. However, as the traffic increases to the medium-to-high interval, the packets are dropped depending on the application period for periodic packets and the rate parameter for urgent packets.

Therefore, an increase in the number of contention slots would improve the packet drop characteristic of the DQ algorithm. As a result, the algorithm can support significant traffic load from the network without the packets being dropped.

4.6. Average waiting time per packet in the DTQ

Let us consider the DTQ subsystem to comprise of c (with $c=1,2,\dots$) DTQ channels because the nodes are contending in a multichannel communication network. After leaving the CRQ, both the periodic and urgent packets are uniformly distributed within all the DTQ channels with a probability mass function $p_{dtq} = 1/c$.

Let crq_{out} be the average service rate from the CRQ. Depending on the average traffic load from the network, we have:

(i) if $n_x \leq n_x^{(l)}$, then all the contending packets get immediate access to the channel and enter the DTQ subsystem. Therefore, the average service rate from the CRQ corresponds to the average number of packets generated during a DQ frame:

$$crq_{out} = n_x \quad (27)$$

(ii) if $n_x > n_x^{(l)}$, then the average service rate crq_{out} from the CRQ was observed to increase only with the number of contention slots m in a DQ frame. In Table 4, we present the results of the average service rate from the CRQ obtained statistically through numerical simulations. It should be noted in Table 4 only the uplink traffic is considered.

Table 4. Average service rate from the CRQ for the DQ algorithm

Number of contention slots	crq_{out} , packets/frame	
	μ	σ^2
$m = 3$	1.10	1.0×10^{-4}
$m = 4$	1.39	4.0×10^{-4}
$m = 8$	2.08	4.0×10^{-4}
$m = 12$	2.44	2.5×10^{-3}
$m = 16$	2.83	2.1×10^{-2}

Given the data presented in Table 4, the service rate from the CRQ can be considered small. At most, three packets leave the CRQ per frame independent of the number of packets contending per CRQ session for $m \leq 16$.

If the base station shares the data slots with the nodes, then the service rate from the CRQ should be incremented with the traffic from the base station. Therefore, the total service rate should be:

$$crq_{tot} = crq_{out} + \lambda_{bs} \quad (28)$$

where λ_{bs} is the traffic rate from the base station to the rest of the network in packets per frame.

At the end of the contention resolution process, the packets are distributed with equal priority in the DTQ subsystem. Consequently, the CRQ service rate corresponds to the DTQ subsystem arrival rate dtq_{in} . Therefore, we have:

$$dtq_{in} = crq_{out} \quad (29)$$

Let us consider the DTQ subsystem as an M/D/c queue where the packets enter the queue following a Poisson arrival process with a rate parameter dtq_{in} . Each DTQ channel can serve only a packet during a DQ frame. Therefore, the service rate from each DTQ channel follows a degenerate distribution with a rate parameter of $\mu = 1$ and the number of servers for the model corresponds to the number, c , of DTQ channels. The queue buffer is considered indefinite because there is no limit on the number of packets that the DTQ subsystem can hold. Consequently, the average utilization factor for the DTQ subsystem ρ can be defined as follows:

$$\rho = \frac{crq_{out}}{\mu c} \quad (30)$$

The DTQ subsystem in a multichannel network with n nodes from N applications with c channels is stable if the utilization factor is less than one, i.e., $\rho < 1$. Therefore, for a given number of contention slots, the condition for the stability of the DQ algorithm can be stated as follows:

$$\begin{cases} n_x < c & \text{if } n_x \leq n_x^{(l)} \\ crq_{out} < c & \text{if } n_x > n_x^{(l)} \end{cases} \quad (31)$$

In Figure 6, we present the variation of the utilization factor of the DTQ subsystem with different channels over the network traffic load. It can be observed that when we have $c \geq n_x^{(l)}$, the DTQ subsystem is stable for any traffic load from the network.

