LUND UNIVERSITY

LUPMAC: A cross-layer MAC technique to improve the age of information over dense WLANs

Franco, Antonio; Fitzgerald, Emma; Landfeldt, Björn; Pappas, Nikolaos; Angelakis, Vangelis

LUPMAC: A Cross-Layer MAC Technique to Improve the Age of Information Over Dense WLANs

Antonio Franco*, Emma Fitzgerald*, Björn Landfeldt*, Nikolaos Pappas†, Vangelis Angelakis†

*Department of Electrical and Information Technology, Lund University, Lund, Sweden

{antonio.franco, emma.fitzgerald, bjorn.landfeldt}@eit.lth.se

†Department of Science and Technology, Linköping University, Norrköping SE-60174, Sweden

{nikolaos.pappas, vangelis.angelakis}@liu.se

Abstract—**Age of Information (AoI) is a relatively new metric introduced to capture the freshness of a particular piece of information. While throughput and delay measurements are widely studied in the context of dense IEEE 802.11 Wireless LANs (WLANs), little is known in the literature about the AoI in this context. In this work we study the effects on the average AoI and its variance when a sensor node is immersed in a dense IEEE 802.11 WLAN. We also introduce a new cross layer MAC technique, called Latest UPdate MAC (LUPMAC), aimed at modifying the existing IEEE 802.11 in order to minimize the average AoI at the receiver end. This technique lets the MAC layer keep only the most up to date packets of a particular piece of information in the buffer. We show, through simulation, that this technique achieves significant advantages in the case of a congested dense IEEE 802.11 WLAN, and it is resilient to changes in the variance of the total network delay.**

Index Terms—**Age of information, Dense WLANs, IEEE 802.11.**

I. INTRODUCTION

The concept of the Age of Information (AoI) was first introduced in [1], and then formalized in [2]. This new metric answers the question: how fresh is that particular information stored at the receiver? It is different from the delay, since it includes the time from when a destination has received the last update about a particular piece of information (e.g. the temperature, the water flow/level etc.) from a source. It also has a broader scope than the delay, since it measures a quality of a particular piece of information not a quality of the individual packets themselves.

In some sensor applications, only the most updated measurement of a particular piece of information is relevant, e.g. the current water level in a sewer pipe in order to ensure it does not exceed a given threshold. In this sense, the AoI metric is of crucial importance. Especially in the context of the new paradigm of the Internet of Things (IoT), or the Smart City paradigm [3], a typical application scenario might be sensor

nodes continuously measuring and sending data, using a dense IEEE 802.11 Wireless Local Area Network (WLAN) shared amongst numerous other devices. For example, the sensor node might be interested in uploading the measured information to a remote unit, for storing or further processing. If the remote server is only interested in the freshest possible piece of the information sent by the sensor node, it is interested in the sensor node trying to minimize the AoI at the receiver.

In this work we will study a scenario where a sensor node is immersed in a dense IEEE 802.11 WLAN, where a number of devices are subscribed. It tries to send information to a remote destination. Dense WLANs are a specifical scenario that will be covered in the forthcoming IEEE 802.11ax HEW (High Efficiency WiFi) standard [4]. The IEEE 802.11ah standard is also specifically designed for the IoT [5]. In this standard, an Access Point (AP) can cover up to 1 km in range, and is possible to foresee that overlapping networks with hundreds of devices would not be uncommon. Devices will have to compete for the channel with possibly hundreds of other devices, with a very heteroegeneous population of traffic patterns. For example, there could be devices trying to offload traffic from the existing cellular infrastructure, futher congesting existing IEEE 802.11 WLANs, as in the 5G HETereogeneous NETwork (HETNET) paradigm [6]. Competing with numerous devices degrades both throughput and delay performance, due to the increasing number of collisions, and in case of traffic burstiness, increases the idle time [7]. In case also of a high number of small frames, it deteriorates further [8]. The effects on the AoI are, however, not entirely clear.

