Optimizing Freshness in IoT Scans

Johannes K Becker
Electrical & Computer Engineering
Boston University
Boston, MA, USA
jkbecker@bu.edu

David Starobinski
Electrical & Computer Engineering
Boston University
Boston, MA, USA
staro@bu.edu

***Abstract*—Motivated by IoT security monitoring applications, we consider the problem of a wireless monitor that must implement a multi-channel scanning policy to minimize the Age of Information (AoI) of received information. We model this problem as a Markov Decision Process (MDP). To address the curse of dimensionality, we propose various scanning policies of low computational complexity. We compare the performance of these policies against the optimal one in small instances, and further simulate them using time-series data obtained from real IoT device communication traces. We show that a policy, coined Greedy Expected Area (GEA), performs well in many scenarios.**

***Index Terms*—Wireless monitoring, Internet of Things, Modeling, Simulation.**

I. Introduction

The number of consumer and industrial Internet of Things (IoT) applications have exploded in recent years. The diversity of devices communicating in overlapping channel configurations (especially in the busy unlicensed spectrum bands) brings an interesting challenge of visibility, i.e., being able to monitor as much activity as possible simultaneously. One key motivation is security monitoring. The goal is to get a picture as complete and current as possible of local wireless devices to prevent wireless data exfiltration or any other activity of unauthorized devices.

As a general-purpose technology for addressing any number of wireless protocols, Software-Defined Radio (SDR) technology has grown into a multibillion-dollar market [1], [2]. One compelling characteristic of wireless transceivers implemented in software is the support for multiple protocols at the same time, and the ability to – within the limits of device capabilities – receive multiple channels at once. Despite these advantages, typical SDR hardware is currently not yet capable of capturing the entire ISM band concurrently, let alone monitor multiple bands of relevant radio traffic across the spectrum simultaneously.

We study the case of a passive monitor, which does not control the transmission behavior of its data sources, and aims to optimize the freshness of information across multiple channels. The monitor can cover one or more (but not all) channels at a time, and needs to decide which channel(s) to tune into. This problem is motivated by SDR-based multi-channel device enumeration across multiple IoT protocols [3].

This work contains the following key contributions:

- We introduce the problem of AoI minimization in multi-channel monitoring of wireless IoT devices and show that this problem can be formulated as an MDP optimization.
- We propose several monitoring policies of low complexity and characterize their performance, for the general case of a monitoring system with one or more devices per channel and/or a single or multi-channel monitor.
- We prove the optimality of several of the proposed scanning policies in the special case of a symmetric network with one device per channel and a monitor capable of scanning one channel at a time,
- We numerically compare the performance of the proposed policies with the MDP solution. We show that one of these policies, called Greedy Expected Area (GEA), performs close to optimal even under adverse conditions.
- We further compare the performance of the various policies using time-series of traffic collected from real-world IoT device communication [3], [4].

The remaining sections of this paper are organized as follows. Section II gives an overview of related work on multi-channel monitoring and AoI optimization. Section III defines key terms, proposes a general formulation for multi-channel monitoring, and shows that it can be cast as an MDP optimization problem. Section IV introduces various monitoring policies of low complexity and analyzes their performance. Section V evaluates the performance of the policies against the optimal one, and with actual traces of IoT traffic. Section VI concludes the paper.

II. Related Works

The *Age of Information (AoI)* metric is frequently used to characterize freshness of information in networked systems [5]. Optimizing for AoI in a communication system is not equivalent to simply maximizing throughput, as shown by Kaul et al. [6] and further explored in a number of contexts over the last decade, often in the context of wireless sensor networks. Most research on AoI in wireless sensor networks has focused on controlling transmission under certain constraints such as energy conservation [7]–[11], limited capacity for concurrent communication [12]–[16], or lossy transmission media [12], [13].

Alsliety and Aloi [17] discuss the challenges of using software-defined radio (SDR) in telematics across multiple protocols, and Choong [18] demonstrates a multi-channel Zigbee receiver implementation. These practical works motivate the theoretical multi-channel monitoring problem studied in our paper.