It can also be observed that the stability of the DQ algorithm is improved with the number of DTQ channels. However, due to the small service rate from the CRQ, the DQ algorithm is stable for any traffic load from the network with at least three DTQ channels for $m \leq 16$.

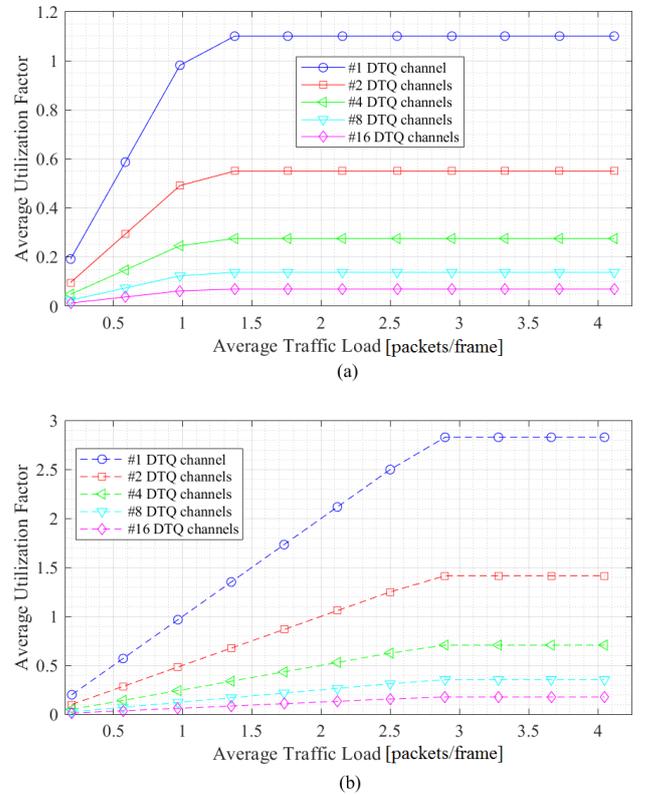


Figure 6. Average utilization factor of the DTQ subsystem vs. average traffic load for (a) $m=3$ and (b) $m=16$ contention slots

Let $t_{dtq/packet}$ be the average data transmission time per packet (both periodic and urgent), i.e., the average time a packet spends in the DTQ subsystem waiting to be transmitted to the wireless channel. Considering the DTQ subsystem to be an M/D/c queue and using Little's law [39] about the mean service time, the average data transmission time per packet can be defined as follows:

$$t_{dtq/packet} = \frac{L(\rho) - \rho}{crq_{out}} \quad (32)$$

where $L_{(\rho)}$ is the steady-state expected number of packets in the DTQ subsystem depending on the channel utilization factor ρ . The values of $L_{(\rho)}$ have been tabulated and are given graphically in [39].

Therefore, taking into consideration the average traffic load n_x from the network, the average data transmission for any type of the packet is:

$$t_{dq/packet} = \begin{cases} \frac{L_{(\rho)}}{n_x} - \frac{1}{c} & \text{if } n_x \leq n_x^{(l)} \\ \frac{L_{(\rho)}}{c_{rqout}} - \frac{1}{c} & \text{if } n_x > n_x^{(l)} \end{cases} \quad (33)$$

In Figure 7, the analytical and simulation results of the average data transmission time per packet $t_{dq/packet}$ over the network traffic load are given. It can be observed that an increase in the number of DTQ channels leads to a decrease in the average data transmission time.

Moreover, it can also be observed that both the number of contention slots and the network traffic load have a slight impact on that performance metric. For $c = 1$, the significant values obtained when the network traffic is high, i.e. $n_x \geq 1$, are due to the instability of the DQ algorithm. For comparison goals, the values of the analytical and numerical solutions are considered the same when the DQ algorithm is unstable in Figure 7.