In this paper, we extend the work in [1], [9] with a more practical implementation by introducing a new cross layer approach between the application layer and the MAC layer, called Latest UPdate MAC (LUPMAC), aimed at modifying the existing IEEE 802.11 in order to minimize the average AoI at the receiver end. We let the MAC know about the "freshness" of a packet received from the application layer, along with the particular application that generated it, in order to develop a strategy to minimize the AoI at the receiver. Briefly, it will always try to send the packets carrying the

freshest update of that particular information, without trying to transmit (or re-transmit) older packets.

The remainder of this paper is organized as follows. In Section II we will present an overview of the related work. In Section III we describe the concept of Age of Information. In Section IV we describe LUPMAC. In Section V we describe the scenario under test. In Section VI we present our simulation results and in Section VII we conclude the paper.

II. RELATED WORK

The Age of Information in IEEE 802.11 systems was first addressed in [1]. The authors study the age of information in a vehicular network (VANET) via simulation and with a VANET testbed. In their scenario, each vehicle acts as a node. Each node beacons a particular piece of information to nearby vehicles, and it is interested in the other vehicles having the most up to date piece of that information. Each node broadcasts its information, so no acknowledgements are involved. The authors introduce a cross layer MAC technique called "Latest state Out" (LO), in which the application sensing informations fills the packet at the front of the MAC buffer with the latest available piece of information whenever the opportunity of transmitting a frame arises. They show how this technique efficiently minimizes the average AoI in all the nodes in the VANET. They also show that using the optimal Contention Window (CW) from the Bianchi model [10] the average AoI is further minimized. They then show how neither maximizing the throughput nor minimizing the delay automatically minimizes the average AoI. Finally they introduce a cross-layer rate control mechanism that works with a normal FIFO queue and no CW adaptation in order to minimize the average AoI at the nodes.

Their work differs from the work carried out in this paper, since it studies a vehicular network, while we study a dense IEEE 802.11 WLAN of static nodes; we are interested in minimizing the AoI in a remote server instead of distributing the information to a set of nodes in the same network. Also they do not address the problem of other contenders (i.e. other devices trying to access the same wireless channel) in the network. Additionally, they are broadcasting the information, thus using only the first CW, not retrying to send the frame in case of a missing acknowledgment. Finally, in our work the MAC layer should be aware only of the application that generated the packet and the packet's age, while in LO the MAC layer should signal the application whenever a transmission opportunity arises. In our work also, if the packets are sent by the application in order, the MAC layer will automatically infer the new packet is the freshest, thus not even needing an additional field with the packet's age. The proposed LO technique is impractical. The time needed for the MAC layer to signal the application when it is ready to transmit, and then wait for the application layer to fill the MAC buffer is bigger than one IEEE 802.11 slot time ($\sim 10\mu s$), that is the time granularity in an IEEE 802.11 MAC. In addition, with this approach, the application must be allowed to write in the MAC buffer. This is in most of the cases, impractical. In

short, this approach requires very close coupling between the MAC and the application that is both difficult and undesirable in practice. Finally, we will not use the optimal CW from the Bianchi's model, since it is not possible in current hardware to change it at run time [10].

In [2], [9], [11]–[13] the authors study the AoI in different simple queuing systems with multiple classes of service, modeling the channel as a single server. For example, in [2] the authors derive a lower bound for the AoI given any service distribution in a simple queuing system with only one server. In [12] the authors study the minimization of the AoI under energy constraints, particularly a sensor that harvests energy from the environment via numerical simulation. While important properties of the AoI are derived, the effects on the AoI in a real life scenario such as a dense IEEE 802.11 WLAN are not investigated.

The only other study that uses a real network scenario in order to study the AoI, as far as the authors are aware, is [14]; there the authors study the AoI in an emulated WLAN with 2 nodes and compare their results with the theoretical results for various simple queuing systems. The study focuses on a small WLAN, and an IEEE 802.11 stack is not used, whereas we consider a dense WLAN with many more nodes and conduct simulations using a full 802.11 implementation.

