# Adaptive Collision Avoidance Scheduling based on Traffic and Priority for IoT Sensor Networks

Linh T. Nguyen[1], Thu-Hang T. Nguyen[1], Hai-Chau Le[1], Thanh Vinh Vu[2], Chien Trinh Nguyen[1]

[1] Posts and Telecommunications Institute of Technology, Hanoi, Vietnam.

[2] Thai Nguyen University, Thai Nguyen, Vietnam.

Email: linhnt.B17VT215@stu.ptit.edu.vn, {hangntt, chaulh, trinhnc}@ptit.edu.vn, vtvinh@ictu.edu.vn

***Abstract*— Recently, the proliferation of Internet of Things (IoT) and industrial networks poses new challenges and issues on the QoS requirements such as adaptation, real time, reliability, and energy efficiency. It is expected that important or urgent data need to be delivered in real time with higher reliability than ordinary data. To cope with that, we propose an Adaptive Collision Avoidance Scheduling based on Traffic and Priority (ACASTP) for IoT sensor networks. Our proposed solution employs data prioritizing and traffic adaptive scheme at MAC layer to ensure that higher-priority packets have more privileged access to shared channels. We have also evaluated the developed solution performance using numerical experiments on OMNeT++. The obtained results imply that, comparing to conventional approach, i.e., Timeout Multi-priority-based MAC (TMPQ MAC) protocol, the proposed scheduling scheme improves the network performance significantly.**

***Keywords—* *Internet of things, Wireless sensor networks, MAC protocol, contention window, traffic, priority.***

## I. INTRODUCTION

Nowadays, with the rapid growth of Internet of Things (IoT), not only human but also physical and virtual things are interconnected based on low-cost devices, sensors or smart objects which may observe and interact with their internal and external environments [1-3]. It is predicted that IoT's rapid expansion has global economic influence of up to $11.1 trillion per year [4] and up to approximately 75.44 billion connected devices by 2025 [5]. With the ability of sensing, collecting, processing and exchanging of Sensor Nodes (SN) or smart devices, IoT has engaged a variety of applications including forest monitoring, healthcare monitoring, intelligent transportation, smart-wearable systems and automated industrial processes [6] - [7].

The expansion of IoTs is posing some potential challenges and issues on total delay and energy consumption while IoT smart devices are generally energy and computation restricted. In addition, within the data analysis and track estimation over a range of emerging IoT data, there is a need to classify network traffic into various classes in terms of priority levels to enhance the network performance and responsibility. To support multi-priority levels, the wireless medium access control (MAC) should be high throughput, low latency, scalable and resilient to interference [8]. Until now, to the best of our knowledge, cutting-edge MAC protocols rarely meet the emerging demands of IoTs, especially for simultaneously various and multi-classed events. In particular, some kinds of data events such as abnormal/warning data, control data, … may need to transfer to destination in the quickest way. This is because, upon emergency, many devices and sensors will send their data simultaneously to a central node at high rates to assess the situation's severity. Hence, to support a vast number of connected sensors and devices, the development of efficient scheduling mechanisms considering the event priority to ensure various QoS requirements is critical.

Moreover, one of the potential approaches to deal with QoS/priority requirements and reduce system delay and energy consumption in IoT sensor networks is to design a suitable MAC protocol. Besides, intelligent IoT devices and wireless sensor nodes have similar features and characteristics in the networking and medium access protocols (MAC), MAC protocols for IoT can leverage and inherit from existing MAC protocols for WSNs [9]. The importance of MAC protocols related to delay has been a prevalent research topic for both WSNs and IoT recently. In [10], authors proposed an asynchronously duty-cycled MAC protocol for IoT devices, named RIVER-MAC, which reduce idle listening based on magnitude for senders and contention between receiver nodes. In [11], a modification of IEEE 802.11 MAC protocol and an opportunistic channel selection scheme, which better channel utilization and goodput of the system were addressed for cognitive radio and ad hoc sensor networks in IoT. However, aforementioned proposals do not consider the scenario of different QoS/priority-level events occur simultaneously in IoT networks. Besides, the protocol in [12], named QAEE MAC, has been designed to consider two priority levels of the data packet (high or low). However, it still has only two kinds of priority. Furthermore, average delay of higher priority packets subjects to waiting timer that causes extra delay. Authors in [13] presented a noticeable priority-based energy-efficient MAC, named PRIN, which uses two kinds of priorities, can achieve high throughput by making use of priority queues and different processing of packet arrival. But under interference, this proposal is less efficient in terms of throughput and ease than S-MAC and T-MAC. More recently, Timeout Multi-priority-based MAC (TMPQ MAC) [14] is a receiver-initiated MAC protocol provided a synchronized approach and considers QoS and four different packet priorities to reduce average end-to-end delays and prolong network lifetime. Nevertheless, TMPQ-MAC has not pondered the impacts of the contention window in sensor nodes on average packet delay and energy consumption.