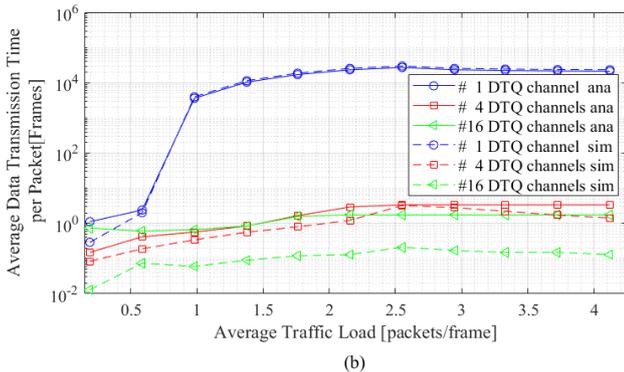
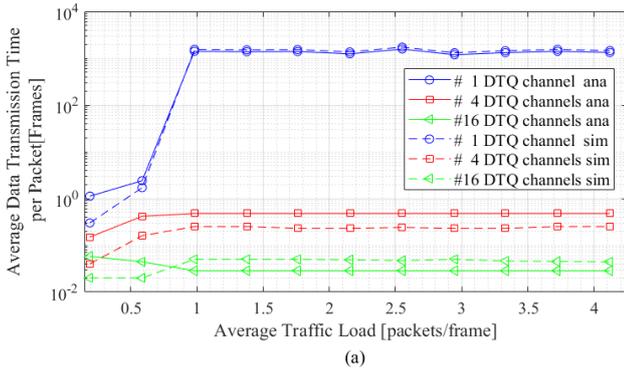


Figure 7. Average data transmission time per packet

Compared to the waiting time before entry in the CRQ and the time spent in the CRQ, the data transmission time per packet is very negligible when the DQ algorithm is stable. However, as it can be observed from Figure 8, the introduction of the traffic from the base station has a

significant impact on the time spent by a node in the DTQ subsystem and the utilization factor of the DQ algorithm. With the downlink traffic, the service rate from the CRQ is increased. Therefore, the data transmission time per packet and the average utilization of the DTQ subsystem are also increased. Consequently, the network traffic load that can be serviced by the DQ algorithm is decreased.

The traffic from the base station, once it shares the resources with the nodes, needs to be introduced carefully because it may lead to the DQ algorithm instability.

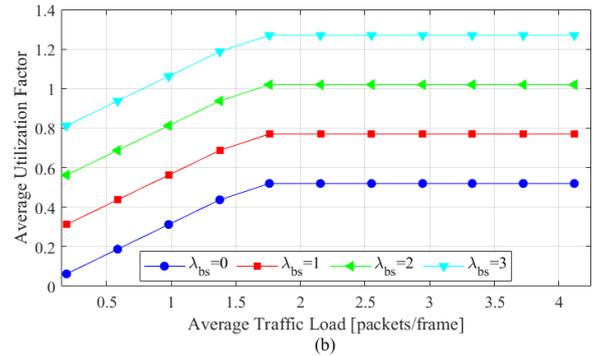
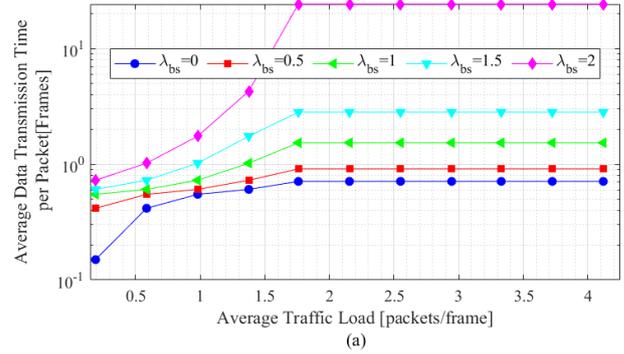


Figure 8. Impact of the traffic from the base station on (a) the data transmission time per packet and (b) the utilization factor

4.7. Average throughput per DTQ channel

Let us define C_i as the average throughput for an i th DTQ channel, i.e., the average number of successful data slots. A DQ frame is considered successful if a packet is assigned to the data slot of that DQ frame for transmission. After leaving the CRQ, the packets are distributed uniformly within all the DTQ channels with a probability mass function $p_{dq} = 1/c$. Therefore, we have:

