Age of Information for Remote Sensing with Uncoordinated Finite-Horizon Access

Pooja Hegde[a], Leonardo Badia[b,*], Andrea Munari[a]

[a] *Inst. of Communications and Navigation, German Aerospace Center (DLR), Oberpfaffenhofen, Wessling, 82234, Bayern, Germany*
[b] *Dept. of Information Engineering, University of Padova, via Gradenigo 6B, Padova, 35131, Padova, Italy*

Abstract

We analyze a remote sensing system in the Internet of things, where uncoordinated nodes send status updates to a common receiver to achieve information freshness, quantified through age of information. We consider a finite horizon scheduling over a random multiple access channel, where colliding messages are lost. We show that nodes must adopt a further randomization to deviate from identical schedules and escape collision deadlocks. Moreover, we discuss the impact of feedback availability if, due to, e.g., energy expenditure, it decreases the number of transmission opportunities.

Keywords: Age of Information, Data acquisition, Random access, Scheduling, Distributed systems, Feedback.

1. Introduction

Age of information (AoI) [1, 2] is a performance metric describing the freshness of data received from a remote source, often applied in the Internet of things (IoT) when real-time data enable remote control for transportation, industrial applications, eHealth, and more [3, 4].

Most AoI studies take an oversimplified view, characterizing the metric over an infinite time horizon [5, 6, 7]. We argue that real-time system control is more likely to take place over intervals of finite duration. For example, production tasks are usually limited in time [8], and tracking a moving node is confined to the transit inside the region of interest [9]. Also, the dynamic nature of links justifies AoI-oriented control over a finite horizon [10], which can have better performance in terms of AoI stability [11].

A finite horizon allows for a more expressive representation of the limitations in the frequency of updates. In practical settings, constraints relate to energy expenditure and overhead [12]. IoT systems typically leverage LoRa or semi-persistent scheduling [13, 14]. These impose restrictions to the activity of the individual nodes [15], while at the same time implying an uncoordinated random access that makes information exchanges prone to collisions [16, 17], which severely impacts information freshness.

Finally, many works evaluating AoI assume that a given transmission pattern can be adapted based on a feedback provided by the sink [5, 18]. Feedback acquisition entails the consumption of additional time, processing, and energy, which is inconvenient for the node's budget [19], and can lead to additional collisions if performed in-band.

Taking the lead from these remarks, we study how to schedule a limited number of status updates over a given time span so as to minimize the average AoI, within a scenario where nodes share a common channel relying on uncoordinated access akin to slotted ALOHA [17, 20].

Although AoI-efficient scheduling received a lot of attention in recent years, our work is the first to study both a finite horizon and a collision-prone channel. In [5] (and many of the references therein), the scheduling is centralized, i.e., at most one node is allowed to transmit at every epoch, and all data is available from the start. In turn, [15] proposes age-based policies for ALOHA, for an infinite horizon and leaning on feedback about the current AoI, whereas [6] gives a game-theoretic analysis for slotted ALOHA without scheduling over an infinite horizon.

Stability of multi-source AoI-driven systems is studied in [21, 22], for an infinite horizon over an erasure channel without collisions from medium access. A finite horizon is considered in [11] for a single node transmission with stateful scheduling. A single link with erasures is also considered in [10], whereas [23] focuses on multiple nodes but again with a stateful procedure, and the scheduling is centralized over a given collision domain. The proposal of [24] focuses on collision resolution, assuming nodes have information not only on the outcome of their specific packets, but also on the collision domain. Such a feedback would have a prohibitive cost for a simple medium access with strong real-time requirements, and nevertheless we will show how we obtain better results with our approach. In [18], the AoI objective is translated into staying below some node-specific thresholds, still with a centralized approach. Finally, [19] focuses on a finite horizon for point-to-point links but disregards channel contention.