There are various schemes are presented for improving MAC protocol. Especially, adjusting the size of the contention window size to enhance the network performance is one of the important solutions. In [15], authors proposed an adaptive contention window backoff mechanism to improve the network performance by adjusting the backoff time according to the number of active STAs in each access category (AC). Authors in [16] presented the MAC Adaptive Contention Window (ACW), which was adjusted according to the node's active queue size and the remaining energy. In this way, the proposed model improves network throughput, reduce MAC overhead and retransmissions.

In this paper, we propose a priority and traffic-based collision avoidance MAC protocol (named as ACASTP MAC protocol) for improving the IoT sensor network performance, in term of packet loss rate, energy consumption and end-to-end delay. Our proposed MAC protocol is based on the combination of the collision-avoidance and the priority-based serving guarantee. We also evaluate the performance of our proposed MAC protocol by using numerical experiments.

## II. Proposed Solution

### A. Network Model

The network model implemented in this paper is equivalent to [17], where sensor nodes are uniformly distributed in a circular of radius and assumed to be placed one hop away from the sink node acted as a receiver exchanging information from sender nodes is located at the center of the circular area. We make the following assumptions about the network model:

(1) Each sender node sends data packets of events at one packet per second with four priority levels.
(2) Each sender knows its traffic ratio of different priority data.
(3) The initial energy of the sensor node has no restriction. The energy of sender nodes is mainly used to listen to the channel, receive data and send data [18].

### B. Contention Window Model

Equivalence to S-MAC [19], one of the original MAC protocols for WSNs and based on a synchronous duty-cycled protocol, our proposed ACASTP MAC protocol employs a duty cycle and duration of active and sleep periods that are fixed, depending on the application requirements. RTS/CTS (Request-To-Send/Clear-To-Send) handshaking mechanisms are implemented to avoid collision, overhearing and hidden terminal issues. When a node wants to send a packet, it will send RTS to the sink node randomly in the contention window, RTS contains the time required for the node completing one round of data transmission (denoted by NAV-Network Allocation Vector value), so based on that, other nodes will be implied that they should sleep during the NAV time; this will help to save other nodes energy. The receiver then replies by a CTS to confirm the following data transmission. The sending node starts data transmission after receiving its CTS packet. The receiver node replies an ACK after receiving the data. Despite the fact that SMAC uses energy efficiently and guarantees relatively low latency due to its small competitive window and sending CTS as soon as first RTS is received, SMAC is lack of prioritized mechanism for many types of data, which means all packets have been treated the same way, that is why packets have the same latency and reliability. So, to accommodate different data transmission requirements, there is a need for data differentiation mechanism to support low delay and high reliability for high priority data. So, in our proposed ACASTP MAC protocol, we inherit the idea of data prioritization of MPQ MAC protocol [20]. We differentiate priority levels based on types of data (for example: urgent, most important, important, and normal). Furthermore, the window will be divided into different parts adaptively based on different traffic priority.

Fig. 1 shows the operation of our ACASTP MAC, in which the information of data priority level from application layer is passed down through the routing layer to the MAC layer. After waking up, the receiver senses the medium for $T_g$ (guarantee time) and it informs its availability to senders by broadcasting wake up beacon. MAC layer of senders could adjust its contention window size and position by their data priority levels and traffic ratio while the receiver MAC layer works the same way to SMAC. The contention window in the figure is virtual because it will close when it receives a successful RTS (with no collision), after that receiver starts sending CTS and waits for data from the success sender. If, for example, there are $M$ senders which sender $i$ and $M$ have data to send, if sender $i$ has chance to send RTS before sender $M$ because sender $i$ has higher priority data than sender $M$, then the RTS of sender $i$ will reach the receiver and it will receive its CTS from the receiver. Then sender $i$ send its data while other senders go to sleep during that data transmission time.

