QBAIoT: QoS Based Access for IoT Environments

Ahmad Khalil, Nader Mbarek, Olivier Togni
University of Bourgogne Franche-Comté
Dijon – France
emails: Ahmad.Khalil@u-bourgogne.fr, Nader.Mbarek@u-bourgogne.fr, Olivier.Togni@u-bourgogne.fr

***Abstract*—Providing users with a service level guarantee in the Internet of Things (IoT) is a challenging task in order to ensure a better user experience in such environment. We present in this paper an adaptation of the slotted Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) method used in the Media Access Control (MAC layer) of the IEEE 802.15.4 standard in order to provide IoT smart objects with a differentiated wireless access according to the Quality of Service (QoS) class of their generated traffic (Real Time Mission Critical, Real Time Non Mission Critical and Non Real Time). The proposed method ensures a service level guarantee for a Low Rate Wireless Personal Area Network (LR-WPAN) in an IoT environment. Our adaptation consists in creating different Contention Access Periods (CAP); each will be specific for a traffic type and so for a specific QoS class. To do so, we propose firstly a QoS based wireless access method specified by an algorithm to be used by the coordinator, known as the gateway in the IoT architecture. Secondly, we propose an algorithm used by the IoT smart objects. This method enables the coordinator to configure different contention periods with a specific number of slots so that the nodes (i.e., IoT objects) of the same QoS class will access the channel only during their respective contention periods without collision with the nodes of other classes.**

Keywords-Internet of Things; IEEE 802.15.4; Slotted CSMA/CA; QBAIoT; IoT Gateway; IoT objects .

I. INTRODUCTION

The impact of the Internet of Things (IoT) on our everyday life is very important. It changes how people interact with connected objects in order to improve life's quality. Therefore, the enhancement of the corresponding IoT services is an important challenge enabling its expansion. In order to expand the usage of this environment, a better user experience is expected. Consequently, Quality of Service (QoS) mechanisms should be implemented within the IoT environment and especially the communication technologies used in the sensing layer of the IoT architecture such as IEEE 802.15.4 [1]. IEEE 802.15.4 standard specifies the physical (PHY) and the Media Access Control (MAC) layers to provide an important foundation for other standards. Indeed, it is used by 6LowPAN [2], ZigBee [3] for their lower layers implementation.

In this context, we specify a novel QoS based wireless access method for IoT environments called QBAIoT. It is an enhancement of the slotted Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) technique, used by the IEEE 802.15.4 standard, to ensure a differentiation between traffics while using the wireless channel of the IoT sensing layer. QBAIoT allows serving different IoT generated traffics while respecting the requirements of each traffic type (i.e., reduced delay for Real Time traffic). In this paper, we aim to present the design details of our proposed QoS based access method, as well as the corresponding simulation results. The reminder of the paper is organized as follows. We present in Section 2 the state of the art concerning the IoT environment, as well as the related technologies and we introduce the important characteristics of the IEEE 802.15.4 technology. Section 3 describes the QoS motivations in the IoT and some related research works. Then, we specify in Section 4 our proposed method enabling QoS based access for IoT environments. Section 5 presents a detailed performance evaluation of our novel access method along with a comparison with the standard access method. Finally, we conclude the paper in Section 6 and present future works.

II. STATE OF THE ART

A. IoT environment

By 2020, 20 to 30 billion objects will be connected to the Internet inducing an important expansion of the IoT [4]. The IoT uses external resources such as cloud computing and fog computing for the processing and the storage of huge amount of data. Cloud computing functionalities enhance reliability and efficiency of IoT service provision [5]. On the other hand, fog computing decentralizes the computing capacities and distributes the operations on network extremities [6]. Based on definitions and concepts presented by different standardization organizations and international research projects, we can propose the following IoT definition: IoT is a system of systems interconnected via standard and interoperable communication technologies. This interconnection allows creating a considerable network of communicating objects, each addressed uniquely, in order to offer new services for improving the quality of human life. Also, self-management capabilities are essential in the IoT in order to offer autonomous self-managed objects.

Different application domains with a variety of services are used in the IoT environment. These application domains cover a wide variety of everyday services like health services, industry services, road services, city management services, etc. In order to offer these services, various communication technologies interconnect IoT objects and gateways within IoT environments. Each technology is suitable for a specific scenario based on different criteria such as energy consumption, CPU utilization, range of the technology, etc. We describe in the following section one of these technologies, which is the foundation of our proposed QoS based access method.