Tripathi and Moharir [7] and Sombabu and Moharir [8] discuss a system in which multiple sensors communicate to a monitor via a number of communication channels. They propose scheduling policies that optimize AoI at the monitor and energy consumption of sensors. Zhou and Saad [9], [10] introduce system comprised of distributed, energy-constrained sensors, and propose a semi-distributed policy in which sampling is controlled by the sensors, and update transmission by the monitor. Kadota et al. [12] develop and characterize transmission scheduling policies, and analyze the performance of these policies under various channel conditions. These findings are applied to improve the MAC layer implementation of Wi-Fi networks to improve real-world information freshness in wireless communication [19]. Our work differs from these above works in that it assumes that wireless nodes send data at arbitrary times out of the control of the monitor, and the monitor instead needs to switch channels to receive updates. In our work, we discuss multiple receiver-side channel switching policies, and characterize their performance under different monitoring scenarios.

Gu et al. [13] and Leng et al. [14] characterize AoI measurement and optimization in cognitive radio-based IoT, where multiple classes of IoT devices operate at different priorities and signal-to-noise ratios, competing for a shared medium. Similarly, Chen et al. [11] and Leng et al. [15] discuss AoI in the context of energy harvesting nodes, in which AoI minimization is in direct competition with energy constraints of the transmitting nodes. Our work considers a monitor that is not energy constrained, and arbitrary devices out of the control of the monitor. While we do not address transmission challenges of shared spectrum in cognitive radios, we envision that our work could be applied in that context.

Atay et al. [20] introduce the concept of "AoI regret" as the penalty of operating without the knowledge of certain system parameters needed to truly minimize AoI, and propose an algorithm that minimizes AoI regret faster than previously known algorithms. In our work, we focus on the case of a static population of devices exhibiting certain known transmission statistics (which may have been obtained by a preceding learning phase).

Zou et al. [16] propose a Markov Decision Process (MDP)-based formulation of heterogeneous multi-channel systems and scheduling algorithms that minimize total system AoI. Their work, as well as all previous work discussed, focuses on optimal *transmission* scheduling under certain constraints, whereas our work assumes lack of control over transmissions, and focuses on minimizing AoI based on scheduling the set of channels at the receiver side.

III. Problem Formulation

In this section, we define terminology used in this work, and introduce a general model for minimizing Age of Information (AoI) in multi-channel monitoring with the help of a Markov Decision Process (MDP).

A. Model

We consider a communication medium with C *channels*, indexed $c = 1, 2, \dots, C$, and D *devices*, indexed $d = 1, 2, \dots, D$. The sets of channels and devices are respectively denoted $\mathbf{C}$ and $\mathbf{D}$. Each device $d \in \mathbf{D}$ transmits on one channel $c \in \mathbf{C}$. Devices do not switch between channels.

We assume that each device d transmits packets following a Poisson process with rate λ_d, independent from the activity of other devices. Thus, the probability that device d transmits during a time slot Δt is

$$p_d = \Pr(d \text{ transmits within } \Delta t) = 1 - e^{-\lambda_d \Delta t}. \quad (1)$$

From now on, we thus consider a discretized time-slotted model, wherein the probability that device d transmits during a given slot is p_d, independent of any other event. The time slots are denoted $t = 1, 2, \dots, T$, where T is the time horizon.

Throughout this paper we assume a loss-free channel. Note that is possible for two devices to successfully transmit during the same slot Δt on the same channel, because the transmission time of a packet is assumed to be significantly shorter than the duration of a slot. The model can be generalized to a lossy channel, whereby p_d represents the probability that d transmits *and* the transmission is successfully received.

A *monitor* collects messages from the set of devices $\mathbf{D}$. The monitor can observe one *or more* contiguous channels at the same time, limited by the capabilities of the underlying hardware constraining its instantaneous bandwidth, i.e., the range of RF spectrum it can simultaneously capture. We denote by $\underline{\mathbf{C}}$ the set of channel tuples that the monitor can listen to and C_{max} the number of channels in each tuple. In each time slot, the monitor can only tune to one channel tuple. For instance, if there are $C = 4$ channels and the monitor can listen to $C_{max} = 2$ channels at a time, then $\underline{\mathbf{C}} = \{(1, 2), (2, 3), (3, 4)\}$.

