Super Fast Link Set-Up in Wi-Fi HaLow Networks

Dmitry Bankov, *Member, IEEE*, Evgeny Khorov, *Senior Member, IEEE*, Katarzyna Kosek-Szott, and Marcin Trebunia

***Abstract*— In Wi-Fi HaLow networks, sensors can transmit data only after the link set-up procedure (LSP). After a power outage or when a swarm of sensors appears in an Internet of Things network, all of them contend for the channel to set up links with the access point. For thousands of sensors the LSP can last for hours. However, it can be shortened with advanced LSP coordination algorithms. We show that the best standard-compatible solution found in the literature leaves a gap between the achievable link set-up time and the estimated lower bound. In this paper, we design and evaluate an adaptive solution to control the LSP that fills in this room for improvement and provides significant gains against the existing algorithms.**

***Index Terms*— Fast association, IEEE 802.11ah, Internet of Things, sensor networks, centralized control.**

I. Introduction

THE 802.11ah amendment [1], [2] introduces new algorithms for supporting Internet of Things (IoT) applications. In particular, it allows an access point (AP) to effectively manage large groups of sensor stations (STAs). However, before the regular operation, the STAs need to set up links with the AP. The link set-up procedure (LSP) consists of authentication and association handshakes performed with random channel access. Therefore, in the worst-case scenario, when many STAs appear in an IoT network (e.g., after an AP reboot), they concurrently begin their LSPs, which leads to frequent collisions. As shown in [3], the LSP duration grows exponentially with the number of STAs. To speed up the LSP, 802.11ah defines two types of authentication control mechanisms: distributed and centralized. In this paper, we concentrate on the centralized authentication control (CAC) because it outperforms the distributed approach [4].

802.11ah provides a general framework for the CAC signaling but it does not define any algorithm on how to control authentication. The main challenge here is that before finishing LSP by all the STAs, the number of STAs is unknown to the AP. Thus, it needs to learn how many STAs are going to set-up links and then control their LSPs accordingly. If the AP fails to obtain an accurate estimation on the number of STAs, the time needed to complete the LSP grows. Specifically, because of such mistakes, to our best knowledge, the most efficient algorithm proposed in the literature so far [4] (which we refer to as *Old*) provides 30% to 40% slower LSP than the estimated lower bound (which we refer to as Oracle). In the paper, we have studied why and when the *Old* algorithm fails and designed an adaptive solution that modifies its behavior and makes it more resilient to changing network conditions and fluctuations in the number of sensor STAs authenticating in different beacon intervals (BIs). Our solution complies with the standard, requires only software modifications for the AP, and fills in the room for improvement left by its predecessors.

The rest of the paper is organized as follows. Section II describes the LSP in Wi-Fi Halow networks. Section III reviews the previous works. In Section IV, we introduce and explain in detail the designed solution. In Section V, we describe the scenarios and discuss the obtained numerical results. In Section VI, we summarize our contribution.

II. Link Set-Up Procedure

The LSP consists of two major handshakes: authentication and association [2]. The first one allows exchanging security keys or ensuring that a STA may access the network using Open System Authentication. The STA sends an *Authentication Request* (AuthReq) to the AP, which responds with an *Authentication Response* (AuthRep). In the second handshake, the STA sends an *Association Request* (AReq) to the AP, and awaits the *Association Response* (ARep), which provides the STA with network parameters.

Frames exchanged during the LSP are sent according to the enhanced distributed channel access (EDCA), the performance of which significantly depends on the number of STAs contending for the channel (N_{STA}). The more STAs contend, the higher is the collision probability. In case of a collision, the STAs repeat their transmission attempts, which increases delays and energy consumption.

To solve this problem, 802.11ah [1] defines CAC, which works as follows. The AP selects an authentication control threshold (ACT), $ACT \in \{0, 1, \ldots, 1023\}$, and broadcasts it in beacons or in probe response frames. When a STA is turned on, it generates a random value $p \in \{0, 1, \ldots, 1022\}$ and compares p to the received ACT. If p is smaller than the ACT, the STA generates an AuthReq and sends it according to the EDCA rules. Otherwise, the STA waits for the next frame containing an ACT. If the STA fails to receive an AuthRep within an authentication timeout, it waits for the next beacon and repeats the described procedure. Note that the STA generates p only once at the beginning of LSP, and may generate a new value only after a successful association.