Thus, our contribution stands out from the existing literature, as in a finite horizon scenario with collisions one ought to successfully exploit every transmission op-

*Corresponding author
Email addresses: pooja.hegde@dlr.de (Pooja Hegde), leonardo.badia@unipd.it (Leonardo Badia), andrea.munari@dlr.de (Andrea Munari)

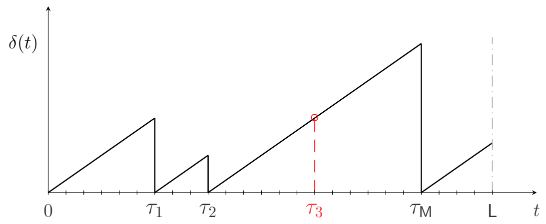

Figure 1: Sample AoI evolution over an epoch for a node performing $\mathsf{M} = 4$ transmissions, of which the third one fails due to a collision.

portunity (not just compensating for losses in the long-run). This leads to the following contributions. First, we show how an uncoordinated medium sharing policy calls for a randomization of the transmission pattern, which may trump over precisely allocating the best scheduling instant. In addition, we argue about the usefulness of a resource-consuming feedback, reducing the number of updates that can be sent, and characterize the number of nodes and achievable AoI, providing insights for system design, especially in settings where devices have limited capabilities in terms of energy or complexity.

2. System Model

Consider a remote sensing system, where U nodes share a common channel towards the data sink. Time is divided in slots of equal duration. Nodes are slot synchronized, each monitoring a process of interest, and, when accessing the medium, generating a fresh reading, sent as a time stamped status update to the sink. The reporting task spans a finite horizon (or epoch), of duration L slots, over which a node can perform at most M transmissions.

In practical IoT networks, the number of updates is restricted by battery or data traffic limitations [12], thus $\mathsf{M}/\mathsf{L} \ll 1$. The reporting intervals of distinct nodes, which may perform uncorrelated and non-synchronized monitoring, are in general not aligned. The channel is shared with slotted ALOHA uncoordinated access [6, 24]. Each node independently decides whether to access the channel to send an update. Following the collision channel model [17], a packet being the only one received in a slot is correctly received, but the presence of multiple transmissions in the same slot prevents all of them from being decoded.

Under this setting, we seek the instants $\tau_1, \dots, \tau_{\mathsf{M}}$ in which a node shall perform its M transmissions, to obtain up-to-date monitoring. To capture this, we track the instantaneous AoI for a reference terminal, defined as $\delta(t) := t - \sigma(t)$, where $\sigma(t)$ is the time of the last update received as of time t. An example is shown in Figure 1, epitomizing a saw-tooth profile that grows linearly over time, until being reset upon successful updates. We gauge the average AoI at the sink over the time span L as

$$\Delta := \frac{1}{\mathsf{L}} \int_0^{\mathsf{L}} \delta(t)\, dt. \tag{1}$$

The definition highlights the role of the finite horizon. The transmission constraints set a *precise limit* on the number of transmissions M within an epoch, as commonly encountered in practical systems due to normative regulations. If L goes to infinity, and similarly M is scaled up, such a precise limit no longer holds, and the constraint is relaxed to a long-term average transmission activity, rendering the problem less realistic [12, 19].

In this setting, we study three distinct access strategies:

1) Random transmission policy: in the simplest approach, nodes select M transmission slots uniformly at random in $\{1, \dots, \mathsf{L}\}$. This policy is relevant as an archetype of slotted ALOHA not coping specifically with AoI.

2) Fixed-allocation (no-feedback) policy: each terminal determines in advance the M slot indexes at which to perform transmissions [25]. This algorithm, described in Sec. 3, only requires knowledge of L, M, and an estimate of the collision probability p_c. The strategy is appealing for practical systems in view of its simplicity, allowing to pre-load a schedule without feedback from the sink.

3) Dynamic-allocation policy: this requires the sink to provide feedback at the end of each transmission, on whether the update was received or not (collision) [15]. As per [26], it is not restrictive to assume this feedback to be error-free. Thus, each node is aware of the current AoI $\delta(t)$, and can dynamically adapt its schedule, e.g., postponing the next update in case of success, or preempting it in case of failure. This potential advantage is counterbalanced by a cost to implement the feedback. The reception of acknowledgements requires indeed a node to remain active (e.g., postponing sleep mode) and attempt decoding of the packet. To account for this, we introduce a cost coefficient $\gamma \geq 0$, so that implementing feedback reduces the number of available transmissions to $\mathsf{M}_{\mathrm{f}} = \mathsf{M}/(1+\gamma)$.