In our proposed scheme, RTS is sent from a sender with collision window varied by data priority level and traffic ratio. If a sender has data, it listens to the medium to check if the medium is clear and send its RTS frame randomly in its resized contention window. If the sender finds the medium is busy, it will sense the medium again. The start time of sending RTS is random in order to avoid collision of the same priority level RTSs from many senders. So, in ACASTP MAC, the RTS (like Tx-Beacon in TMPQ MAC) for data packet with highest priority level will have chance to appear earlier than the other one with lower priority, so that the packet delay of highest priority would be lower than the one of lower priority. By doing so, our protocol will shorten the waiting time of receiving CTS of many senders, which can compare to $T_w$ of TMPQ MAC protocol ($T_w$ is like contention window in SMAC), the earlier sending CTS (like Rx-Beacon in TMPQ MAC) scheme will also help other senders know to avoid sending their frames, and save energy by sleeping during NAV time.

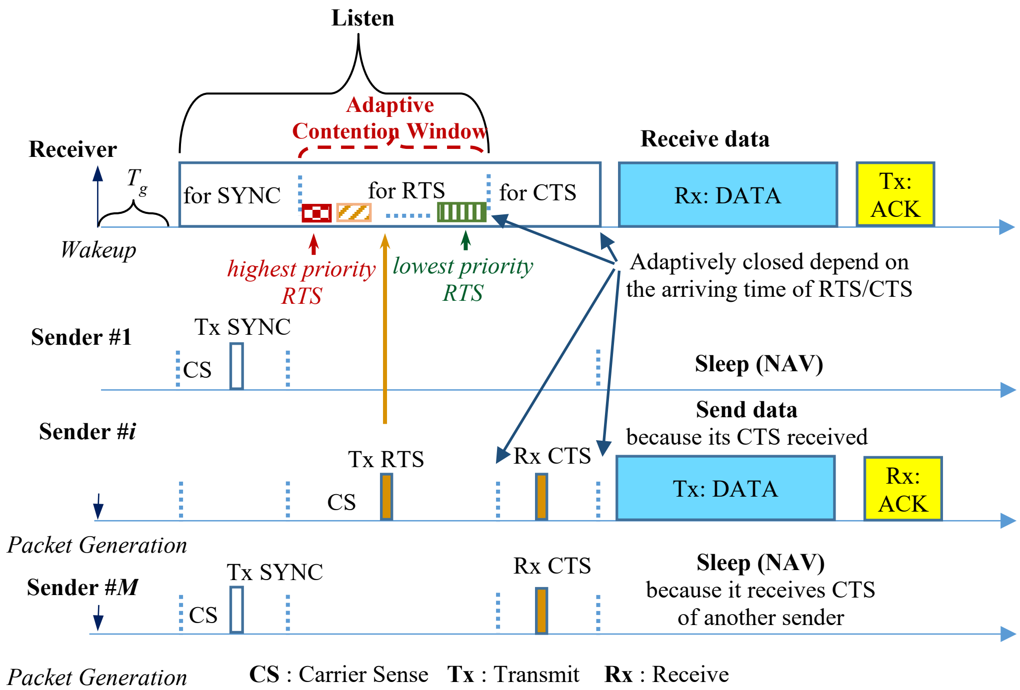


Fig. 1. Description of ACASTP MAC operation

*C. Theory Analysis of Access Delay*

*1) Assumptions and Notations*

In this paper, a network with limited number of senders is considered and one sink as the receiver in the center. All senders and receiver are in each other transmission range. As the selected scenario points to the IoT and industrial applications which do not meet the hidden and exposed terminal problems, therefore, the discussion will be limited to small independent networks like in smart agricultural gardens or smart houses. In addition, the following assumptions and notation are used:

a. A CSMA/CA with contention window for RTS is used for ACASTP MAC while a $p$-persistent CSMA for TxBeacon access is used for TMPQ MAC. Hence, each sender accesses the channel in the idle state with probability 1 for ACASTP MAC or $p$ for TMPQ MAC of sending its RTS frame, where $p$ is $\sum_{i=1}^{M} p = 1$.
b. The number of contending senders is $M$.
c. $N$ priority levels are used, where, probability of a frame which has priority level $j$ is $p_j$. TMPQ MAC is assumed that all types of priority frames have equal probability, that means $p_j = 1/N \; with \; j = 1,2,\ldots,N$. For ACASTP MAC, $p_j$ is changed adaptively with different traffic ratio.
d. The receiver contention window size of ACASTP MAC is denoted as $CW$ and is the same as $T_w$ in TMPQ MAC.
e. The propagation delay is opined to be significantly smaller than the slot time, so, it is neglected [21].
f. The maximum RTS/TxBeacon retransmission value is limited.