Manuscript received March 16, 2020; revised May 1, 2020; accepted May 27, 2020. Date of publication June 9, 2020; date of current version October 9, 2020. The work of Dmitry Bankov and Evgeny Khorov was supported by the Russian Government (Contract No 14.W03.31.0019). The work of Katarzyna Kosek-Szott was carried out as part of a project financed by the Polish National Science Centre (DEC-2018/30/M/ST7/00351). This research was supported in part by PL-Grid Infrastructure. The associate editor coordinating the review of this letter and approving it for publication was T. Han. *(Corresponding author: Marcin Trebunia.)*

Dmitry Bankov and Evgeny Khorov are with the Wireless Networks Lab, Institute for Information Transmission Problems of the Russian Academy of Sciences, 127051 Moscow, Russia (e-mail: bankov@wireless.iitp.ru; khorov@wireless.iitp.ru).

Katarzyna Kosek-Szott and Marcin Trebunia are with the Department of Telecommunications, AGH University of Science and Technology, 30-059 Kraków, Poland (e-mail: kosek@kt.agh.edu.pl; marcintutka94@gmail.com).

Digital Object Identifier 10.1109/LCOMM.2020.3001179

In general, the ACT value determines the percentage of STAs that can start the LSP, and the AP can manage the contention level in the network by setting the ACT. In [5] it is shown that such a procedure may significantly hasten the LSP. However, the standard does not define any algorithm for ACT management, and much research is dedicated to developing the algorithms that shorten the LSP.

III. State of the Art

The first ACT management algorithm is proposed in [6]. It requires the AP to check the size of its management queue (which stores AuthReps and AReps) at the end of each BI. If the queue length (q) is greater than some threshold, the AP decreases the ACT by some fixed Δ (and consequently, the number of STAs allowed to transmit in a given BI). Otherwise, the AP increases the ACT by Δ. This algorithm allows quickening the LSP in comparison with default behavior [7].

In [8], it is shown that the performance of the ACT management algorithm depends on Δ and N_{STA}. Specifically, Δ should be high if N_{STA} is small and vice versa. However, the AP does not know N_{STA} in advance, and cannot set Δ correspondingly. To the best of our knowledge, the most efficient standard-compliant adaptive algorithm that changes Δ depending on N_{STA} has been designed in [4]. We use this algorithm as our reference algorithm.

CAC is also studied in [9]–[13], which present non-standard solutions to hasten the LSP, including the ACT control algorithms. They use the approach for contention-free channel access, initially proposed in [9]. This approach is based on virtual carrier sensing, which allows a STA/AP to specify the time for which the channel will be virtually busy. The AP, after receiving an AuthReq, can use this mechanism to free the channel and provide contention-free channel access for the transmission of AuthRep, AReq, and ARep frames. In ideal conditions such a solution might save channel resources by limiting contention for channel access only for the transmission of the AuthReqs.

In [10], [11] the authors consider that AuthReps/AReps are transmitted instantly after AuthReqs/AReqs. However, HaLow defines that these frames should be first acknowledged by the acknowledgment frames (Acks). Such a limitation is partially caused by the fact that the AP needs time to process the AuthReqs/AReqs.

IV. Proposed Solution

Similarly to the *Old* algorithm [4], the proposed algorithm operates in three modes: *waiting*, *learning*, and *working* (Fig. 1). In the *waiting* mode, the AP sets the ACT to its maximal value, allowing all STAs to start the authentication procedure. Every BI the AP checks its management queue, and if by the end of a BI the queue has at least one AuthRep or ARep it sets ACT to zero and Δ to one and switches to the $learning$ mode.

In the *learning* mode, the AP firstly waits for several BIs with the minimal ACT until there are no more AuthReps and AReps in its queue. Afterward, the AP starts the learning process, trying to find the optimal Δ value. For that, every BI the AP increases the ACT by Δ and then doubles the Δ.