As a final note, our cross-layer MAC technique is also a continuation of the work in [9], where the authors study the AoI in a system with N sources, a single queue and a delay channel. They introduce a new queuing discipline based on the age of information. It only holds the freshest packets of each class of information in the queue. On the other hand, the authors study the AoI in an abstract queueing system with N sources, where we make use of a full 802.11 implementation.

III. AGE OF INFORMATION

We will now give an overview of the concept of Age of Information. Consider a transmitter sensing and sending updates of the information I over a channel to a receiver. The receiver is interested only in the freshest update of information I. An example curve of the age of information I over time is depicted in Fig. 1.

Assume a packet with the desired information I is generated at time t_{i-1} s from a a source sensing that information. The receiver receives it at time t'_{i-1} s. The packet will then have an age of $\epsilon_{i-1} = t'_{i-1} - t_{i-1}$ s, so the age of the information I will be at that time ϵ_{i-1} s. Then, if it is not receiving new packets, the AoI will increase over time with slope 1. The next packet carrying the updated information I is generated from the transmitter at time t_i s. It is received at time t'_i s. The age of that packet would then be $\epsilon_i = t'_i - t_i$ s. If this packet is fresher than the current AoI (i.e. $\epsilon_i < t'_i - t'_{i-1} + \epsilon_{i-1}$) then the AoI will jump down to ϵ_i seconds, otherwise it will continue increasing. The AoI will continue to have this characteristic sawtooth behaviour, and it is possible to reconstruct its curve by interpolating between the various samples when packets are received. Then it is possible to reconstruct various metrics; for example, it is possible to reconstruct the average AoI by
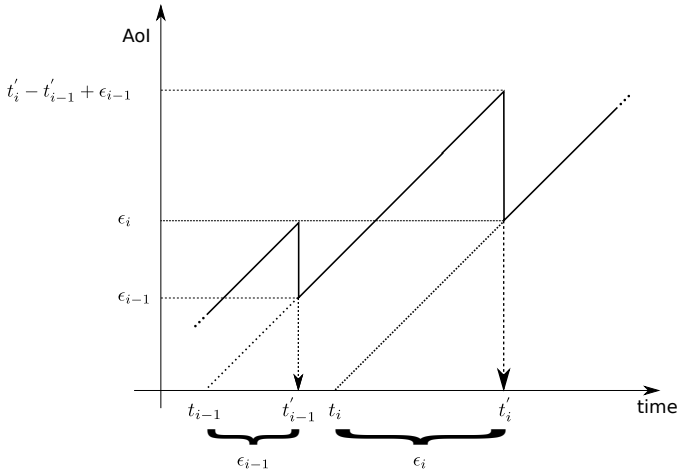
Fig. 1. Example of the Age of Information over time at the end of a receiver.

calculating the integral over time of the curve as a sum of trapezoids and dividing over the elapsed time [11].

In our work, in order to avoid the so-called catastrophic cancellation in the computation of the variance of the AoI, instead of computing the square sum of the trapezoids forming the AoI curve, we compute the average AoI as a running weighted mean, and the AoI variance as a running weighted variance [15].

IV. LATEST UPDATE MAC

We extend the work in [1] and [9] with a more practical implementation of their algorithm, in order to apply a more advanced cross layer approach in an IEEE 802.11 MAC. The procedure is summarized in Algorithm 1.

Algorithm 1 The LUPMAC algorithm.

1: **on event** p' comes from the network layer **do**
2:      $n \leftarrow 0$
3:      **for all** $p \in \mathcal{P}$ **do**
4:          **if** $p.id == p'.id \bigwedge p'.age < p.age \bigwedge n < 2$ **then**
5:              Substitute p with a copy of p'
6:              $n \leftarrow n + 1$
7:      **if** n == 0 **then**
8:          Append p' at the end of $\mathcal{P}$
9:      **else if** n==1 $\bigwedge$ p' is at the front of $\mathcal{P}$ **then**
10:          Append p' at the end of $\mathcal{P}$
11: **on event** ACK received upon transmission of p' **do**
12:      **for all** $p \in \mathcal{P}$ **do**
13:          **if** $p.id == p'.id$ **then**
14:              remove p from $\mathcal{P}$