- (i) if $n_x \leq n_x^{(l)}$, then the average number of packets entering the DTQ subsystem in the long run corresponds to the average number of packets generated during a DQ frame n_x . Thus, the average throughput for an i th DTQ channel is:

$$C_i = \frac{n_x}{c} \quad (34)$$

- (ii) if $n_x > n_x^{(l)}$, then on average crq_{out} packets are entering the DTQ subsystem and therefore, the average throughput for the i th DTQ channel is:

$$C_i = \frac{crq_{out}}{c} \quad (35)$$

For a given DTQ channel, as the average service rate is one, i.e., $\mu = 1$, the average throughput corresponds to the average utilization factor of the DTQ subsystem when the DQ algorithm is stable. In addition, the parameter C_i is also independent of the network traffic for $n_x > n_x^{(l)}$.

The performance metric $(1 - C_i)$ is the average number of empty DQ frames and those with collisions for an i th DTQ channel. However, if the first rule of data transmission is not taken into consideration, or if the probability of collision in an empty data slot is insignificant, then the number of DQ frames with collisions can be neglected compared to the number of empty DQ frames. Therefore, the parameter $(1 - C_i)$ corresponds to the average number of empty frames for a given DTQ channel i .

In Figure 9, it can be observed that the increase in the number of DTQ channels in the system leads to a decrease in the average throughput per DTQ channel by increasing the number of empty data slots per channel. Thus, a trade-off needs to be made between the average waiting time in the DTQ subsystem and the average channel throughput. On the other hand, a mechanism such as a system with multiple parallel CRQ channels or an adaptation of the first data transmission rule to such a system should be included in the DQ algorithm to allow an efficient use of the empty data slots for data transmission.

The introduction of the traffic from the base station can also be used to improve the utilization of the data slot resources. Consequently, the number of DTQ channels should be chosen according to the total service rate from the CRQ and the base station.

5. Conclusion and future works

In this paper, a performance evaluation of the DQ algorithm in a multichannel environment has been presented. The network is composed of nodes from various applications that generate traffic comprising of periodic and urgent packets with different periods and rates. However, the urgent packets are assumed to have a high priority for contention over the periodic ones.

After an adaption of the classic DQ algorithm to a multichannel setting with multiple DTQ channels, we have presented a steady-state evaluation of different algorithm performance metrics. These are the number of packets contending per CRQ session, the waiting time for any packet in the DQ system before its transmission, the number of attempts required by a packet to access the channel, the drop rate for both types of the packet, the distribution of contention slots, and the system throughput.

The number n_{crq} of packets contending per CRQ session has been found to increase with the network traffic load. Besides, that number was further used to define three different intervals of the traffic from the network: small, medium and high. The first interval is characterized by immediate access

to the channel for any node present in the network. As for the high traffic interval, due to a significant collision resolution time, all the nodes are taking part in every contention. The medium traffic interval is characterized by a number of nodes contending per CRQ session that varies from a value corresponding to the network traffic load up to the total number of nodes in the network. The lengths of the traffic intervals were found to increase logarithmically with the number of contention slots in the DQ frame.