*2) RTS/TxBeacon Access Delay Analysis using ACASTP MAC and TMPQ MAC protocols*

Because the difference in delay of different priority packets at the MAC layer depends mostly on the access time to send the RTS [18], in this paper the RTS access delay is basically evaluated. This access delay will be impacted by the RTS sending and accepting scheme, RTS size, the number of competing senders at a time, and the size of the contention window. To quickly grasp the differences between the two MAC protocols, the one sender analysis is presented, then follows the explanation for the case of multi senders.

*a) One sender case*

Assume that there is one sender so there is no collision and the starting time of contention window is $t_{start}$ in Fig. 3. The RTS access delay $t_{accessMAC}$ will be the summarization of $t_{start}$ and duration time from the starting time $t_{start}$ to the time the RTS is accepted by the receiver.

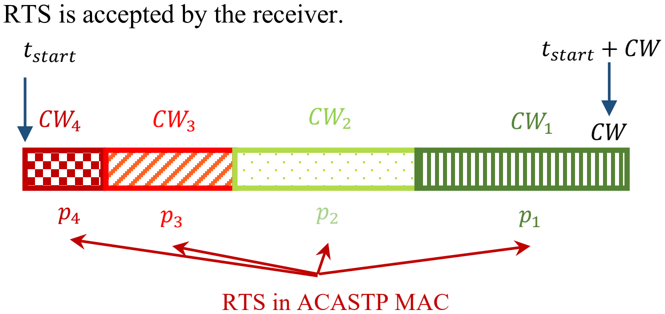


Fig. 2. The RTS arrival time of ACASTP MAC

In practical scenarios, the amount of urgent data is smaller in comparison with other types of data. To evaluate the impacts of the contention window on average delay, we will consider increasing ratio of the traffic from the highest packet priority to the lowest one. Besides, all MAC protocols include multi-priority values assigned for data packets, so RTSs are also treated as different types. Let denote the access delay with priority $p_j$ with $j = 1..4$. In this paper, we consider the RTS arrival time in case of the contention window is divided adaptively based on the ratio of the traffic of different priority packets as shown in Fig. 2. After that, the RTS will be sent randomly inside its resized window.

In TMPQ MAC, RTS with highest priority level ($p_4$) is accepted as it arrived at the receiver while other lower priority is accepted at the closing time of the contention window. Then, in the case of one sender, the numerical expression of TMPQ MAC RTS average access delay $t_{accessTMPQMAC}$ is

$$t_{accessTMPQMAC} = \begin{cases} t_{start} + t_{RTS} & if\ priority\ level\ is\ 4 \\ t_{start} + CW + t_{RTS} & if\ lower\ priority\ level \end{cases} \quad (1)$$

Correspondingly, regarding one sender, the numerical expression of ACASTP MAC RTS average access delay $t_{accessACASTP}$ is

$$t_{accessACASTP} = t_{start} + \frac{CW}{2}\left[\sum_{4-i}^{4} CW_i\right] + t_{RTS} \quad (2)$$

where $R_i$ is the raito of $p_j$ priority traffic with $j = 1..4$.

*b) Multi-sender case*

Assume that there are multiple senders have data to send. But, there is one receiver only, one RTS is accepted in one cycle only, then another RTS will be delay to the next cycle, and so on. Following that, RTS contention may be occurred, so even the soonest RTS may randomly appear earlier, the average delay of RTS access delay will be longer.

In TMPQ MAC, when the number of sending nodes is high, for example $M$, the value $p_M = 1 / M$ will be relatively small causes RTS is not sent immediately due to the sender having to sow and decides to send or not, so even highest priority RTS will not be sent as soon as the window starts, and as larger as the number of senders as higher the access delay. In addition, the RTS of lower priority will be sorted at the end of the contention window and only one RTS with higher priority is accepted by that time, so the delay of lower priority RTS will be even worse.