The MAC layer is aware of the time a packet is generated in the upper layer. If we assume the sources sending the respective pieces of information do not scramble the order of the generated packets, LUPMAC can simply assume the newest packets from the source are also the freshest. The applications running in a sensor all map one-to-one to an information source, and have an ID. The ID thus identifies one information stream. This ID is stamped into the packet at generation time, for example in a field in the header of the network packet. When a new packet p' arrives from the upper layer, the MAC inspects the packets in the transmission buffer $\mathcal{P}$, including the packet in backoff (i.e. the one at the front of the buffer queue), to check if there is one that has the same ID as the newly arrived packet. We call this subset $\mathcal{P}_i$, where i is the source ID. Then, the MAC checks each packet $p \in \mathcal{P}_i$; if p is older than p', it is substituted with a copy of p'.

In the IEEE 802.11 standards the access mechanism is the so-called Distributed Coordination Function (DCF). A frame (that encapsulates a packet) waits a random time before being transmitted. A frame in this state is said to be in "backoff". If a collision occurs after a backoff period, the frame goes again into the backoff state, with a longer period to wait (on average). After a number of retransmissions, 7 in the current basic access mechanism, the frame is dropped. In case of an heavily loaded network, there is always a chance that the packet in front of $\mathcal{P}$ has already been into several retransmissions. So the chance for this particular packet to be dropped is higher, with negative effects on the AoI at the receiver end. In order not to have a newer packet at the last stage of the backoff be thus penalized, if the only substituted packet is the one currently in backoff, a copy is appended at the end of $\mathcal{P}$. Also, in order to not have too many packets of a particular source in $\mathcal{P}$, only two copies of a packet from a particular source are allowed in the buffer. If there are no packets substituted, p' is appended at the end of $\mathcal{P}$.

In order not to transmit multiple copies of the same piece of information, upon the reception of an ACK for p' (i.e. p' is succesfully transmitted), LUPMAC will delete every packet in $\mathcal{P}$ having the same ID as p'.

It is important to point out that LUPMAC is not doing deep packet inspection in order to substitute or remove packets in the MAC buffer. The application ID could be inserted in the packet header in the application layer, and then propagated all the way to the MAC layer in a field in the header. It is also unreasonable for applications in the sensor node to scramble the order of the generated packets, so LUPMAC will just infer the freshness of the piece of information contained in the packet by the time it is received from the upper layer, i.e. the latest received packet is the freshest.

V. SCENARIO DESCRIPTION

The scenario considered in our work is depicted in Fig. 2. It models a sensor node immersed in a dense IEEE 802.11 WLAN with no hidden nodes, in order to better inspect the effects of LUPMAC on the average AoI. Such scenarios occur, for example, in a city, where a public hotspot serves a large number of users, and sensor nodes, such as smartgrid sensors or water flow sensors, use the existing infrastructure to send information remotely. Another example that could be modeled by this scenario is an industrial one, where sensor nodes have to send status updates about machinery to a central server while competing for the channel with other devices.
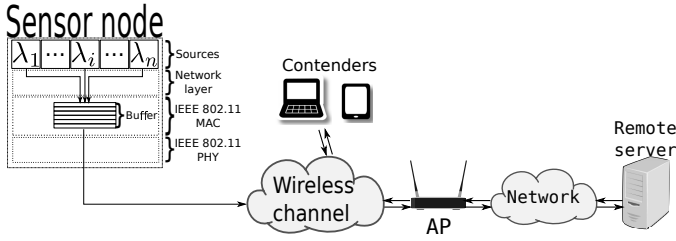
Fig. 2. Scenario for our simulation studies. This diagram also utilises the following third party images: [16]–[19].

These sensor nodes send information fairly frequently, and the last reading of this information is what counts, i.e. we are interested in minimizing the average AoI.