B. IEEE 802.15.4

The IEEE 802.15.4 standard is an IEEE proposed standard for Wirelsss Personal Area Networks with low data rate (LR-

 ISBN: 978-1-61208-650-7

WPAN). It defines the physical and the MAC layers to provide a basic format. This format will be used by other technologies and protocols by adding their own specificities through the specification of the higher layers. The IEEE 802.15.4 physical layer specifies different essential parameters: 250 Kbit/s of data rate for a 2.4 GHz band, control functions like the activation or deactivation of the radio module, the test of the channel occupation and the choice of the transmission channel. On the other hand, the MAC layer defines the data management format and specifies the usage of the slotted CSMA/CA. It provides also some management features, such as access to medium, frame exchanges, synchronization, etc. As for data encryption, the IEEE 802.15.4 standard uses AES-128 to ensure data confidentiality [7]. Different standards use IEEE 802.15.4 as a foundation for their lower layers. We can mention as an example the 6LowPAN standard that combines IPv6 with low power WPAN networks. Another example is ZigBee, a specification for a series of high-level, low-power communication.

IEEE 802.15.4 supports a beacon-enabled mode using a superframe structure. The superframe consists of an active part known as the Superframe Duration (*SD*) and can be followed by an inactive period. The active part is formed by 16 equally sized time slots partitioned into a Contention Access Period (*CAP*) where nodes compete to gain the access to the channel; and an optional Contention Free Period (*CFP*) where nodes are allocated guaranteed time slots. In beacon-enabled mode, the coordinator sends periodically a beacon frame on the network including all the superframe specifications. The beacon, sent at the Beacon Interval (*BI*) time, allows the coordinator to identify its WPAN and ensure that all the objects are synchronized. The Becon Order (*BO*) and Superframe Order parameters determine the Beacon Interval (*BI*) and *SD*, respectively as mentioned in (1) and (2). The Base Superframe Duration (*BSFD*) corresponds to the minimum duration of the superframe ($SO = 0$).

$$BI = BSFD * 2^{BO} \tag{1}$$

$$SD = BSFD * 2^{SO} \tag{2}$$

BSFD is fixed to 960 symbols of 4 bits or 15.36 ms assuming the data rate of 250 Kbit/s for the 2.4 GHz band. In addition, *BO* and SO should respect the inequality $0 \leq SO \leq BO \leq 14$ [7].

Three variable are used in the slotted CSMA/CA algorithm: the Backoff Exponent (*BE*), the Contention Window (*CW*) and the Number of Backoffs (*NB*). To compute the backoff delay, that an object has to observe before performing the Clear Channel Assessment (*CCA*), the algorithm chooses a random value for the backoff delay between 0 and ($2^{BE} - 1$). CW is the number of backoff periods during which the channel must be idle before accessing the channel. By default, the value of *CW* is fixed to 2. *NB* is the number of backoff executed for channel access. This value is initialized to 0 and is compared to a maximum value, *macMaxCSMABackoffs* by default equal to 5. In case the *NB* value is greater than this maximum value, a failure occurs.

- The slotted CSMA/CA algorithm is activated for each transmission of a new packet and is executed during the *CAP* as follows [7]: *NB* and *CW* are initialized
- If the battery life extension is true *BE* is initialized to the minimum between 2 and *macMinBE* (by default 3). If the battery life extension parameter is fixed to false, *BE* is initialized to 2
- The node using the algorithm waits the backoff delay, and then perform *CCA*
- If the channel is busy, *CW* is re-initialized to 2, *NB* and *BE* are incremented. *BE* must not exceed *aMaxBE* (by default 5). If *macMaxCSMABackoffs* is reached, the algorithm reports a failure to the higher layer. If *NB* < *macMaxCSMABackoffs*, the backoff operation is restarted and the *CCA* should be performed again
- If the channel is sensed idle and *CW* > 0, the *CCA* is repeated and *CW* decremented. Otherwise, the node attempts to transmit if the remaining time in the current *CAP* is sufficient to transmit the frame and receive the acknowledgement. If not, the process is deferred to the next superframe.