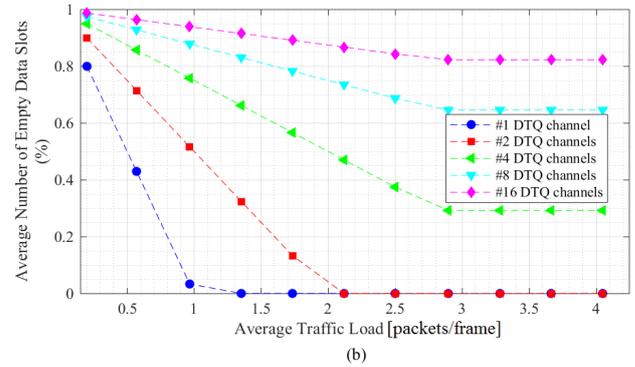
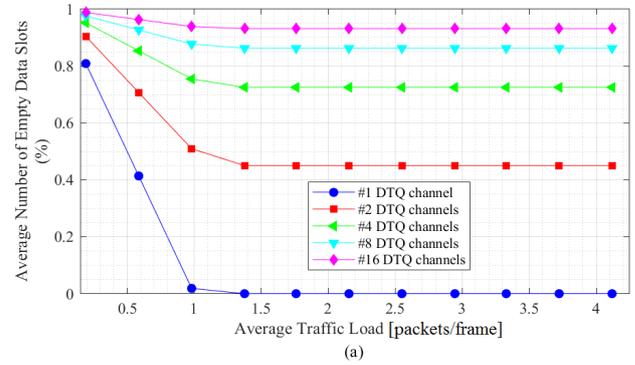


Figure 9. Average number of empty data DQ frames for each DTQ channel vs. average traffic load for (a) $m=3$ and (b) $m=16$ contention slots

For better performance in terms of access delay and packet drop rate, the operation of DQ algorithm in the small traffic interval is preferable because the nodes are assured of getting an immediate access to the channel. The data transmission time was found to be negligible compared to the waiting time before entry in the CRQ and the time spent in the CRQ by a node for a given packet as long as the algorithm was stable.

However, the introduction of the traffic from the base station was found to have a significant impact on the performance of the DQ algorithm. The overall service rate from the CRQ increased with the downlink traffic. Therefore, both the data transmission time per packet and the utilization factor of the data slots were also increased. However, a significant traffic load from the base station was observed to lead to the instability of the DQ algorithm when the overall service rate from the CRQ was greater than the number of DTQ channels in the network.

In general, an increase in the number of contention slots was found to improve all the DQ algorithm performance metrics. The number of contention slots in the DQ frame

should be chosen following the quality of service required by the applications present in the network. Nevertheless, an increase in the number of contention slots needs to be considered carefully because it implies a more precise synchronization technique for the nodes in the network. Otherwise, the concept of a low cost, low power and small node may be compromised

In future work, we plan to evaluate the performance of a DQ system with multiple parallel CRQ channels to increase the service rate from the CRQ to mitigate the inefficient use of the system throughput and to decrease the drop rate for both types of the packet. Moreover, due to the limited computing available, in this paper the number of contention slots in the DQ frame were limited to 16. Thus, the relation between the performance metrics and the number of contention slots for $m > 16$ should also be investigated further to confirm the trends obtained in this paper.

Acknowledgements

The Pan African University funded this research.

Data availability

The data used to support the findings of this study are available from the corresponding author upon request.