In our studied scenario, a sensor node is sending various information streams to a remote server. A packet from the sensor has to be sent first via the wireless channel, then it is routed via a normal fixed-link connection, labelled as "Network" in the diagram, to a remote server. A sensor node is formed by an application layer stack, where there are a number of applications, labeled in the figure as sources, each one of those measures a particular piece of information, and sends updates about their own information to a remote server. In case LUPMAC is used, the applications running in the application layer insert their unique ID in a field in the packet header, that is propagated all the way to the MAC layer, in order to let LUPMAC know which application generated that particular packet. Then, there is a network layer, and then an IEEE 802.11 MAC, that holds the packets generated by the various sources in its buffer. Next there is an IEEE 802.11 PHY to access the channel.

A sensor node is competing for the channel with a number of contenders, each one requesting content from the remote server. The contenders send requests to the remote server, then the server fullfills those requests by sending back content to them. They send relatively small packets for the request, and receive packets of various sizes back. This models a variety of users that stream content, offload traffic using the IEEE 802.11 WLAN or simply browse the web.

The remote link introduces a delay according to a random distribution. Since the metro (or backbone) part of the network is usually reliable, at least in big cities, we will assume the remote link to be reliable, so no packet is dropped there. This models, for example, a routed path to a remote destination via the internet. On the other end, we tested the reliability of LUPMAC by measuring the average AoI both with high and low variance in the network part of the simulation.

VI. RESULTS

We have conducted our simulation studies using OMNeT++ and the INET package [20]. The parameters used in the simulations are summarized in Table I.

All the plots are presented with 95% confidence, allowing for a sufficient warm-up period before taking measurements. The scenario simulated is the one described in Section V.

TABLE I
PARAMETERS AND CONFIGURATION VALUES.

	Parameter	Value
Physical	Frequency	2.4 GHz
	Noise Power	-110 dBm
	SINR Threshold	4 dB
	Transmission Power	20 mW
	Reception Threshold	-85 dBm
	Data Rate	54 Mbps
	Slot Time (σ)	9 μs
Scenario	Scenario dimensions	600 x 400 m
	Channel model	Free space
	Free space exponent	2
App	number of sensor nodes	1
	number of contenders	variable
	information generation (sensors only)	every 0.1 s
	request generation (contenders only)	$\sim \exp\{0.01\}$ s
	Packet length (sensors)	10 bytes
	Packet length (contenders)	$\sim \exp\{10\}$ bytes
	Requested packet length (contenders)	$\sim U(14, 1000)$ bytes[1]
MAC	type	802.11g (AC1)
	buffer length (packets)	100

There is a sensor node uploading data to a remote server. It has an application (source) running, taking measurements. The sensor node is using an IEEE 802.11g WLAN with a number of contenders varying from 0 to 60. It is uploading small packets deterministically at a fairly slow rate (10 pk/s). The contenders are issuing requests to a remote server with exponentially distributed interarrival times, with an average rate of 100 pk/s, in order to increase the traffic load on the WLAN and congest it. The request packets are small (10 bytes on average, exponentially distributed), while the reply packets are uniformly distributed from small packets (14 bytes, a control frame) to big packets (1000 bytes). The delay on the wire connecting the access point to the remote server is considered to be a reliable metro/backbone connection. The average roundtrip time is however considered to be challenging with respect to VoIP traffic (150ms).

The average AoI and its variance are measured with an increasing number of contenders in the case that the delay has narrow variance, i.e. the one-way delay is uniformly distributed between 74ms and 76ms (so as to have an average roundtrip time of 150ms) with LUPMAC or the standard IEEE 802.11 FIFO approach. Then it is tested in the case it has a large variance, i.e. the one-way delay is uniformly distributed between 0s and 150ms (still an average round trip time of 150ms) with LUPMAC or the standard IEEE 802.11 FIFO approach. In Fig. 3 the AoI for all the cases is presented.