3. Fixed and Dynamic Transmission Policies

The problem at hand resembles the scheduling of status updates over an erasure channel [21], where the medium suffers from packet losses with probability p_c. However, losses are due to collisions, hence $p_c = \mathbb{P}\{\#\text{ transmitting nodes} \geq 2)$, with non-trivial interdependency across nodes.

Each individual node faces the problem of how to schedule its updates, which are finite in number and may be lost due to collisions, so as to minimize AoI [10, 18]. Under this lens, it is immediate to represent the problem as a *stateless* optimization, i.e., an offline choice of the best transmission instants, taking expectations on the channel performance, thus resulting in the following problem:

$$\begin{aligned} \min_{\mathbf{y}} \quad & \Delta(\mathbf{y}) \\ \text{subject to} \quad & y_i \geq 0, \quad i = 0, \dots, \mathsf{M}, \\ & \sum_{i=0}^{\mathsf{M}} y_i \leq \mathsf{L}. \end{aligned} \tag{2}$$

where we set $\mathsf{M}+1$ integer variables $y_0, y_1, \dots, y_{\mathsf{M}}$. Each $y_i = \tau_{i+1} - \tau_i$ is the duration of the i-th inter-update interval, with $\tau_0 = 0$ and $\tau_{\mathsf{M}+1} = \mathsf{L}$. The expression of the average AoI $\Delta(\mathbf{y})$ as a function of the y_is can be derived from the following result.

Theorem 1. *The average AoI over a finite horizon* L *for an update pattern at instants* τ_i*,* $i = 1, \dots, \mathsf{M}$*, resulting in* $\mathsf{M}+1$ *inter-update intervals* y_i *is computed as*

$$\Delta(\mathbf{y}) = \frac{1}{\mathsf{L}} \sum_{i=0}^{\mathsf{M}} \left[\frac{{y_i}^2 - y_i}{2} + \sum_{j=i+1}^{\mathsf{M}} y_i y_j (p_c)^{j-i} \right]. \quad (3)$$

This follows from geometric arguments (see Figure 1), along the lines of [2, 25]. However, important modifications are required, i.e., adjusting for a finite horizon and a discrete time. Moreover, the loss probability has a different meaning, which requires some adjustments discussed further. Specifically, the first term inside the summation is the sum of integers from 0 to y_i-1, corresponding to the area of the right triangles and always present even in the absence of losses, whereas the second term describes the increase in AoI due to losses.

The *fixed allocation strategy* is found as the solution to (2) and computes an AoI-driven schedule a priori.

Remark 1. *In the absence of losses, the fixed allocation policy obtained from* (2) *implies to equally space transmissions over the epoch, whereas the updates get closer to one another as the error probability increases [19].*

Remark 2. *The same approach can be used to derive a* random transmission policy*, skipping the minimization of the average AoI* $\Delta(\mathbf{y})$ *and setting any choice of the* y_i*s to be valid as long as it meets the constraints in (2).*

Alternative to setting the transmission instants in advance, we can consider a *stateful* optimization, obtained by dynamic programming [22], tracking feedback obtained through acknowledgments of transmitted packets [26], which allows an individual node to know its current AoI value $\delta(t)$. The transmission instants over discrete time slots result from a dynamic program, as follows.

For the synchronized slotted ALOHA system under study with U users over a finite horizon of size L, the action space comprises L slots, where the ith of them is the time interval $[i-1, i]$, $i = 1, \dots, \mathsf{L}$, and thus there are $\mathsf{L}+1$ meaningful input values for the time (from 0 to L). Accordingly, we define the system state over integer values as $x[n] = \big(\delta[n], m[n]\big)$, where $n \in \mathbb{Z}$, $0 \le n \le \mathsf{L}$ is a discrete time instant; $\delta[n] \in \mathbb{Z}$, $\delta[n] \ge 0$ is the current AoI; $m[n] \in \mathbb{Z}$, $0 \le m[n] \le \mathsf{M}$ is the number of the remaining transmissions. We also set a control action $u(x[n]) \in \{0, 1\}$ that corresponds to either transmit an update or not, as $u = 1$ and $u = 0$, respectively. Finally, a noise process describes lost updates. For the time being (this will be relaxed later) assume that each update may be lost with independent identically distributed (i.i.d) probability p_c.