III. QoS GUARANTEE IN THE IoT

A. *Motivations and challenges for QoS guarantee in IoT*

The ITU-T E.800 [8] has defined QoS as the totality of the characteristics of a telecommunication service to satisfy in order to meet the user requirements. In this context, a QoS requirement is expressed in terms of QoS parameters (Delay, Jitter, Packet Delivery Ratio, Effective Data Rate, etc.). QoS guarantee in the IoT environment requires an effective and optimized management of the corresponding resources to improve users' experience. In order to provide predictable services, QoS mechanisms in the IoT environment handle delays, jitter, bandwidth and packet loss ratio by classifying traffic. As the IoT environment is made of different technologies and heterogeneous networks, different types of data and streams exist on a single system. Hence, it is important to provide the IoT environment with QoS guarantee mechanisms to meet the requirements of each type of traffic [9]. QoS guarantee is a critical challenge in the IoT, as the number of connected objects increases considerably leading to a greater amount of created and transported data with different characteristics. Consequently, the performance of the IoT system will be affected and especially QoS constrained data traffic due to congestion periods. Deploying QoS mechanisms within IoT environment will enhance the performance by identifying traffic and differentiating it allowing a reduced cost and a better scalability [10]. The ITU-T describes the importance of QoS integration in the IoT through various documents such as Y.2066 [1] where it was mentioned that service priority is an important requirement. In addition, Y.2066 indicates that the prioritization functionality satisfies different service requirements of IoT users.

B. *Related research work*

Different international projects and research works had studied the Quality of Service in the IoT environment and its impact on the service provision. The European project OpenIoT [11] specified different QoS parameters and metrics for the IoT. These metrics include utility metrics related tosensors and other metrics related to the network and application. As an example of utility metrics, OpenIoT indicated the Quality of sensors that determines the accuracy of measurement, the energy consumption, data volume, and bandwidth. For the other metrics, system lifetime is taken into consideration. In addition, traditional QoS parameters are used such as latency, jitter, delay, throughput, etc. On the other hand, this project presented a high level architecture based on a QoS Manager that keeps track of the following parameters: quality of sensors, energy consumption, trustworthiness, bandwidth and data volume.

Furthermore, other research works had focused on the QoS in the lower layer of the IoT achitecture (sensor layer). For example, the research work conducted in [12], tried to use different queues and a scheduler to ensure a certain priority for QoS constrained flows. Moreover, different research work tried to adapt the slotted CSMA/CA algorithm to ensure QoS guarantee. Thus, the authors present in [13] a contribution that allows the delivery of critical data with a highest priority during the CFP. In [14], the authors describe the usage of different values for *CW*, *minBE* and *maxBE* to differentiate services thanks to three different priority levels. However, these research works did not take into consideration the existence of real time applications in the IoT environment requiring a reduced delay that does not exceed milliseconds range. For this matter, our proposed QoS based access method aims to provide a differentiation between IoT objects' flows based on different QoS classes' characteristics.

IV. QoS Based Access for IoT

We describe in the following our QoS based access method for IoT environments called QBAIoT. The specification of our novel access method is based on a new superframe structure, as well as algorithms implemented within the IoT Gateway and IoT objects enabling Class based Contention Free Periods.

A. *Class based Contention Free Period Access*

Our proposed access method consists in using an IEEE 802.15.4 superframe that respects the requirements of QoS constrained applications. We had taken into consideration four types of traffic corresponding to four QoS classes as specified in a previous work [15]: Real Time Mission Critical (RTMC), Real Time Non Mission Critical (RTNMC), Streaming and Non Real Time (NRT).

Our specified Real Time QoS classes are more sensitive to delay and jitter variation. The Streaming class is more sensitive to jitter variation while the Non Real Time class is a non-constrained QoS traffic class. In order to achieve our QoS guarantee according to the requirements of these different traffics, we adapt the structure of the IEEE 802.15.4 superframe in order to include a CAP (called QoS CAP) for each traffic corresponding to a specific QoS class. Moreover, there are no CFP and inactive periods in our adapted superframe.

We had removed the inactive period to reduce the delay of Real Time generated data. In this context, we can find up to four QoS CAPs in our superframe in case the IoT gateway (Coordinator) is configured with four QoS classes (see Fig. 1).