As we can see from Fig. 3, the difference between high and narrow variance in the standard case (i.e. IEEE 802.11 FIFO)

---

[1] $U(a, b)$ is the uniform distribution between a and b.
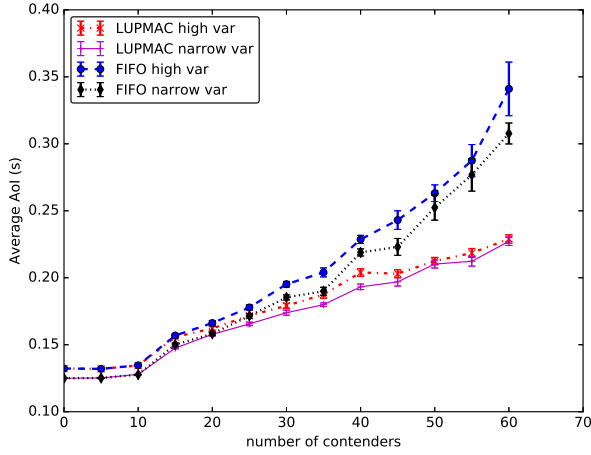
(a)
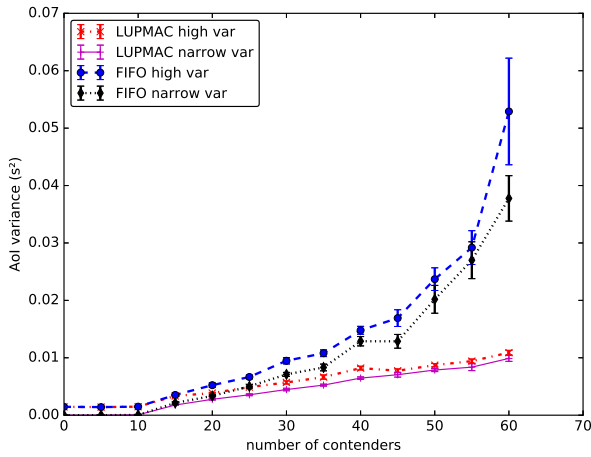
(b)

Fig. 3. Average Age of Information (a) and variance (b) measured at the destination with narrow variance on the wire delay and high variance with LUPMAC or FIFO.

is quite small, only a fraction of the average AoI even with a totally saturated network with 60 contenders. In both cases the average AoI grows almost two tenths of a second from 10 to 60 contenders. This is quite a high increase, considering that the source on the sensor node is generating one packet every tenth of a second.

Then, we tested LUPMAC (introduced in Section IV). As we can see, LUPMAC significantly improves the AoI in case of a highly saturated scenario (when the number of contenders grows over 30), with an improvement of almost a tenth of a second with 60 contenders on the average AoI. Also, the AoI appears more stable, as the variance grows much more slowly when LUPMAC is on. The improvement over the average AoI is extremely good, considering that the source on the sensor node generates one packet each tenth of a second. The improvement can be explained by the number of replaced packets in the MAC buffer when LUPMAC is used. In Fig. 4
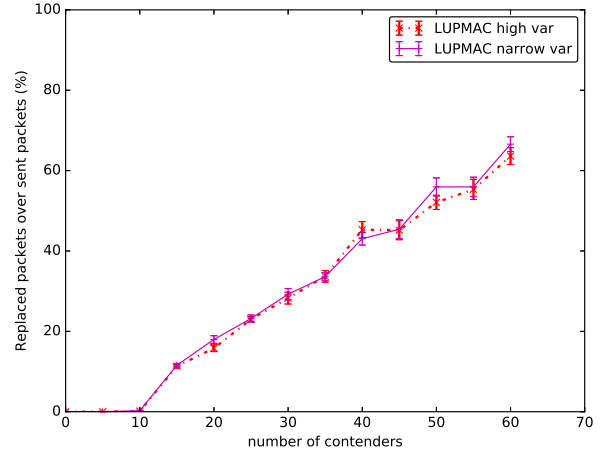

Fig. 4. Percentage of the replaced packets according to Algorithm 1 over the totality of packets sent by the application layer in the sensor.

the percentage of the replaced packets according to Algorithm 1 over the totality of packets sent by the application layer in the sensor is presented. As we can see, LUPMAC starts to replace packets in the MAC buffer as soon as we have a sufficiently high number of contenders in the WLAN (in this case ≥ 15), exactly when the average AoI starts to diverge from the one measured in the standard case (i.e. IEEE 802.11 FIFO).