References

- [1] Ericsson, "Ericsson Mobility Report (June 2020)," *Ericsson White Paper*, p. 36, Jun. 2020. [Online]. Available: www.ericsson.com/mobilityreport
- [2] J. Ding, M. Nemati, C. Ranaweera, and J. Choi, "IoT Connectivity Technologies and Applications: A Survey," *IEEE Access*, pp. 67646 – 67673, 2020.
- [3] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.
- [4] F. Vazquez-Gallego, P. Tuset-Peiro, L. Alonso, and J. Alonso-Zarate, "Combining distributed queuing with energy harvesting to enable perpetual distributed data collection applications," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 7, p. e3195, 2018.
- [5] A. Laya, L. Alonso, and J. Alonso-Zarate, "Contention resolution queues for massive machine type communications in LTE," *IEEE 26th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*, Hong Kong, China, pp. 2314–2318, Aug. 2015.
- [6] S. Xing, X. Wen, Z. Lu, Q. Pan, and W. Jing, "A novel distributed queuing-based random access protocol for Narrowband-IoT," *IEEE International Conference on Communications (ICC)*, Shanghai, China, pp. 1–7, May 2019.
- [7] W. Wu, Y. Li, Y. Zhang, B. Wang, and W. Wang, "Distributed Queuing Based Random Access Protocol for LoRa Networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 763–772, 2020.
- [8] A. Laya, C. Kalalas, F. Vazquez-Gallego, L. Alonso, and J. Alonso Zarate, "Goodbye, ALOHA!" *IEEE access*, vol. 4, pp. 2029–2044, 2016.
- [9] A.-T. H. Bui, C. T. Nguyen, T. C. Thang, and A. T. Pham, "Free Access Distributed Queue Protocol for Massive Cellular-based M2M Communications with Bursty Traffic," *IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Chicago, USA, pp. 1–5, Aug. 2018.
- [10] H. A. T. Bui, C. T. Nguyen, T. C. Thang, and A. T. Pham, "A Comprehensive Distributed Queue-based Random Access Framework for mMTC in LTE/LTE-A Networks with Mixed-Type Traffic," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 12107– 12120, 2019.
- [11] J. Yuan, H. Shan, A. Huang, T. Q. Quek, and Y. D. Yao, "Massive machine-to-machine communications in cellular network: Distributed queuing random access meets MIMO," *IEEE Access*, vol. 5, pp. 2981– 2993, 2017.
- [12] F. Metzger, T. Hofbeld, A. Bauer, S. Kounev, and P. E. Heegaard, "Modeling of aggregated IoT traffic and its application to an IoT cloud," *Proceedings of the IEEE*, vol. 107, no. 4, pp. 679–694, 2019.
- [13] M. Laner, N. Nikaein, P. Svoboda, M. Popovic, D. Drajjic, and S. Krco, "Traffic models for machine-to-machine (M2M) communications," *Machine-to-machine (M2M) Communications: Architecture, Performance and Applications*, p. 133, 2014.
- [14] M. Sansoni, G. Ravagnani, D. Zucchetto, C. Pielli, A. Zanella, and K. Mahmood, "Comparison of M2M traffic models against real world data sets," *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Barcelona, Spain, pp. 1–6, Sep. 2018.
- [15] M. Centenaro and L. Vangelista, "A study on M2M traffic and its impact on cellular networks," *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Milan, Italy, pp. 154–159, Dec.2015.
- [16] N. Nikaein, M. Laner, K. Zhou, P. Svoboda, D. Drajjic, M. Popovic, and S. Krco, "Simple traffic modeling framework for machine type communication," *The Tenth International Symposium on Wireless Communication Systems*, Ilmenau, Germany, pp. 1–5, Aug. 2013.
- [17] K. Smiljkovic, V. Atanasovski, and L. Gavrilovska, "Machine-to-Machine traffic characterization: Models and case study on integration in LTE," *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*, Aalborg, Denmark, pp. 1–5, May 2014.
- [18] W. Xu and G. Campbell, "A near perfect stable random access protocol for a broadcast channel," *IEEE International Conference on Communications*, Chicago, Illinois, USA, 1992. ICC'92, Conference record, SUPERCOMM/ICC'92, Discovering a New World of Communications., pp. 370–374, Jun. 1992.
- [19] H.-J. Lin and G. Campbell, "PDQRAP-Prioritized Distributed Queuing Random Access Protocol." *Proceedings of 19th Conference on Local Computer Networks*, Minneapolis, Minnesota, USA, pp. 82–91, Oct. 1994.
- [20] C.-T. Wu and G. Campbell, "Extended DQRAP (XDQRAP) A cable TV protocol functioning as a distributed switch," *Proc.1st International Workshop on Community Networking*, San Francisco, USA. Computer Communication Review, vol. 23, no. 4, pp. 270–278, Jul. 1994.
- [21] J. Alonso-Zarate, C. Verikoukis, E. Kartsakli, A. Cateura, and L. Alonso, "A near-optimum cross-layered distributed queuing protocol for wireless LAN," *IEEE Wireless Communications*, vol. 15, no. 1, pp. 48–55, 2008.
- [22] L. Alonso, R. Agusti, and O. Sallent, "A near-optimum MAC protocol based on the distributed queuing random access protocol (DQRAP) for a CDMA mobile communication system," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, pp. 1701–1718, 2000.