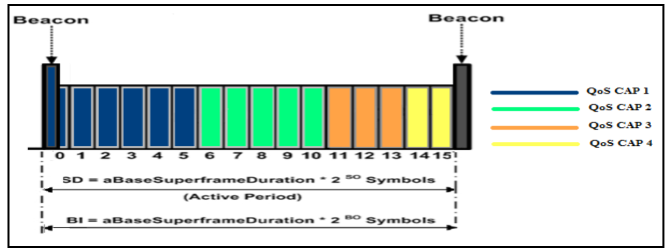

Figure 1. QBAIoT superframe structure

During each QoS CAP, only objects belonging to the corresponding QoS class can try to use the slots in order to send their data. The slots configuration and the number of QoS CAPs in the superframe is based on the number of QoS classes available in the IoT gateway environment. Different configurations for the superframe based on the existence of Real Time applications and the number of QoS classes in the considered IoT environement are possible. If the network includes one QoS class, a single CAP will exist in the superframe and the normal IEEE 802.15.4 slotted CSMA/CA algorithm is used. If there are multiple QoS classes with a minimum of one Real Time class in the network, *BO* and *SO* will be configured with the value 2 in order to minimize the latency of Real Time traffic thanks to a reduced Superframe Duration among others. Consequently, based on (1) and (2), *BI* and *SD* correspond to 61,44 ms with a slot time of 3,84 ms. If multiple QoS classes exist with no Real Time classes, *BO* and *SO* are set to 3 fixing BI and SD to 122,88 ms with a slot time of 7,68 ms. We specify for each QoS CAP a fixed number of slots. This configuration differs according to the number of existing QoS classes in the IoT Gateway environment. For example, in the case of 4 QoS classes the superframe slot configuration is as follows: RTMC class QoS CAP is allocated 6 slots, RTNMC class QoS CAP is allocated 5 slots, Streaming class QoS CAP is allocated 3 slots and NRT class QoS CAP is allocated 2 slots. So, slots configuration and the number of QoS CAP in the superframe is based on the number of existing QoS classes.

B. *IoT Gateway QoS based access method design*

For the coordinator part (i.e., IoT Gateway) of our proposed QBAIoT access method, we specify Algorithm 1 (see Fig. 2) among with the corresponding variables described in Table I.

Algorithm 1 Gateway QBAIoT Access Method Algorithm

Input: Nb_QoS_Classes, RT_Classes

N ← 1
if (Nb_QoS_Classes = 1) **then**
    BO, SO ← 14
    MAC ← Slotted_CSMA
    While true **do**
        Send_Bcn (BO, SO, CAP)
        Receive_Data ()
    end while

```
else
    if (RT_Classes = 0) then
        BO, SO ← 3
        MAC ← QBAIoT
        Initial_Slots_Configuration ()
        While true do
            Send_Bcn (BO, SO, QoS CAPs)
            While(N<=Nb_QoS_Classes) do
                Receive_Data (QoS CAP)
                N ← N + 1 // Next QoS CAP
            end while
        end while
    else
        BO, SO ← 2
        MAC ← QBAIoT
        Initial_Slots_Configuration ()
        While true do
            Send_Bcn (BO, SO, QoS CAPs)
            While(N<=Nb_QoS_Classes) do
                Receive_Data (QoS CAP)
                N ← N + 1 // Next QoS CAP
            end while
        end while
    end if
end if
```

Figure 2. Gateway QBAIoT Access Method Algorithm

TABLE I. VARIABLE SPECIFICATION OF ALGORITHM 1

Name of the variable	Description
Nb_QoS_Classes	Number of QoS classes
RT_Classes	Number of Real Time classes
N	Index of QoS classes
MAC	Channel access algorithm
QoS CAP; CAP	Configuration of the CAP (CAPStart and CAPEnd)
Initial_Slots_Configuration()	Algorithm that computes the slots configuration based on the Number of QoS classes and Number of Real Time classes.

The IoT Gateway using our QoS based access method (i.e., QBAIoT gateway) will receive data from objects during the corresponding QoS CAPs. At each Beacon Interval, the gateway sends the beacon including the information regarding the values of *BO*, *SO* and the first and final slot for each QoS CAP. These values are used by the IoT objects to calculate the slot time and to determine during which time they are allowed to compete for the channel. A QBAIoT gateway should include also self-management capabilities. A self-configuring capability enables the gateway to adapt the superframe slots configuration according to the existing number of QoS classes within its environment. A self-optimizing capability is performed in case of unused slots in a QoS CAP thanks to a slot reallocation mechanism covering the entire superframe. The self-management capabilities design is out of the scope of this paper.