If we allow for a faster update generation, the benefits are overwhelming. In Fig. 5, the source on the sensor node is allowed to generate up to 100 pk/s, i.e. one packet every hundreth of a second with 30 contenders and narrow variance on the one-way network delay. Notice that the y-axis is in log-scale.

When LUPMAC is enabled, the average AoI is improved by up to an order of magnitude compared with when the sensor is simply relying on an unmodified IEEE 802.11 MAC. In addition, when LUPMAC is used, the average AoI is fairly stable, and its variance limited.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we investigated the effects of contenders on the age of information of a sensor node immersed in a dense IEEE 802.11 WLAN via simulation. We then investigated the effects of variance of the transmisson delay on the age of information. We also introduced a new MAC technique called LUPMAC designed to improve the performance of the IEEE 802.11 MAC for sensor nodes in terms of the average AoI.

An extension to this work will be to dynamically assign different traffic priorities to different information sources (i.e. different ACs, as defined in the IEEE 802.11e EDCA) according to priority and traffic load, following the findings from [1]. Another approach would be to use the technique described in [21], in order to use a probabilistic technique on top of LUPMAC, thus approximating the throughput optimal CW in order ot minimize further the average AoI at the receiver
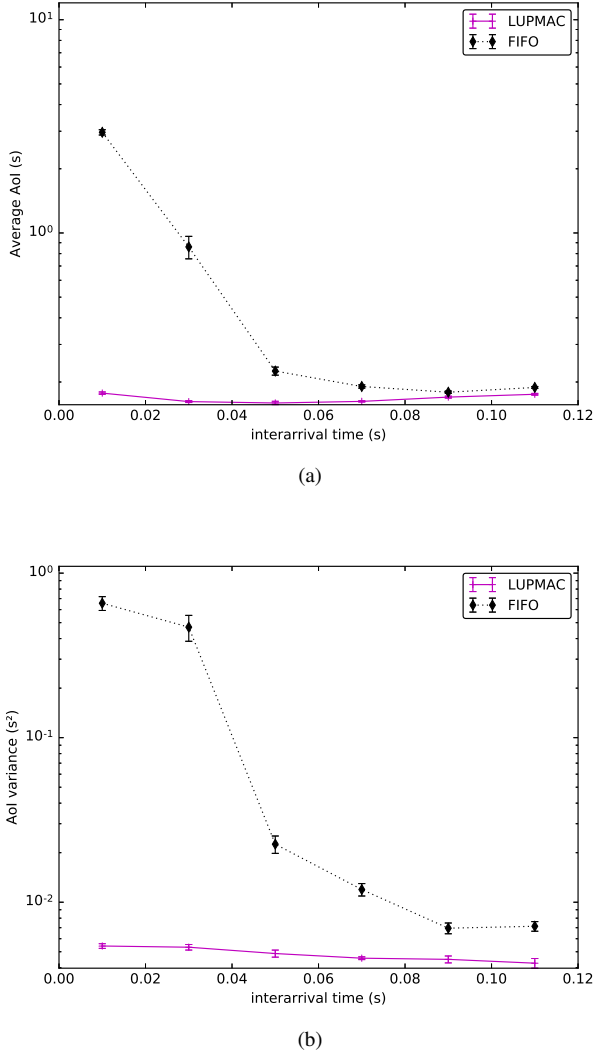
Fig. 5. Average Age of Information (a) and variance (b) measured at the destination both with and without LUPMAC with the sensor generating up to 100 pk/s, 30 contenders and narrow variance on the one-way network delay. Notice that the y-axis is in log-scale.

end. Since sensors are usually low power devices, we will investigate the effects of the contenders in terms of energy usage of the sensors, while trying to minimize the average AoI at the receiver end taking inspiration from the findings in [12] and extending them in a realistic environment. Further steps are also a real implementation of LUPMAC in a sensor node and a mathematical evaluation of its performances in terms of the average AoI at the receiver end.