- [23] B. Ota, L. Alonso, and C. Verikoukis, "Towards energy saving wireless body node networks in health care systems," *IEEE International Conference on Communications Workshops (ICC)*, Cape Town, South Africa, pp. 1–5, May 2010.
- [24] J. Alonso-Zarate, C. Verikoukis, E. Kartsakli, A. Cateura, and L. Alonso, "Saturation throughput analysis of a passive cluster-based medium access control protocol for ad hoc wireless networks," *IEEE International Conference on Communications*, Beijing, China, pp. 2348–2352, Dec. 2008.
- [25] J. Alonso-Zarate, C. Verikoukis, E. Kartsakli, and L. Alonso, "A novel near-optimum medium access control protocol for a distributed cooperative ARQ scheme in wireless networks," *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, Cannes, France, pp. 1–5, Sep. 2008.
- [26] X. Zhang and G. Campbell, "Performance analysis of distributed queueing random access protocol-DQRAP," *DQRAP Research Group Report*, vol. 93, no. 1, pp. 1–24, 1993.
- [27] F. Vazquez-Gallego, J. Alonso-Zarate, P. Tuset-Peiro, and L. Alonso, "Energy analysis of a contention tree-based access protocol for machine-to-machine networks with idle-to-saturation traffic transitions," *IEEE International Conference on Communications (ICC) 2014*, Sydney, Australia, pp. 1094–1099, Jun. 2014.
- [28] P. Tuset-Peiro, F. Vazquez-Gallego, J. Alonso-Zarate, L. Alonso, and X. Vilajosana, "LPDQ: A self-scheduled TDMA MAC protocol for one-hop dynamic low-power wireless networks," *Pervasive and Mobile Computing*, vol. 20, pp. 84–99, 2015.
- [29] A. Samir, M. M. Elmesalawy, A. S. Ali, and I. Ali, "An Improved LTE RACH Protocol for M2M Applications," *Mobile Information Systems*, 2016.
- [30] C. Yoon, "Distributed queueing with preamble grouping for massive IoT devices in LTE random access," *International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, South Korea, pp. 103–105, Oct. 2016.
- [31] K. Lee and J. U. W. Jang, "An Efficient Contention Resolution Scheme for Massive IoT Devices in Random Access to LTE-A Networks," *IEEE Access*, vol. 6, pp. 67118–67130, Nov. 2018.
- [32] R. G. Cheng, Z. Becvar, and P. H. Yang, "Modeling of Distributed Queueing-Based Random Access for Machine Type Communications in Mobile Networks," *IEEE Communications Letters*, vol. 22, no. 1, pp. 129–132, 2018.
- [33] A. Marchiori, "Maximizing coverage in low-power wide-area IoT networks," *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Kona, Big Island, Hawaii, USA, pp. 467–472, Mar. 2017.
- [34] R. Nibitanga, E. Mwangi, and E. Ndung'u, "Steady-State Analysis of the Distributed Queueing Algorithm in a Single-Channel M2M Network," *Journal of Computer and Communications*, vol. 8, pp. 28–40, 2020.
- [35] R. Nibitanga, E. Mwangi, and E. Ndung'u, "Statistical Analysis of a Distributed Queueing Random Access Protocol in a Massive Communication Environment," *EAI Endorsed Transactions on Internet of Things*, vol. 6, no. 23, 2020.
- [36] A. M. Law, *Simulation modeling and analysis*. McGraw-Hill New York, USA, 2015.
- [37] J. Banks, J. Carson, B. Nelson, and D. Nicol, *Discrete-Event System Simulation*. Pearson, USA, 2014.
- [38] A. J. Janssen and M. de Jong, "Analysis of contention tree algorithms," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2163–2172, 2000.
- [39] G. J. Lieberman and F. S. Hillier, *Introduction to operations research*. McGraw-Hill New York, USA, 2015.