The AoI of device d represents the timeliness of information available to the monitor based on the difference between the current time and the last time it received a packet from d. Thus, we denote $q_d(t) = t - u_d(t)$ the age of information of device d, where t is the current time slot and $u_d(t)$ is the last time slot in which the monitor received a packet from d.

Our work differs from most previous works that deal with AoI, in that the monitor has no control over the transmission behavior of the devices it observes. However, the monitor can select the channels on which to listen. Thus, at each time slot, the monitor needs to select a channel tuple $\underline{c} \in \underline{\mathbf{C}}$. The AoI of each device $d \in \mathbf{D}$ is updated as follows:

$$q_d(t+1) = \begin{cases} 0 & \text{if } d \text{ transmits and is on } c \in \underline{c} \\ q_d(t) + 1 & \text{otherwise.} \end{cases} \quad (2)$$

A common metric for measuring the freshness of the information is the sum of the AoIs averaged over all devices and time slots, that is,

$$J = \frac{1}{TD} \sum_{t=1}^{T} \sum_{d=1}^{D} \mathbb{E}[q_d(t)]. \quad (3)$$

The expectation is taken over the random realizations of packet transmissions by devices. We assume that the AoI of each device is initialized to 0, i.e., $q_d(1) = 0$ for all $d \in \mathbf{D}$. Note that Eq. (3) can easily be generalized to a weighted sum to capture devices of varying importance. For simplicity, in this paper, we assume that each device has the same weight.

B. Markov Decision Process (MDP)

Our goal is to determine the optimal policy that the monitor should follow to minimize J. We consider an infinite-horizon average reward problem, where $T \to \infty$, so we can focus only on stationary policies (which do not depend on t). A policy determines an action based on the current state. A state s of the system can be represented by the AoI of each device, i.e., $s = (q_1, q_2, \ldots, q_D)$. We drop the time index t, since it does not matter for stationary policies. Since each device always stays on the same channel, there is no need to add channel state variables (i.e., there is a fixed mapping between each device and the channel on which it resides).

A policy $\boldsymbol{\pi}$ determines an action a for each state s. In our case, the action a is the selection of a channel tuple $\underline{c} \in \underline{\mathbf{C}}$. We denote by $J^{\boldsymbol{\pi}}$ the AoI under policy $\boldsymbol{\pi}$. Thus, our goal is to determine the optimal policy $\boldsymbol{\pi}^*$ that minimizes $J^{\boldsymbol{\pi}}$, that is,

$$\boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}} J^{\boldsymbol{\pi}}. \quad (4)$$

A technicality is that the optimal solution π^* is generally guaranteed to exist only under a finite state space [21]. This can be addressed by truncating the state space such that the AoI of any device cannot exceed a certain value.

The optimization problem of Eq. (3) can be cast as an MDP, which can be solved using relative value iteration [21]. Computing the optimal policy is possible in small instances but becomes intractable for larger instances due to state-space explosion. Hence, in the next section, we present several scanning policies of low computational complexity.

IV. Scanning Policies

In this section, we introduce various scanning policies that are computationally feasible. We prove the optimality of several of these policies in a special case that reduces to a wireless broadcast scheduling problem studied in the literature [12].

A. General case

We consider the general case of a monitoring system with one or more devices per channel and a monitor capable of listening to one or more channels. We next present several policies that easily lend themselves to implementation.