LUPMAC can be integrated into the existing IEEE 802.11ah standard, and fits in the wider scope of the IoT and 5G. It shows substantial benefits in terms of both the average AoI and its variance compared to the normal, unmodified IEEE 802.11 when the WLAN becomes saturated with traffic. This technique is also resilient to changes in the variance on the experienced delay.

REFERENCES

[1] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*, June 2011, pp. 350–358.

[2] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 2731–2735.

[3] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali, "Smart cities of the future," *The European Physical Journal Special Topics*, vol. 214, no. 1, pp. 481–518, 2012. [Online]. Available: http://dx.doi.org/10.1140/epjst/e2012-01703-3

[4] D. J. Deng, K. C. Chen, and R. S. Cheng, "IEEE 802.11ax: Next generation wireless local area networks," in *Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), 2014 10th International Conference on*, Aug 2014, pp. 77–82.

[5] W. Sun, M. Choi, and S. Choi, "IEEE 802.11 ah: A long range 802.11 WLAN at sub 1 GHz," *Journal of ICT Standardization*, vol. 1, no. 1, pp. 83–108, 2013.

[6] L. Hu, L. L. Sanchez, M. Maternia, I. Z. Kovacs, B. Vejlgaard, P. Mogensen, and H. Taoka, "Modeling of Wi-Fi IEEE 802.11ac offloading performance for 1000x capacity expansion of LTE-advanced," in *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, Sept 2013, pp. 1–6.

[7] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "Understanding link-layer behavior in highly congested IEEE 802.11b wireless networks," in *Proceedings of the 2005 ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis*, ser. E-WIND '05. New York, NY, USA: ACM, 2005, pp. 11–16. [Online]. Available: http://doi.acm.org/10.1145/1080148.1080151

[8] E. Fitzgerald and B. Landfeldt, "The failure of CSMA in emerging wireless network scenarios," in *Wireless Days (WD), 2014 IFIP*, Nov 2014, pp. 1–4.

[9] N. Pappas, J. Gunnarsson, L. Kratz, M. Kountouris, and V. Angelakis, "Age of information of multiple sources with queue management," in *Communications (ICC), 2015 IEEE International Conference on*, June 2015, pp. 5935–5940.

[10] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[11] M. Costa, M. Codreanu, and A. Ephremides, "Age of information with packet management," in *Information Theory (ISIT), 2014 IEEE International Symposium on*, June 2014, pp. 1583–1587.

[12] B. T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," in *Information Theory and Applications Workshop (ITA), 2015*, Feb 2015, pp. 25–31.

[13] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," in *Information Theory (ISIT), 2015 IEEE International Symposium on*, June 2015, pp. 1681–1685.

[14] C. Kam, S. Kompella, and A. Ephremides, "Experimental evaluation of the age of information via emulation," in *Military Communications Conference, MILCOM 2015 - 2015 IEEE*, Oct 2015, pp. 1070–1075.

[15] D. H. D. West, "Updating mean and variance estimates: An improved method," *Commun. ACM*, vol. 22, no. 9, pp. 532–535, 1979. [Online]. Available: http://doi.acm.org/10.1145/359146.359153

[16] Lalitpatanpur, "Linksys rangebooster [SVG file]," Open Clipart, https://openclipart.org/detail/169452, April 2012.

[17] Argumento, "Laptop and tablet [SVG file]," Open Clipart, https://openclipart.org/detail/218353, May 2007.

[18] Ujmoser, "Application server [SVG file]," Open Clipart, https://openclipart.org/detail/182736, August 2013.

[19] Barnheartowl, "Grey cloud [SVG file]," Open Clipart, https://openclipart.org/detail/236010, January 2016.

[20] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools '08, 2008, pp. 60:1–60:10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1416222.1416290

[21] M. Karaca, Z. Zhang, and B. Landfeldt, "An opportunistic backoff scheme for dense IEEE 802.11 WLANs," in *2015 IEEE Globecom Workshops (GC Wkshps)*, Dec 2015, pp. 1–6.