As per the simplified analysis above, the highest priority RTS of ACASTP MAC will have lowest delay, especially it will be higher when there are many senders sending data. So to get the lower delay of priority data in case of multiple priority events, the ACASTP MAC will be the best choice.

## III. Experimental Results and Discussion

In this section, we evaluate the performance of IoT sensor networks utilizing our proposed solution, ACASTP MAC protocol with adaptive contention window sizes, by using numerical experiments based on Castalia 3.3 [22] and OMNeT++ 4.6 [23] with CC2420 transceivers [24]. Table I summarizes the simulation parameters applied. We also consider the following three performance parameters:

- *Average packet delay:* With the same ratio of traffic, we compare how much time packets start generating data to the time it reaches the sink.
- *Average energy consumption:* The average energy consumption per bit.
- *Average packet loss rate:* It is a ratio of the total number of packets that was lost at the sink to the total packets sent from all sensor nodes.

TABLE I. Simulation Parameters

| Parameter | Value |
|---|---|
| Sensor area | 10m x 10m |
| Sender nodes | 1 to 15 |
| Bandwidth | 250kb/s |
| Radio | CC2420 |
| SYN size | 6 bytes |
| RTS/Tx-Beacon size | 13/14 bytes |
| CTS/Rx-Beacon size | 13 bytes |
| MAC overhead | 11 bytes |
| Listen interval | 17ms |
| RTS/Tx Beacon retransmission | 0 to 9 |
| Packet rate | 1 packet/s |
| Application header | 5 bytes |
| DATA packet size | 28 bytes |
| ACK packet size | 11 bytes |
| CCA Check Delay | 0.128ms |
| Physical frame overhead | 6 bytes |
| $T_g$ | 6.7ms |
| $CW/T_w$ | 10ms |

On the other hand, Table II shows the ratio of multi-priority traffic and respective values of contention window in ACASTP MAC while TMPQ MAC makes use of the $T_w$ timer.

TABLE II. The percentages of different priority traffic

| Priority | $p_4$ | $p_3$ | $p_2$ | $p_1$ |
|---|---|---|---|---|
| The ratio of traffic (TMPQ MAC and ACASTP MAC) | 15% | 20% | 30% | 35% |

### A. Result Analyses

#### 1) Average end-to-end delay of priority packets

The performance of TMPQ-MAC and ACASTP-MAC protocols under the same traffic load is analyzed in Fig. 3. The average packet delay based on different priority levels in TMPQ MAC is higher than in ACASTP MAC. TMPQ MAC uses a $T_w$ timer to receive TxBeacon at the receiver node and check the priority field. If receiver receives $p_4$ TxBeacon, it then stops the timer and sends back CTS for sender to send $p_4$ packet, other senders must wait until the next frame to send their packets. If receiver does not receive $p_4$ TxBeacon, it will have to wait until $T_w$ timer expires, and then choose the highest priority RTS to confirm. ACASTP MAC uses adaptive contention window which is closed as soon as receiver receives the first RTS which is sent within its priority window, then the CTS is sent earlier lead to shorter packet delay.

In addition, in TMPQ MAC, the higher number of sensor nodes, the smaller the probability $p$ is and sensor nodes have to sow in a long time before they can send their packets. As a result, highest priority packets are sent first, then lower priority packets will be sent one after another, so, the delay differences are enlarged more according to the priority levels. There is a gradual growth in the average end-to-end packet delay of each type of data when the number of sending nodes increases.