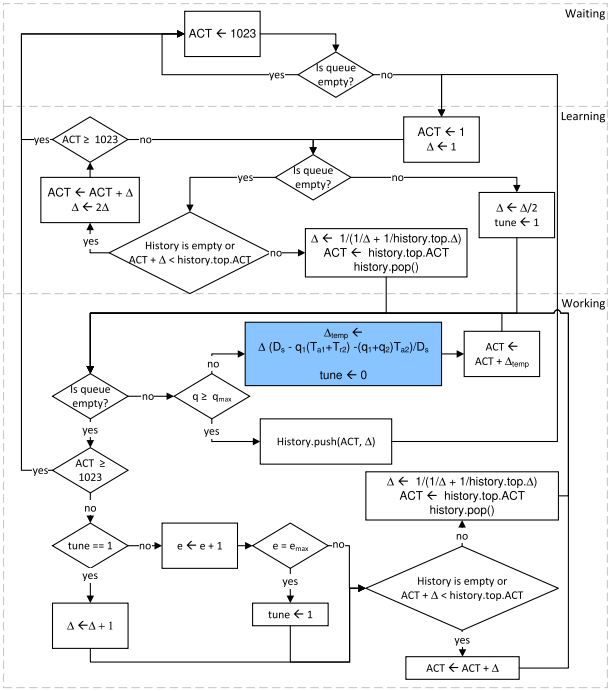


Fig. 1. *New* algorithm. Novel elements are marked in blue.

The AP runs this procedure until the AP's queue becomes non-empty by the end of some BI, which indicates that the current Δ is higher than the optimal value. Therefore the AP halves its Δ value and switches to the *working* mode.

In the *working* mode, the AP increases the ACT by Δ after each BI. At the same time, the AP tries to improve its estimate of the optimal Δ by increasing Δ by one if its management queue has been empty by the end of the last e_{max} BIs. When the ACT reaches its maximum value, the AP switches back to the *waiting* mode. This far, the algorithm operation repeats the *Old* one [4].

We observe that the *Old* algorithm does not reduce Δ in the *working* mode. However, Δ may be occasionally overestimated, or the network conditions may worsen during the LSP. In both situations, the estimated Δ is too large and cannot be easily reduced by the *Old* algorithm. It slows down the adaptation to changing network conditions.

As a workaround, the *Old* algorithm considers the number of incomplete LSP. If there are several STAs from the previous BI that have not finished their LSP, the AP forbids new STAs to start the LSP. As a result, the *Old* algorithm does not serve any new STAs in the next BI, but only the STAs for which the authentication process was not completed. This may lead to channel time underutilization.

Here we propose a modification of the *working* mode that eliminates the aforementioned drawbacks. The resulting algorithm is referred to as *New*. Assume that for given network conditions, there exists an optimal number of STAs k_o that can be served during one BI. Let the AP know the corresponding optimal Δ_o value such that when the the ACT is increased by Δ_o, the average number of STAs allowed to start the LSP equals k_o. After such an ACT increase, the number of STAs allowed to start the LCP can actually exceed k_o because the STAs draw their values p randomly and independently. In such

a situation, the AP cannot serve them all, and some AuthReps and AReps remain in its management queue. As a result, the *Old* algorithm stops the ACT increase, and in the next BI the AP serves only the remaining STAs, which leads to channel underutilization if the number of remaining STAs is less than k_o. This behavior can be optimized to let the AP serve the old STAs (for which the AuthReps and AReps are waiting in the queue) together with a portion of new ones. Therefore, in the *New* algorithm, we propose to increase the ACT by a temporarily decreased Δ according to the management queue size and the channel occupancy.