C. *Class based access for IoT objects*

For the IoT object part of our proposed QBAIoT access method, we specify Algorithm 2 (see Fig. 3) among with the corresponding variables described in Table II.

Algorithm 2 Object QBAIoT Access Method Algorithm

```
Receive_Beacon (BO, SO, QoS CAPs)
Configuration (BO, SO, QoS CAPs)
while (Slot ∈ [CAPStart, CAPEnd] and Data = true) do
    if (Slotted_CSMA (Slot) = Success) then
        Send_Data (Success, PAN Coordinator)
// slotted CSMA/CA returns a success state
    else
        Send_Data (Failure, PAN Coordinator)
// slotted CSMA/CA returns a failure state
    end if
end while
if (Slot < CAPStart) then
    Wait_until (Slot ∈ [CAPStart, CAPEnd])
else
    Wait_Until (Beacon) // Wait until next superframe
end if
```

Figure 3. Object QBAIoT Access Method Algorithm

TABLE II. VARIABLE SPECIFICATION OF ALGORITHM 2

Name of the variable	Description
QoS CAP	Configuration of the CAP (CAPStart and CAPEnd)
CAP_Start_Slot	The first slot for the corresponding QoS CAP assigned to the object
CAP_End_Slot	The last slot for the corresponding QoS CAP assigned to the object

Any object in the IoT Gateway environment receives the beacon. According to the QoS class it belongs to, the object will determine during which QoS CAP it can compete to access the shared medium. When an IoT object generates data, it should test if it has the right to compete in order to send its traffic. If the corresponding QoS CAP of the object has not started, it waits until its CAP time and then competes to send the data according to our adapted slotted CSMA/CA

 ISBN: 978-1-61208-650-7

algorithm. If the object QoS CAP had passed, it should wait until the corresponding QoS CAP in the next superframe.

V. PERFORMANCE EVALUATION AND RESULTS

A. Simulation environment

In order to evaluate our proposed QBAIoT access method, we conduct a simulation study using OMNeT++ based on the IEEE 802.15.4 model [16] including all the necessary features like the beacon, the superframe structure, etc. We had adapted this model to take into consideration our proposed QoS based access method thanks to a superframe with no CFP and different QoS CAPs. In our simulation scenario, we simulated four QoS classes (RTMC, RTNMC, Streaming, NRT). We used a star topology with a single coordinator (i.e., IoT Gateway) where all devices (i.e., IoT objects) are in each other's radio range. Each device transmits data to the coordinator. The data packets are generated periodically but are transmitted during the corresponding QoS CAP. Table III shows the used simulation parameters.

TABLE III. SIMULATION PARAMETERS

Parameter	Value
Carrier Frequency	2.4 GHz
Transmitter Power	1 mW
Bit rate	250 Kbps
Simulation Time	100 s
Max Frame Retries	3
Mac Payload Size	50 Bytes

In the considered simulation scenario, we fixed the data packet generation interval to 0.25 seconds and we increased the number of IoT objects from 4 (one per QoS class) to 12 (three per QoS class). The IoT objects are sending data simultaneously as they start generating data at the same time with the same interval of packet generation.

B. Performance evaluation

The evaluation of our proposed QoS based access method is based on different performance parameters concerning the traffic of our QoS classes. The importance of these parameters depends on the characteristics of the corresponding traffic. Indeed, the average delay is very important and critical for the RTMC and RTNMC traffic whereas it is less important for Streaming traffic and not important for NRT traffic. In this context, we considered the following performance parameters.

- Average Delay: It refers to the average time experienced by a generated packet to be received by the destination. It is computed by dividing the total delay experienced for all the packets by the number of packets.
- Packet Delivery Ratio (PDR): It expresses the degree of reliability achieved by the system for successful transmissions. It is obtained by dividing the number of received packets by the number of generated packets. Non received packets are either lost due to a collision or still in the sender buffer waiting for channel access.
- Effective data rate: It evaluates the link bandwidth utilization. It is computed by multiplying the number of received packets by their sizes to obtain the total length of the data frame, which is divided by the simulation time.