Theorem 2. *Finite-horizon problem* $(x[n], u(x[n]), p_c)$ *admits optimal control.*

The theorem follows from backward induction arguments [18]. Control $u(x[n])$ is initialized as 0 for states where $m[n] = 0$, and 1 when $n = \mathsf{L}$ and $m[n] > 0$, i.e., there are transmission opportunities left. The system transitions to state $x[n+1]$ based on $x[n]$, $u(x[n])$, and noise; $m[n]$ either decreases by 1 or stays the same, depending on the control action, and noise determines whether, in the presence of an update, $\delta[n]$ is reset to 0, or increases by 1 if the update fails (as would happen if no update is scheduled).

Remark 3. *The optimal control action derived from Theorem 2 is* non stationary*, as it changes with* n *[7], yet the solution method works because of the finite horizon [10].*

This approach is *not* exact due to the nature of the multiple access. AoI minimization ought to be cast as a multi-agent problem because the collision probability p_c is not an external noise but rather depends on the actions of all nodes. An exact globally optimal schedule is in principle possible, albeit impractical to be computed locally due to its formidable complexity, and it cannot be disseminated throughout the network without a heavy centralization.

Thus, we estimate the collision probability p_c based on the average transmission rates. This allows to derive a preliminary schedule, which, as argued in the next section, can be refined by adding randomization, to improve collision avoidance. To estimate p_c, note that, if each node allocates M transmissions over L slots, it will globally transmit on average with probability M/L (this is still exact). If all transmissions of different nodes are independent and identically distributed (i.i.d.) over multiple slots, p_c is the probability of at least another terminal transmitting in the same slot, thus

$$p_c \simeq 1 - (1 - \mathsf{M}/\mathsf{L})^{\mathsf{U}-1}. \quad (4)$$

For both static and dynamic policies, we encounter collisions since nodes decide the transmission instants based on a local (myopic) optimization.[1] Hence, a *deadlock* may arise, i.e., nodes that collided once will keep doing so as they follow the same pattern in subsequent slots and one ought to randomize the transmission pattern, as it is better to have successful status updates than to schedule them in the precise best moment [24]. In the next section, we show the extent of this issue, and we argue for the benefit of a randomization in the scheduling.

4. Results and Discussion

We investigate the performance of the introduced AoI-driven policies. Unless otherwise specified, we consider $\mathsf{U} = 50$ nodes and epoch duration $\mathsf{L} = 500$ slots. To avoid border effects,a random offset uniformly distributed in $\{0, \dots, \mathsf{L}\}$ slots is applied to the start of the epoch for each node, so that different nodes are not aligned.

[1]For our model, retransmitting lost packet is of no use towards AoI minimization [25]. Thus, it is prioritary to avoid collisions. As a byproduct, the ALOHA system studied is inherently stable.

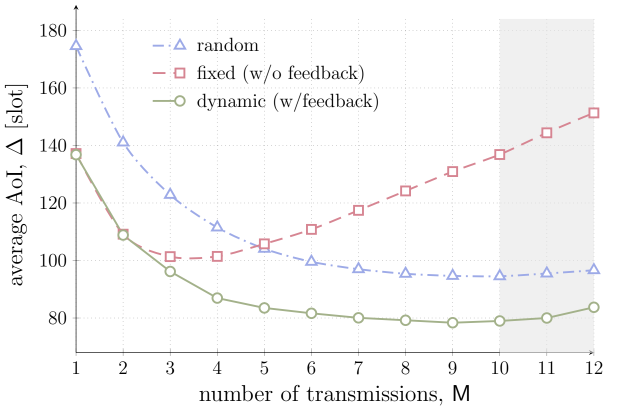

Figure 2: Average AoI vs number of performed transmission per node over a task duration. $\mathsf{U} = 50$, $\mathsf{L} = 500$ slots, $\gamma = 0$.