For the purpose of illustrating the policies, consider a system with $C = 3$ channels and a monitor than can listen to $C_{max} = 2$ channels. Thus $\underline{\mathbf{C}} = \{(1, 2), (2, 3)\}$. Suppose that the system contains $D = 4$ devices with device 1 on channel 1, devices 2 and 3 on channel 2, and device 4 on channel 3. Last, assume that the transmission probability of the devices are $p_1 = 0.5, p_2 = 0.1, p_3 = 0.2, p_4 = 0.8$ and the AoI of the devices, at a certain time slot, are $q_1 = 4, q_2 = 2, q_3 = 3, q_4 = 3$.

a) Random: In the Random policy, the monitor selects a (contiguous) channel tuple $\underline{c}$ at random within the set $\underline{\mathbf{C}}$ in each time slot. This policy has the advantage of being very simple to implement since the monitor does not to keep track of the state of individual devices (i.e., it is stateless). However, it generally performs worse than stateful policies.

b) Greedy AoI (GAoI): The GAoI policy identifies the device with the highest AoI and the channel c on which this device operates. The GAoI policy then selects at random any channel tuple $\underline{c}$ that contains the channel c. Using the example above, device 1 has the highest AoI (equal 4), and therefore $c = 1$. As a result, the monitor selects the channel tuple $\underline{c} = (1, 2)$ (the only tuple that contains channel 1).

c) MaxSum: The MaxSum policy sums up the AoI of all the devices in each channel tuple, and selects the channel tuple with the largest sum. Denote by $\mathbf{D}_{\underline{c}}$ the set of devices operating on any channel belonging to channel tuple $\underline{c}$. Then, the sum of AoIs for tuple $\underline{c}$ is

$$\sum_{d \in \mathbf{D}_{\underline{c}}} q_d. \quad (5)$$

For the example above, the sum of AoIs in tuple $(1, 2)$ is 9, while for tuple $(2, 3)$ it is 8, resulting in selection of $(1, 2)$.

d) Greedy Expected Area (GEA): The GEA policy is the most sophisticated policy discussed, taking into account the transmission probability of devices. It also observes that Eq. (3) is in effect a form of area minimization. Indeed, the sum of ages of each device over time can be viewed as the sum of areas. In an interval between two AoI resets, if the AoI of a device reaches value Q, then the "area" contribution during this interval is $\sum_{q=0}^{Q} q = Q(Q + 1)/2$.

The GEA policy computes for each channel tuple $\underline{c}$ the area that one can expect to trim by the next time slot, that is

$$\sum_{d \in \mathbf{D}_{\underline{c}}} p_d(q_d + 1)(q_d + 2). \quad (6)$$

The GEA policy then selects the tuple $\underline{c}$ for which Eq. (6) is the largest. Back to our example, the expected area in tuple $(1, 2)$ is 20.2, while for tuple $(2, 3)$ it is 21.2. Hence, the monitor selects channel tuple $(2, 3)$.

B. Analysis

We next analyze the performance of the above policies in a special case.

Theorem 1. *Consider a monitoring system with one device per channel and a monitor that can listen to only one channel at a time, i.e., $C_{max} = 1$. Then, the optimization problem* *of Eq. (3) is identical to the optimization problem of Eq. (3) in [12].*

Proof. The work in [12] considers a single-hop wireless network with a base station (BS) sending information to M clients. In each slot, the BS transmits a packet to a selected client i over the wireless channel. The packet is successfully delivered to client i with probability p_i and a transmission error occurs with probability $1-p_i$. The AoI of client i is reset only if the BS transmits to client i and client i successfully receives the packet. Otherwise, the AoI increments by 1. Eq. (3) in [12] corresponds to the minimization of the AoI averaged over time and devices which is identical to Eq. (3) in our paper. Thus, for the special case of one device per channel and one channel to tune in, the problem of selecting which channel to listen to is the same as the problem of selecting which client to transmit to in [12]. In terms of notation, M and p_i in [12] respectively correspond to D and p_d in our model. □

The work in [12] proposes and analyzes several policies, namely the Random, Greedy, Max-Weight and Whittle's Index policies. The immediate corollary of Theorem 1 is that all the results obtained in [12] for these policies in the special case of one device per channel and a monitor capable of listening to only one channel at a time apply directly to our model. Specifically, in that special case, the GAoI and MaxSum policies are identical to the Greedy policy of [12], and the GEA policy is very similar to the Max-Weight and Whittle's Index policies of [12]. We can further establish the following result.