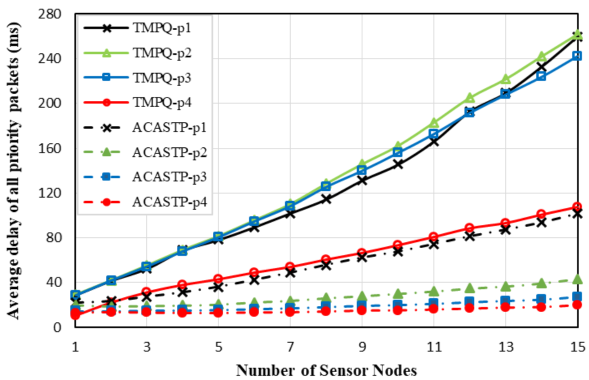


Fig. 3. Analysis of the average multi-priority packet delays

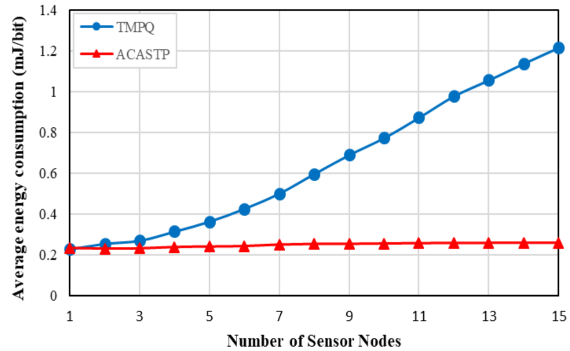


Fig. 5. Analysis of average energy consumption in mJ/bit of TMPQ MAC and ACASTP MAC protocols

According to the above discussion, Fig. 4 is obtained, the average end-to-end delay of the highest priority packets is reduced significantly because of the adaptive setting contention window values based on the ratio of four-priority traffic in proposed ACASTP MAC protocol compared to TMPQ MAC protocol.

*2) Average energy consumption per bit*

Fig. 5 shows the average energy consumption in *mJ* per bit for the successful transmission with TMPQ-MAC and ACASTP-MAC protocols. From the simulation result, the average consumed energy of the proposed ACASTP MAC is sharply lower than that of TMPQ MAC, and the energy consumption difference becomes more obvious as the number of nodes increases. The ACASTP has the advantages of adaptive contention window which is closed sooner than $T_w$, then only one sender could send its packet while others sleep till next frame. In TMPQ, the sender has to sow and wait until it can send its TxBeacon, then if there is no $p_4$ packet, all senders will have to wake up till $T_w$ expired. So, the wake-up time of TMPQ nodes is longer compare to ACASTP, that why the spending amount of energy for wakeup state is much more.

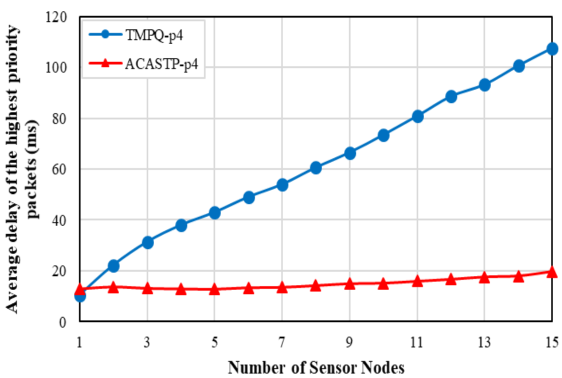


Fig. 4. Comparison of the average highest priority packet delay (p4)

Furthermore, when the higher the number of senders increases, the higher the level of competition, and more energy consumption becomes. In this case, ACASTP has more advantages over TMPQ with adaptive window for each sending node, the more nodes, the smaller the total congestion window time of each node compared to TMPQ.

*3) Average packet loss rate*

Fig. 6 shows the comparison of the average packet loss rate of TMPQ-MAC and ACASTP-MAC protocols. We can see that the contention window size does not affect only the average delay and energy consumption but also the packet loss rate.

The larger contention window of TMPQ in case of many senders lead to serious TxBeacon contention situation, not only in the first frame but also last to the next frames. Then each time there is contention, senders have to retransmit and waste more energy. If the number of retransmission time is limited, then the TxBeacon may be lost and the data packet cannot be delivered.