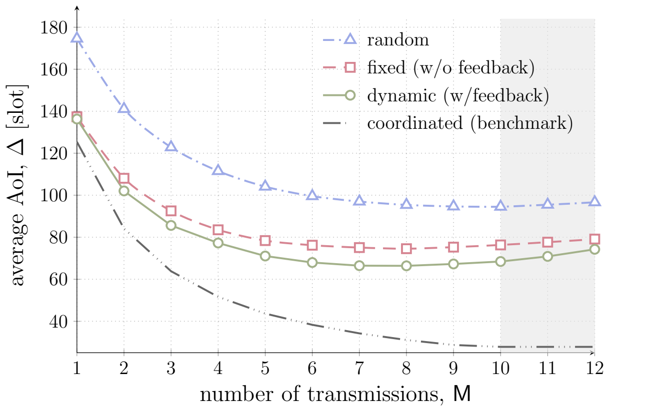

Figure 3: Average AoI vs number of performed transmissions per node over a task duration. A randomization of the transmission instant is considered. $\mathsf{U} = 50$, $\mathsf{L} = 500$ slots, $\gamma = 0$.

Figure 2 reports the average AoI vs. the number of transmissions M. For the dynamic approach (solid line), ideal cost-less feedback is assumed ($\gamma = 0$). For all policies, AoI is high for low values of M due to very sporadic transmissions, as well as for too large ones, due to collisions. When few transmissions are performed (leftmost part of the plot), the feedback does not play a significant role. Indeed, if the channel is lightly loaded, and collisions are rare, the knowledge of transmission outcomes has little use. Yet, a careful scheduling of the updates, even without feedback, decreases AoI over the random policy.

As M increases, a sharp performance degradation is undergone by the static approach, driven by the increased collision rate experienced when all nodes employ the same pre-computed schedule. We stress that neither approach performs collision avoidance, yet the availability of feedback plays a key role, allowing nodes to dynamically react to failed transmissions by rescheduling their updates.

All policies ultimately obtain an exploding Δ when M is high, as even the availability of feedback can do little when the schedule is so clogged that collisions are unavoidable. This is certain to happen when $\mathsf{MU}/\mathsf{L} \geq 1$, and is highlighted in the plot by the gray-shaded on the rightmost part of the figure ($M \geq 10$). For medium loads (i.e., M=5 instead of 10), the static solution performs closely to the random one, whereas an AoI reduction of $\sim$20% is achieved with the dynamic scheme.

These results highlight how the performance of the considered scheduling policies is severely beset by recurrent collisions among nodes. This also impacts, although to a lesser extent, the dynamic scheme. Terminals involved in a collision will reschedule their next transmissions sooner, aiming to avoid an excessive growth of their AoI.

This reactive behavior leads to harsher contention in the upcoming slots, increasing the collision probability. Thus, we propose a heuristic approach to mitigate the issue, where nodes send their updates as randomly chosen within an interval of k slots around the deterministic value indicated by the scheduled instant. This means that, whenever the static or dynamic policies dictate a transmission at time τ, a random number between 1 and k is chosen with probability $1/k$ and the instant in $\{\tau - \lfloor k/2 \rfloor, \dots, \tau + \lceil k/2 \rceil\}$ in the corresponding position is chosen for the transmission. This avoids that users colliding once adopt the same deterministic scheduling in the following transmissions. Other proposals [24] adopt a similar mechanism as *collision resolution* assuming full information about how many nodes underwent collision. Our mechanism is instead preemptive and totally decentralized, i.e., agnostic to the collision domain. Still, despite having less information, we will show that it obtains better performance, thanks to the optimization of the scheduling.

Such a strategy trades off the benefits of an AoI-driven schedule for a higher success probability. From this standpoint, the width of the randomization interval is paramount, e.g., for large intervals the behavior converges to a fully random transmission policy.

The performance attained with this modification, applied to both static and dynamic strategies, is reported in Figure 3, where the parameter k has been optimized by means of an exhaustive search for each value of M. A significant reduction of the minimum achievable AoI is obtained. Compared to the basic slotted ALOHA of a random scheduling, our fixed and dynamic strategies improve by about 25% and 33%, respectively, for the best case of a transmission load of about 0.5 (i.e., $\mathsf{M} = 5$). Even for a fully loaded channel with $\mathsf{M} = 10$, these improvements still persist, being equal to 18% and 27%, respectively.