Let the AP use the parameter Δ in a BI that is finished with an empty queue. If the AP has q_1 AuthReps and q_2 AReps in its queue by the end of the next BI, the channel time occupied by successful transmissions in this BI equals

$$D_s = r_1 T_{r1} + a_1 T_{a1} + r_2 T_{r2} + a_2 T_{a2}, \quad (1)$$

where r_1, a_1, r_2 and a_2 are the numbers of successful AuthReqs, AuthReps, AReqs, AReps in this BI, and T_{r1}, T_{a1}, T_{r2}, T_{a2} are the average time intervals needed to transmit AuthReq, AuthRep, AReq, ARep, respectively, including the frame and Ack durations, and inter-frame spaces. During the next BI, the AP has to send the frames which are still in its management queue. Therefore the available channel time is decreased correspondingly. For this particular BI, the AP should use a temporary lower value Δ_{temp}:

$$\Delta_{temp} = \Delta \times \frac{D_s - q_1 (T_{a1} + T_{r2}) - (q_1 + q_2) T_{a2}}{D_s}. \quad (2)$$

While calculating Δ_{temp}, we take into account the fact that after the AP sends an AuthRep for a particular STA, the STA has to send an AReq and the AP has to send an ARep.

With the proposed modification, in the *working* mode, when the AP finds its queue non-empty at the end of a BI, it saves Δ (which provided an empty queue in the previous BI), and for the next BI uses the re-scaled Δ_{temp}, calculated according to Equation (2). If the queue is still non-empty at the end of the BI, the AP repeats the process of re-scaling Δ until the queue becomes empty again.

Additionally, similarly to the *Old* algorithm, we propose to re-start the learning process if the management queue length is greater than q_{max}.

V. Numerical Results

We have implemented the *New* algorithm in the ns-3 simulator and defined the following simulation scenarios.

- *Basic scenario:* A group of STAs appears at once and performs the LSP. It is the simplest scenario, studied, e.g., in [6].
- *Typical IoT network:* Several STAs perform the LSP. Immediately after finishing its LSP, each STA sends one packet. Such a behavior corresponds to a typical network of temperature or humidity sensors.
- *Highly loaded scenario:* Each of ten initially associated STAs generates a 100 B packet every 0.04 s. A new group of STAs appears and performs the LSP without sending any data. A similar scenario was studied in [4].

TABLE I
Simulation Parameters

ns-3 parameter	**Value**
`ErrorRateModel`	`YansErrorRate`
`EnergyDetectionThreshold` [dBm]	-105
`CcaMode1Threshold` [dBm]	-107
`RxNoiseFigure` [dB]	3
`TxPowerStart, TxPowerEnd` [dBm]	30 (AP), 16.02 (STA)
`TxGain, RxGain` [dB]	3 (AP), 1 (STA)
`DataMode`	`OfdmRate600Kbps`
`ChannelWidth` [MHz]	1
`BeaconInterval` [ms]	500, 1000
`AuthenticationTimeout` [ms]	512, 1024
`AIFSN`	1 (AP), 2 (STA)
`CWmin, CWmax`	15, 1023
***New* algorithm parameter**	**Value**
q_{max}	100
$T_{r1} = T_{r2}$ [μs]	1880
T_{a1}, T_{a2} [μs]	2680, 2320
e_{max}	5

- *Two groups scenario:* Initially, 2000 STAs start the LSP. After 20 s new STAs arrive, which force the AP to learn a new Δ because the initial STAs did not finish their LSP.

These scenarios represent different types of networks: lightly loaded (basic scenario, typical IoT network), highly loaded, and with strongly changing network conditions (two groups scenario). We consider two area sizes: *small* (20 m $\times$ 20 m) and *large* (400 m $\times$ 400 m). In the small area case, STAs sense each other's transmissions. In the large area case, distant STAs cannot sense each other's transmissions. LSP time is defined as the difference between the time when the group of associating STAs appears in the network and the time when the last associating STA sends an Ack to the AP confirming a successful reception of the ARep. Each simulation point is calculated based on 50 runs. Figures show the mean LSP time as well as the 10^{th} and 90^{th} percentiles. Table I lists the most important simulation parameters.

We compare *New*, *Old* [4], and *Oracle* algorithms. *Oracle* is an idealistic algorithm that *a priori* knows the number of associating stations and the corresponding optimal Δ value, and does not have to make any estimations. Thus its results can be considered as a lower bound for the LSP time.