Theorem 2. *Consider a monitoring system with one device per channel and a monitor that can listen to only one channel at a time, i.e.,* $C_{max} = 1$. *Consider a symmetric network where* $p_d = p \in (0, 1]$ *for all* $d \in \mathbf{D}$. *Then, the GAoI, MaxSum, and GEA policies are all optimal.*

Proof. Theorem 5 in [12] establishes the optimality of the Greedy policy in the symmetric network case. The GAoI, MaxSum, and GEA policies in that case are all equivalent to the Greedy policy of [12], hence their optimality follows. □

V. Evaluation

We evaluate the performance of the various policies by making use of two experimental studies. The first experimental setup consists of an MDP, solved using the Python Markov Decision Toolbox [22]. This setup allows for the creation of arbitrary monitoring scenarios based on definition of the MDP state space, action space, transition model and reward model. Based on this model, the optimal policy can be computed from scratch by means of relative value iteration. Crucially, it can also be used to compare the optimal policy with the scanning policies introduced in Section IV. The second experimental setup is based on real-world traces of wireless traffic collected in over-the-air experiments with smart home devices by Gvozdenovic et al. [3], [4]. Based on these time series, we can simulate a multi-channel monitor and characterize the performance of the various policies.

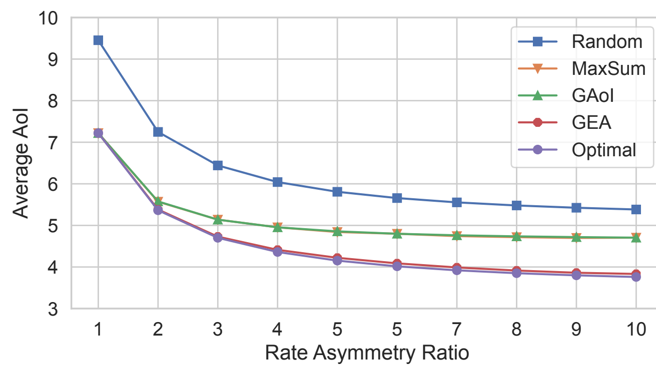

Fig. 1. Two channels with one device on each with transmission rates λ_1 and λ_2. The monitor scans a single channel at a time. The *rate asymmetry ratio* λ_2 / λ_1 on the x-axis represents the ratio of the increasingly faster rate of device 2 on channel 2 over the constant rate of device 1 on channel 1 ($\lambda_1 = 0.2$). The GEA policy performs close to optimal in all scenarios. The GAoI and MaxSum policies perform optimally in the symmetric case but become worse under strongly asymmetric conditions.

A. Evaluation of policies and comparison to optimal

a) Two channels, one device each: First, we implement an MDP model with two channels, each of which is occupied by one device. Device 1 has a message generation rate of $\lambda_1 = 0.2$, while the message generation rate of device 2 increments in steps of 0.2 in the range $\lambda_2 = [0.2 \ldots 2.0]$, such that the message generation varies from a symmetric scenario where $\lambda_2 = \lambda_1$ to an asymmetric scenario with a factor of 10 between the two devices, i.e., $\lambda_2 = 10 \times \lambda_1$. We refer to the ratio λ_2/λ_1 as the *rate asymmetry ratio*. We assume a time slot of length 1, i.e., $\Delta t = 1$. We truncate the state space to a maximum AoI of 20, resulting in 882 distinct states in the MDP, then solve it numerically using relative value iteration to obtain an optimal solution. The algorithm stops when an ϵ-optimal policy for a stopping criterion of $\epsilon = 0.01$ has been found, or 1000 iterations have been exhausted. Subsequently, we apply the policies introduced in Section IV to the model. Then, we calculate the stationary distribution of the Markov chain and calculate its AoI.