To provide further insights, Figure 3 also shows a fully coordinated scheduling (black dash-dotted line). Specifically, this benchmark policy fairly assigns transmission slots to each node, avoiding collisions and minimizing the target metric Δ.[2] The gap between a centralized approach and the fully uncoordinated dynamic solution is not large, and can be possibly reduced with more complex policies.

[2]The best performance of the scheme is achieved for a fully utilized channel, i.e. $\mathsf{ML} = 1$. After this point, the policy does not allocate more transmissions, as they would results in collisions.

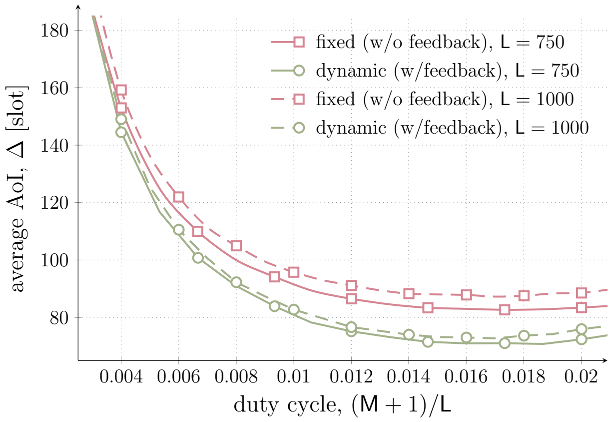

Figure 4: Average AoI vs. duty cycle $(\mathsf{M}+1)/\mathsf{L}$, considering $\mathsf{L} = 750$ and $\mathsf{L} = 1000$ slots. $\mathsf{U} = 50$, $\gamma = 0$.

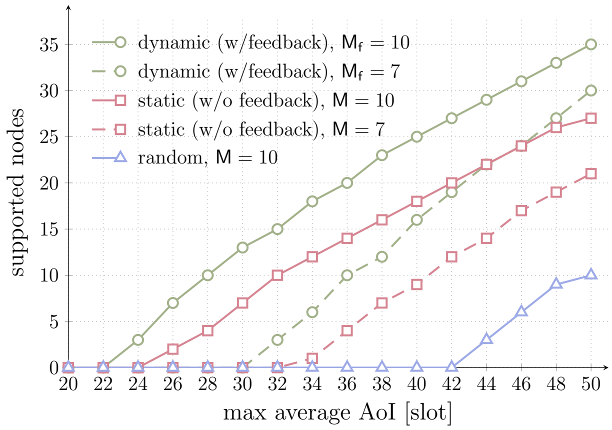

Figure 5: Number of nodes supported in the network without violating a target average AoI. A randomization of the transmission time is considered. In all cases, $\mathsf{L} = 500$.

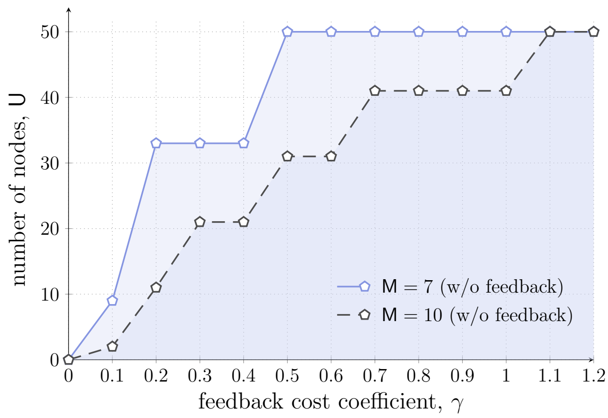

Figure 6: (γ, U) pairs for which the use of feedback provides lower (white region) or higher (shaded regions) average AoI compared to the static approach. In all cases, $\mathsf{L} = 500$ slots.

Our result can be compared (with proper adaptations) with [24]. The latter cannot be directly juxtaposed, as it considers infinite horizon scheduling, without constraints in the number of transmissions, and optimizes only collision resolution, performed under the assumption of knowing the number of colliding nodes; yet, it claims an AoI gain of 18% over random transmissions. We get a larger improvement (up to 33%), in spite of more challenging conditions, i.e., limited transmissions and no information about the collision domain size.