Fig. 2 shows the LSP time for the basic scenario in a small area network as a function of N_{STA}. *New* outperforms *Old* by up to $\approx$30% and only insignificantly loses to *Oracle*. Because of a finer ACT tuning, during LSP, in the basic scenario *New* utilizes about 70% of the channel time comparing to $\approx 55\%$ used by *Old* (the figures are not shown because of space limitations). Fig. 2 compares the results for the BI duration of 0.5 s and 1 s. For 0.5 s, the LSP time is shorter by $\approx$20% because the AP has more control over the level of contention for channel access during the LSP. However, the gains of *New* over the other algorithms are the same for both BI lengths and, therefore, we further show only the results for the shorter BI.

The LSP time for the basic scenario in a large area network (see Fig. 3) is slightly lower than for the small area network because of the hidden STA effect: when a STA does not sense another STA's transmission, it does not freeze its backoff counter and, thus, has a lower delay for channel access.

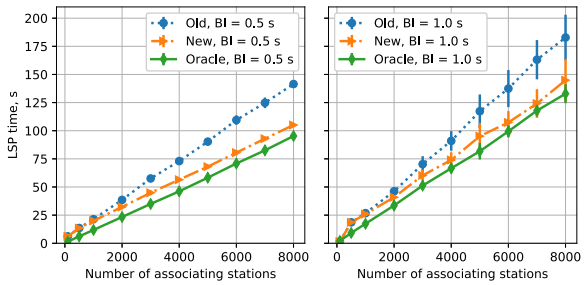


Fig. 2. Basic scenario, small area network.

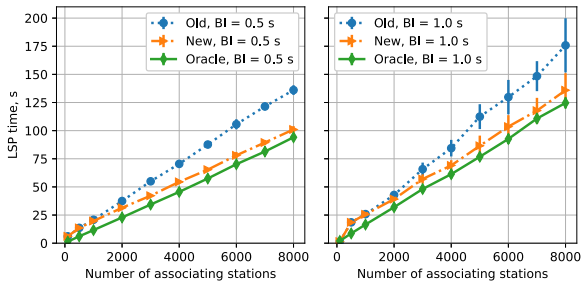


Fig. 3. Basic scenario, large area network.

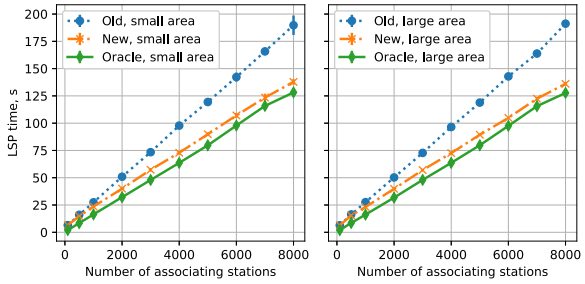


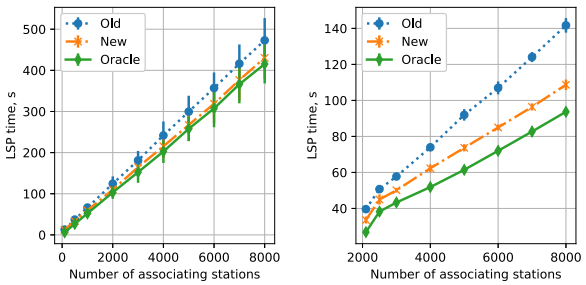
Fig. 4. Typical IoT network, BI duration set to 0.5 s.

The hidden STA effect also increases the collision rate, but the delay decrease has a higher impact on the LSP duration in this scenario.

For a typical IoT network, the LSP time is approximately 25% higher in comparison with the basic scenario (see Fig. 4). Such a difference is explained by the fact that the number of frames that should be exchanged for each STA increases from four to five. However, the performance of the algorithms relative to each other is the same as for the basic scenario: while *Old* shows $\approx 45\%$ higher LSP time than the Oracle, *New* fills in almost the whole room for improvement. Unlike for the basic scenario, in this scenario, the increased collision rate has a higher impact on the LSP time than the decreased channel access time, therefore the LSP time in a large area network is higher than for the small area network.