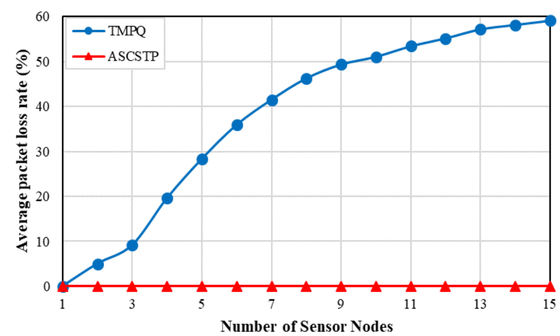


Fig. 6. Average packet loss rate gained by TMPQ MAC and ACASTP

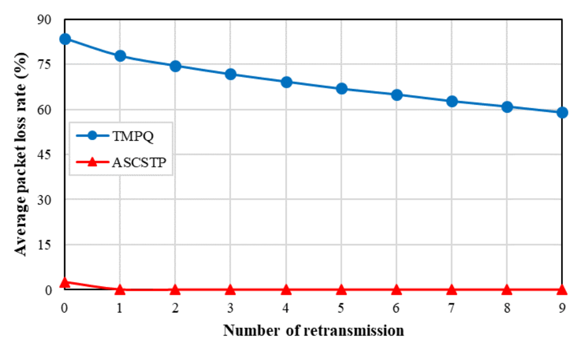


Fig. 7. Impact of the retransmission number

As shown in Fig. 7, in case of 15 sensor nodes, we compare the average packet loss rate of TMPQ-MAC and

ACASTP-MAC protocols with the retransmission time ranges from 0 to 9. The results demonstrate that while TMPQ-MAC protocol requires more retransmission that may cause higher delay and energy consumption, our proposed ACASTP protocol only needs one time retransmission to deliver packets successfully thanks to the flexible and adaptive window size allocation.

## IV. Conclusion and Future Work

In this paper, we have successfully proposed an ACASTP protocol to improve the IoT sensor network performance, in terms of delay, energy consumption and packet loss rate. Our developed ACASTP protocol takes advantages of a duty cycle, duration of active and sleep periods with the RTS/CTS handshaking mechanism and data prioritization scheme to adjust adaptively the contention window and transfer the highest priority data or urgent data in the medium access control (MAC) protocol. Numerical experiments show that our proposed ACASTP algorithm outperforms the notable conventional solution, TMPQ. It not only effectively reduces the average delay of data transmission, but also significantly enhances the packet transmission success rate of the network, and save more energy.

## References


[1] T. Yu, X. Wang and A. Shami, "Recursive Principal Component Analysis-Based Data Outlier Detection and Sensor Data Aggregation in IoT Systems," in IEEE Internet of Things Journal, vol. 4, no. 6, pp. 2207-2216, Dec. 2017, doi: 10.1109/JIOT.2017.2756025.

[2] Yaqoob, Adeel, Muhammad Adeel Ashraf, Faria Ferooz, Ahmad Hassan Butt, and Yaser Daanial Khan. "WSN Operating Systems for Internet of Things (IoT): A Survey." In 2019 International Conference on Innovative Computing (ICIC), pp. 1-7. IEEE, 2019.

[3] Bou Dargham, Nadine et al. "Efficient Hybrid Emergency Aware MAC Protocol for Wireless Body Sensor Networks." Sensors (Basel, Switzerland) vol. 18,10 3572. 22 Oct. 2018, doi:10.3390/s18103572.

[4] McKinsey Global Institute, "The Internet of Things: Mapping the Value beyond the Hype EXECUTIVE SUMMARY," McKinsey Co., no. June, pp. 1–18, 2015, [Online]. Available: www.mckinsey.com/mgi.

[5] Alam, Tanweer, A Reliable Communication Framework and its Use in Internet of Things (Iot) (May 1, 2018). CSEIT1835111 | Received : 10 May 2018 | Accepted : 25 May 2018 | May-June-2018 [ (3)5 : 450-456, Available at SSRN: https://ssrn.com/abstract=3619450.

[6] F. John Dian, R. Vahidnia and A. Rahmati, "Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey," in IEEE Access, vol. 8, pp. 69200-69211, 2020, doi: 10.1109/ACCESS.2020.2986329.

[7] F. Cui, "Deployment and integration of smart sensors with IoT devices detecting fire disasters in huge forest environment," Comput. Commun., vol. 150, pp. 818–827, 2020.

[8] P. Di Marco, G. Athanasiou, P. V. Mekikis, and C. Fischione, "MAC-aware routing metrics for the internet of things," Comput. Commun., vol. 74, pp. 77–86, 2016.

[9] A. Kumar, M. Zhao, K. Wong, Y. L. Guan and P. H. J. Chong, "A Comprehensive Study of IoT and WSN MAC Protocols: Research Issues, Challenges and Opportunities," in IEEE Access, vol. 6, pp. 76228-76262, 2018, doi: 10.1109/ACCESS.2018.2883391.