Figure 4 explores the impact of the horizon length L. For a homogeneous comparison of scheduling with different horizons, but analogous frequency of transmission, we define the *duty cycle* as $(\mathsf{M}+1)/\mathsf{L}$, i.e., the reciprocal of the inter-transmission time. In all cases, the benefit of an online approach is confirmed. Yet, operation over longer time spans (larger L, dashed lines) incurs higher AoI. For low duty-cycles (which is of practical interest), the discrepancy is limited, and stems from the bias of an initial AoI equal to 0 experienced for smaller L. When the transmission frequency increases, the higher collision rate propagates over a longer time horizon. Thus, the increase in average AoI induced by packet losses is more acute for larger L.

All results reported so far were obtained assuming the same number of updates per epoch available to both static and dynamic approaches. We now consider the case in which the implementation of feedback entails a reception energy cost, as per the model of Sec. 2. Figure 5 plots the maximum system size, i.e., the number of nodes that can be admitted in the system without their average AoI exceeding a target value (e.g., dictated by the application of interest). The looser the AoI requirements, the larger the device population that can be served. Solid lines – denoting the schemes with $\mathsf{M} = 10$ – confirm the trends, pinpointing the improvements due to a cost-free feedback. When fewer updates per epoch are available ($\mathsf{M} = 7$, dashed lines), transmissions become even more valuable, and a target AoI can only be attained by limiting the number of devices and thus the collision probability, regardless of feedback.

Specific insights can be gained by comparing the solid, square-marked (static, $\mathsf{M} = 10$) and the dashed, circle-marked curves (dynamic, $\mathsf{M}_f = 7$). The latter corresponds to feedback implying a cost factor $\gamma = 0.4$, which reduces the number of allowed updates. For low target AoI values – and/or small numbers of devices – the static strategy offers better performance. In such conditions, a low channel load is experienced, and more transmissions lead to more frequent status updates. When higher AoI can be tolerated, the dynamic approach becomes convenient, as larger populations can be supported. Notably, under heavier channel contention, the option of a faster recover from a collision avoiding long refresh droughts outweighs the drawback of fewer transmissions being available.

Further light on the non-trivial role played by a costly feedback is shed by Figure 6. The plot considers M=7 and M=10, and reports for any pair (γ, U) whether a lower value of average AoI is achieved with or without feedback. White regions identify configurations of (γ, U) under which the dynamic approach is beneficial for the AoI despite the fewer available transmissions, whereas shaded regions denote the opposite. The step-wise behavior of the separation between regions is due to the granularity in the number of transmissions implied by γ. As shown by the results, the dynamic strategy offers better performance for low values of γ. The static approach is advantageous for progressively larger populations as the cost coefficient grows. The trend is more pronounced when fewer transmission opportunities, e.g., due to reduced battery capacity, are available, as the lower level of channel contention shrinks the region in which feedback shall be employed. Thus, the diagram offers a tool for system design, pinpointing the effect of a costly feedback on AoI.

5. Concluding Remarks

We studied the schedule of status updates in an IoT sensing system over a finite time window and for collision-based multiple access. We gave two take-away messages. First, in contention-based multiple access, individually optimal policies for update scheduling can lead to collision deadlock; the nodes ought to further randomize access, as avoiding collision is more important than precisely choosing the best scheduling instant. Exploiting feedback for a dynamic schedule is not always beneficial if it consumes additional resources. A static schedule with more transmissions is better for light loads; a dynamic approach with fewer transmissions becomes useful if congestion arises.

CRediT authorship contribution statement

Pooja Hegde: Software, Validation, Investigation, Writing – original draft. **Leonardo Badia:** Writing – review & editing, Supervision, Project administration. **Andrea Munari:** Conceptualization, Methodology, Formal analysis, Visualization, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

A. Munari acknowledges the support provided by the Federal Ministry of Education and Research of Germany in the programme "Souverän. Digital. Vernetzt," Joint project 6G-RIC, project id number: 16KISK022.

References

[1] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1183–1210, May 2021.