Fig. 4 presents the delay evaluation for 4 QoS classes traffic while using our proposed QBAIoT access method and we compare it with the traditional IEEE 802.15.4 slotted CSMA/CA method. The Delay QoS parameter is very sensitive for RTMC and RTNMC traffic. The obtained results in Figure 4 shows that for 4 objects, our proposed method enables better delay for the RTMC traffic (10 ms less than the standard) and the RTNMC traffic (7 ms less than the standard). This difference becomes greater while increasing the number of objects. The better delays that we obtain for Real Time traffic with our proposed method are owing to the fact of giving the Real Time classes a more important number of slots in which they can send their data without any collision with other objects belonging to other non real time QoS classes. Consequently, data packets do not need to wait in buffer for a long time. They are served faster than other traffic types. Although it is not critical for NRT traffic, we notice important delays for this traffic when the total number of objects is equal to 12. This delay comes from the fact that this traffic is served during 2 slots in each superframe and that each traffic class generates the same number of packets in our scenario. So, when the number of objects in the NRT class increases, the delay will increase because the generated traffic is greater than the allocated capacity of 2 slots resulting in a great number of packets in the sending buffer.

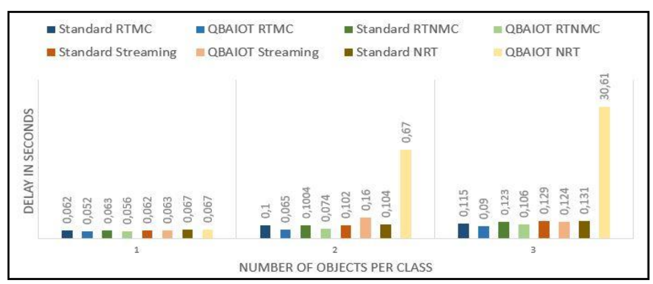

Figure 4. Delay evaluation for different traffic types using QBAIoT and IEEE 802.15.4 Standard

Fig. 5 shows the Packet Delivery Ratio for 4 QoS classes traffic while using our proposed QoS based access method and the IEEE 802.15.4 standard. Our QBAIoT access method is giving, for all QoS classes three times better PDR with one object by class, four times better PDR with two objects by class and 6 times better PDR (except NRT class 1,5 times) with 3 objects by class than IEEE 802.15.4 standard method. We obtain a better PDR with our approach thanks to an optimized channel access per class avoiding collisions between different QoS classes. Indeed, for each QoS CAP, only objects of the corresponding QoS class can compete to access the channel. This way, a lower number of objects are competing for accessing the channel for a given slot. Packets will not run the slotted CSMA/CA algorithm for several times and there is no need to drop packets after several attempts when macMaxCSMABackoffs is reached.

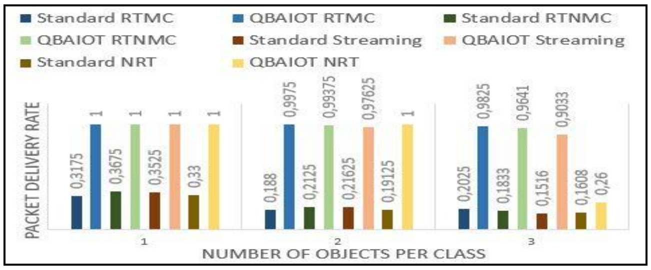

Figure 5. PDR evaluation for different traffic types using QBAIoT and IEEE 802.15.4 Standard

As for the effective data rate, Fig. 6 compares the obtained results using our proposed QBAIoT method and the traditional slotted CSMA/CA of the IEEE 802.15.4.

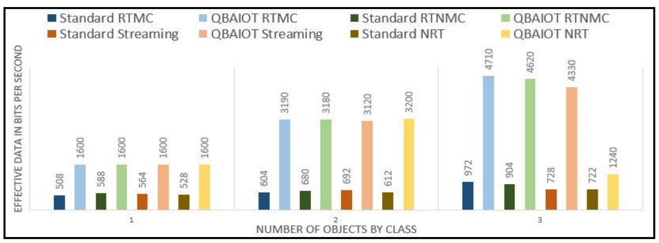

Figure 6. Effective Data Rate evaluation for different traffic types using QBAIoT and IEEE 802.15.4 standard

The obtained results show that QBAIoT allows always better effective data rate than the traditional approach, as the PDR of QBAIoT is always higher.