[10] M. L. Wymore and D. Qiao, "RIVER-MAC: A Receiver-Initiated Asynchronously Duty-Cycled MAC Protocol for the Internet of Things," 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 2019, pp. 860-869, doi: 10.1109/COMPSAC.2019.00126.

[11] Y. Bin Zikria, F. Ishmanov, M. K. Afzal, S. W. Kim, S. Y. Nam, and H. Yu, "Opportunistic channel selection MAC protocol for cognitive radio ad hoc sensor networks in the internet of things," Sustain. Comput. Informatics Syst., vol. 18, pp. 112–120, 2018.

[12] Kim S.C., Jeon J.H., Park H.J. (2012) QoS Aware Energy-Efficient (QAEE) MAC Protocol for Energy Harvesting Wireless Sensor Networks. In: Lee G., Howard D., Kang J.J., Ślęzak D. (eds) Convergence and Hybrid Information Technology. ICHIT 2012. Lecture Notes in Computer Science, vol 7425. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-32645-5_6.

[13] A. K. Subramanian and I. Paramasivam, "PRIN: A Priority-Based Energy Efficient MAC Protocol for Wireless Sensor Networks Varying the Sample Inter-Arrival Time," Wirel. Pers. Commun., vol. 92, no. 3, pp. 863–881, 2017.

[14] A. N. Sakib, M. Drieberg and A. A. Aziz, "Energy-Efficient Synchronous MAC Protocol based on QoS and Multi-priority for Wireless Sensor Networks," 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2021, pp. 347-352, doi: 10.1109/ISCAIE51753.2021.9431806.

[15] I. Syed, S. Shin, B. Roh and M. Adnan, "Performance Improvement of QoS-Enabled WLANs Using Adaptive Contention Window Backoff Algorithm," in IEEE Systems Journal, vol. 12, no. 4, pp. 3260-3270, Dec. 2018, doi: 10.1109/JSYST.2017.2694859..

[16] R. M. Bhavadharini, S. Karthik, N. Karthikeyan, Anand Paul, "Wireless Networking Performance in IoT Using Adaptive Contention Window", Wireless Communications and Mobile Computing, vol. 2018, Article ID 7248040, 9 pages, 2018. https://doi.org/10.1155/2018/7248040.

[17] N. T. T. Hang, N. C. Trinh, N. T. Ban, M. Raza and H. X. Nguyen, "Delay and Reliability Analysis of p-Persistent Carrier Sense Multiple Access for Multi-Event Industrial Wireless Sensor Networks," in IEEE Sensors Journal, vol. 20, no. 20, pp. 12402-12414, 15 Oct.15, 2020, doi: 10.1109/JSEN.2020.3000270.

[18] F. Li, G. Huang, Q. Yang and M. Xie, "Adaptive Contention Window MAC Protocol in a Global View for Emerging Trends Networks," in IEEE Access, vol. 9, pp. 18402-18423, 2021, doi: 10.1109/ACCESS.2021.3054015.

[19] Wei Ye, J. Heidemann and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," in IEEE/ACM Transactions on Networking, vol. 12, no. 3, pp. 493-506, June 2004, doi: 10.1109/TNET.2004.828953.

[20] S. Sarang, M. Drieberg and A. Awang, "Multi-priority based QoS MAC protocol for wireless sensor networks," *2017 7th IEEE International Conference on System Engineering and Technology (ICSET)*, Shah Alam, 2017, pp. 54-58, doi: 10.1109/ICSEngT.2017.8123420

[21] K. Voulgaris, A. Gkelias, I. Ashraf, M. Dohler and A. H. Aghvami, "Throughput Analysis of Wireless CSMA/CD for a Finite User Population," IEEE Vehicular Technology Conference, 2006, pp. 1-5, doi: 10.1109/VTCF.2006.443.

[22] T. Boulis, Castalia Version 3.3 Master. Access: August 20, 2020. [Online]. Available: https://github.com/boulis/Castalia/releases/tag/3.3

[23] A. Varga, OMNeT++ user manual version 4.6, OpenSim Ltd, 2014. Accessed on: July 9, 2021, [Online] Available: http://www.omnetpp.org

[24] Texas Instruments. "CC2420 single-chip 2.4 GHz RF transceiver," Accessed on: August 20, 2020, [Online] Available: http://www.ti.com/lit/ds/symlink/cc2420.pdf