The gain becomes lower in the highly loaded scenario (Fig. 5a), because traffic generated by initially associated stations interferes with CAC operation. Notably, the LSP time significantly grows with respect to other scenarios and to speed up the LSP time, the AP would have to better manage transmissions of the associated STAs.

The *Old* algorithm is capable of detecting the appearance of a new group of STAs in the *working* mode [4]. The *New* algorithm inherits and improves this capability, as shown in the *Two groups* scenario. *New* outperforms *Old*



(a) Highly loaded scenario (b) Two groups scenario

Fig. 5. Large area network, BI duration is 0.5 s.

(up to ≈30% gain), thanks to better channel utilization. However, it cannot reduce the LSP time to the value of *Oracle* because *Oracle* has perfect knowledge of N_{STA} and instantly learns when the new group of STAs appears.

VI. Conclusion

The paper introduces and evaluates the *New* algorithm to manage the CAC parameters that can significantly reduce the LSP duration. While the existing standard-compatible solution provides 30–50% higher LSP time with respect to the *Oracle*, which can be considered as a lower bound, our algorithm provides results close to *Oracle* with the discrepancy of about 5% in the majority of the considered scenarios.

References

[1] *IEEE Standard, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation*, IEEE Std 802.11ah, May 2017, pp. 1–594.

[2] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, "A survey on IEEE 802.11ah: An enabling networking technology for smart cities," *Comput. Commun.*, vol. 58, pp. 53–69, Mar. 2015.

[3] D. Bankov, E. Khorov, and A. Lyakhov, "The study of the distributed control method to hasten link set-up in IEEE 802.11ah networks," in *Proc. 15th Int. Symp. Problems Redundancy Inf. Control Syst. (REDUNDANCY)*, Sep. 2016, pp. 13–17.

[4] D. Bankov, E. Khorov, A. Lyakhov, E. Stepanova, L. Tian, and J. Famaey, "What is the fastest way to connect stations to a Wi-Fi HaLow network?" *Sensors*, vol. 18, no. 9, p. 2744, Aug. 2018.

[5] P. Sthapit, S. Subedi, G.-R. Kwon, and J.-Y. Pyun, "Performance analysis of association procedure in IEEE 802.11ah," in *Proc. ICSNC*, 2015, pp. 1–4.

[6] H. Wang. (2012). *Supporting Authentication/Association for Large Number Stations*. [Online]. Available: http://mentor.ieee.org

[7] L. Tian, S. Deronne, S. Latré, and J. Famaey, "Implementation and validation of an IEEE 802.11ah module for ns-3," in *Proc. Workshop ns-3 (WNS3)*. New York, NY, USA: ACM, 2016, pp. 49–56.

[8] D. Bankov, E. Khorov, A. Lyakhov, and E. Stepanova, "Fast centralized authentication in Wi-Fi HaLow networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[9] D. Bankov, E. Khorov, and A. Lyakhov, "The study of the centralized control method to hasten link set-up in IEEE 802.11ah Networks," in *Proc. Eur. Wireless*. Frankfurt, Germany: VDE, 2015, pp. 1–6.

[10] N. Shahin, L. Tann, and Y.-T. Kim, "Enhanced registration procedure with NAV for mitigated contentions in M2M communications," in *Proc. 18th Asia–Pacific Netw. Operations Manage. Symp. (APNOMS)*, Oct. 2016, pp. 1–6.

[11] P. Sthapit and J.-Y. Pyun, "Station grouping strategy for minimizing association delay in IEEE 802.11 ah," *IEICE Trans. Commun.*, vol. E100.B, no. 8, pp. 1419–1427, Aug. 2017.

[12] N. Shahin, R. Ali, and Y.-T. Kim, "Hybrid slotted-CSMA/CA-TDMA for efficient massive registration of IoT devices," *IEEE Access*, vol. 6, pp. 18366–18382, 2018.

[13] N. Shahin, R. Ali, S. Y. Nam, and Y.-T. Kim, "Performance evaluation of centralized and distributed control methods for efficient registration of massive IoT devices," in *Proc. 10th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2018, pp. 314–319.