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

[3] Z. Qadir, K. N. Le, N. Saeed, and H. S. Munawar, "Towards 6G Internet of things: Recent advances, use cases, and open challenges," *ICT Express*, vol. 9, no. 3, pp. 296–312, 2023.

[4] X. Zheng, S. Zhou, and Z. Niu, "Urgency of information for context-aware timely status updates in remote control systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7237–7250, 2020.

[5] I. Kadota, A. Sinha, and E. Modiano, "Optimizing age of information in wireless networks with throughput constraints," in *Proc. IEEE Infocom*, 2018, pp. 1844–1852.

[6] L. Badia, "Impact of transmission cost on age of information at Nash equilibrium in slotted ALOHA," *IEEE Netw. Lett.*, vol. 4, no. 1, pp. 30–33, 2021.

[7] T. Soleymani, J. S. Baras, and K. H. Johansson, "Stochastic control with stale information-part I: Fully observable systems," in *Proc. IEEE CDC*, 2019, pp. 4178–4182.

[8] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial Internet of things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4925–4934, 2020.

[9] A. Baiocchi and I. Turcanu, "A model for the optimization of beacon message age-of-information in a VANET," in *Proc. ITC*, vol. 1, 2017, pp. 108–116.

[10] O. Ayan, S. Hirche, A. Ephremides, and W. Kellerer, "Optimal finite horizon scheduling of wireless networked control systems," *IEEE/ACM Trans. Netw.*, 2023, to be published.

[11] D. Ramirez, E. Erkip, and H. V. Poor, "Age of information with finite horizon and partial updates," in *Proc. IEEE ICASSP*, 2020, pp. 4965–4969.

[12] J. Yun, C. Joo, and A. Eryilmaz, "Optimal real-time monitoring of an information source under communication costs," in *Proc. IEEE CDC*, 2018, pp. 4767–4772.

[13] LoRa Alliance, "The LoRa Alliance Wide Area Networks for Internet of Things," www.lora-alliance.org.

[14] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.

[15] X. Chen, K. Gatsis, H. Hassani, and S. S. Bidokhti, "Age of information in random access channels," *IEEE Trans. Inf. Theory*, vol. 68, no. 10, pp. 6548–6568, Oct. 2022.

[16] D. Kanellopoulos, "Congestion control for MANETs: An overview," *ICT Express*, vol. 5, no. 2, pp. 77–83, 2019.

[17] R. Yates and S. K. Kaul, "Status updates over unreliable multiaccess channels," in *Proc. IEEE ISIT*, 2017.

[18] C. Li, S. Li, Y. Chen, Y. T. Hou, and W. Lou, "AoI scheduling with maximum thresholds," in *Proc. IEEE Infocom*, 2020, pp. 436–445.

[19] A. Munari and L. Badia, "The role of feedback in AoI optimization under limited transmission opportunities," in *Proc. IEEE Globecom*, 2022.

[20] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *ICT Express*, vol. 3, no. 1, pp. 14–21, 2017.

[21] W. Liu, D. E. Quevedo, B. Vucetic, and Y. Li, "Stability conditions for remote state estimation of multiple systems over semi-Markov fading channels," *IEEE Control Syst. Letters*, vol. 6, pp. 2954–2959, 2022.

[22] X. Xie, H. Wang, and X. Liu, "Scheduling for minimizing the age of information in multi-sensor multi-server industrial IoT systems," *IEEE Trans. Ind. Informat.*, 2023, to appear.

[23] N. Lu, B. Ji, and B. Li, "Age-based scheduling: Improving data freshness for wireless real-time traffic," in *Proc. ACM MobiHoc*, 2018, pp. 191–200.

[24] H. Pan, T.-T. Chan, J. Li, and V. C. Leung, "Age of information with collision-resolution random access," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 11 295–11 300, 2022.

[25] R. Talak, S. Karaman, and E. Modiano, "Distributed scheduling algorithms for optimizing information freshness in wireless networks," in *Proc. IEEE SPAWC*, June 2018.

[26] L. Badia, "On the effect of feedback errors in Markov models for SR ARQ packet delays," in *Proc. IEEE Globecom*, 2009.