VI. CONCLUSION

To ensure better user experience in the Internet of Things environment, researchers try to optimize the delivered services while guaranteeing the QoS. Different access technologies could be used in the sensing layer of the IoT architecture. Several of these technologies are based on the IEEE 802.15.4 standard but the latter does not provide any QoS guarantee for the traffic generated by objects using this standard to access the IoT infrastructure. Therefore, we proposed the QBAIoT access method as an enhancement of the IEEE 802.15.4 slotted CSMA/CA mechanism in order to take into consideration QoS requirements of 4 different kinds of QoS traffic classes generated in the IoT environment. We compared our proposed access method to the IEEE 802.15.4 standard and we showed that we obtain better results while using our QoS based access method to guarantee a reduced delay for Real Time traffic, as well as a greater PDR and effective data rate for all QoS classes.

As ongoing work, we are implementing a closed control loop enabling self-management capabilities within an IoT gateway adopting our QBAIoT access method in order to adapt the superframe configuration according to the environment (slots usage variation within each QoS CAP).

ACKNOWLEDGMENT

The regional council of Bourgogne Franche-Comté, supported and funded this research project.

REFERENCES

[1] ITU-T Y.2066, "Next Generation Networks – Frameworks and functional architecture models", 32 pages, 2014

[2] P. Thubert, C. Bormann, L. Toutain, and R. Cragie, "Pv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header", IETF RFC, 37 pages, April 2017

[3] S. Nath, S. Aznabi, N. Islam, A. Faridi, and W. Qarony, "Investigation and Performance Analysis of Some Implemented Features of the ZigBee Protocol and IEEE 802.15.4 Mac Specification", International Journal of Online Engineering (iJOE), vol.13, pp. 14-32, Nov.2017, ISSN: 1861-2121, doi:10.3991/ijoe.v13i01.5984

[4] A. Nordrum, "Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated", *IEEE Spectrum*, August 2016.

[5] P. Mell and T. Grance, "The NIST Definition of Cloud Computing", NIST, 2 pages. version 15, July 2009

[6] A. Banafa, "Definition of fog computing", IBM, August 2014, https://www.ibm.com/blogs/cloud-computing/2014/08/fog-computing/ (Last access 17 March 2018)

[7] IEEE Standard for Local and metropolitan area networks, Low-Rate Wireless Personal Area Networks, IEEE Computer Society, 311 pages, September 2011

[8] ITU-T E.800, "Definitions of terms related to quality of service", 30 pages, 2008

[9] J.Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions", Future Generation Computer Systems, vol. 29, pp. 1645-1660, 2013, doi: 10.1016/j.future.2013.01.010

[10] R. Bhaddurgatte and V. Kumar, "Review: QoS Architecture and Implementations in IoT Environment", Research & Reviews: Journal of Engineering and Technology, ISSN: 2319-9873, pp. 6-12, 2015

[11] Martin Serrano, "OpenIoT D.4.6 Quality of Service (QoS) for IoT services", OpenIoT Consortium, Project Number 287305, 51 pages, 2014

[12] Syarifah Ezdiani, Indrajit S Acharyya, Sivaramakrishnan Sivakumar, and Adnan Al-Anbuky "An IoT Environment for WSN Adaptive QoS", 2015 IEEE International Conference on Data Science and Data Intensive Systems (DSDIS 2015), , 2015, pp. 586-593, ISBN: 978-1-5090-0214-6, doi:10.1109/DSDIS.2015.28

[13] S. Sarode and J. Bakal, "A Slotted CSMA/CA of IEEE 802.15.4 Wireless Sensor Networks: A Priority Approach", International Journal of Computer Trends and Technology (IJCTT), vol. 44, pp. 33-38, Feb. 2017, ISSN: 2231-2803, doi: 10.14445/22312803/IJCTT-V44P106

[14] F. Xia, J. Li, R. Hao, X. Kong, and R. Gao, "Service Differentiated and Adaptive CSMA/CA over IEEE 802.15.4 for Cyber-Physical Systems", The Scientific World Journal, vol. 2013, Article ID 947808, 12 pages, 2013, doi:10.1155/2013/947808

[15] A. Khalil, N. Mbarek, and O. Togni, "Service Level Guarantee Framework for IoT environments", International Conference on Internet of Things and Machine Learning (IML 2017), 2017, ISBN: 978-1-4503-5243-7, doi: 10.1145/3109761.3158393

[16] M. Kirsche, IEEE 802.15.4-Standalone, https://github.com/michaelkirsche/IEEE802154INET-Standalone (Last Access 17 March 2018)