The results are shown in Figure 1. Per Theorem 1, the setup considered in this case is equivalent to the scheduling problem considered in [12]. All the policies, except for the random one, achieve optimal performance in the symmetric case $\lambda_2 = \lambda_1$, as predicted by Theorem 2 As asymmetry increases, however, the Greedy AoI and MaxSum algorithms fall behind significantly, whereas the GEA policy performs near-optimal throughout.

b) Two channels, two devices each: In the case of multiple devices per channel, we distinguish between variations of *device asymmetry* (DA) (i.e., two channels with similar compositions of slow and fast devices) and *channel asymmetry* (CA) (i.e., one channel containing slow devices and another containing increasingly fast devices). In the following

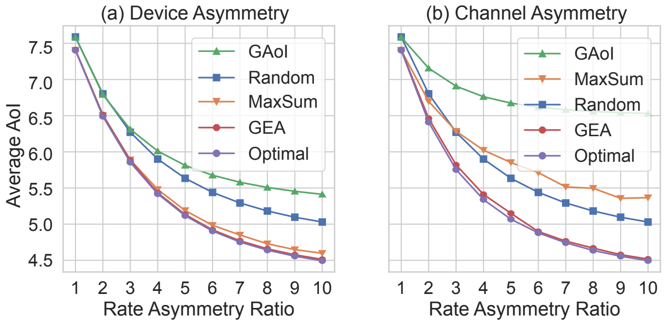

Fig. 2. Two channels with two devices on each. The monitor scans a single channel at a time.
(a) *device asymmetry*: On each channel, one device operates at a constant rate, and the other at an increasingly faster rate. The *rate asymmetry ratio* is the ratio of the faster rate over the slower rate. The GEA and MaxSum policies both perform close to optimal.
(b) *channel asymmetry*: On channel 1, both devices transmit at the same constant rate, and on channel 2, the devices transmit at an increasingly faster rate. Here, the GEA policy performs significantly better than the other policies.

experiments, devices 1 and 2 are on channel 1, and 3 and 4 on channel 2. We again assume $\Delta t = 1$. The ratio of the faster rate over the slower rate is the *rate asymmetry ratio* (RAR).

In the DA case, shown in Figure 2(a), we consider two statistically similar channels. In each channel, one device stays at the same transmission rate, i.e., $\lambda_1 = \lambda_3 = 0.2$, while the traffic rate of the other device increases, i.e., $\lambda_2 = \lambda_4 = [0.2 \ldots 2.0]$, resulting in a RAR ranging from 1 to 10. To limit the size of the state space, the AoI in the MDP model is truncated to a value of 5, resulting in 1250 states (and a 2592×2592 transition matrix). Note that while a higher maximum value of AoI would be desirable, the problem quickly becomes computationally infeasible. Under the DA conditions, the GEA and MaxSum perform close to optimal, while GAoI trails behind with increasing asymmetry.

In the CA case, we consider one channel with two devices sending at rates of $\lambda_1 = \lambda_2 = 0.2$, and another channel with two devices with rates $\lambda_3 = \lambda_4 = [0.2 \ldots 2.0]$, again result in a RAR from 1 to 10. As shown in Figure 2(b), the GEA policy again performs close to optimal under any of these conditions, while GAoI and MaxSum deteriorate even beyond random channel selection under more pronounced channel asymmetry. Interestingly, the MaxSum algorithm behaves significantly different in the DA and CA scenarios. This can be explained by its preference for the slower channel in the CA scenario, in which it needlessly allows the "fast" channel to accumulate age while waiting for slow devices to update.

c) Multi-channel monitoring: We consider a multi-channel scanning scenario with four channels, and a monitor that can simultaneously monitor two channels at a time (i.e., $\underline{\mathbf{C}} = \{(1,2),(2,3),(3,4)\}$). We assume one device on each channel. The device on channel 1 transmits in each time slot ($p_1 = 1.0$), while the devices on all other channels transmit with probability $p_2 = p_3 = p_4 = \alpha$, where α decreases from 1 down to 0.1. As shown in Figure 3, GEA performs near optimally, while the performance of the MaxSum policy degrades as the channels become more asymmetric.

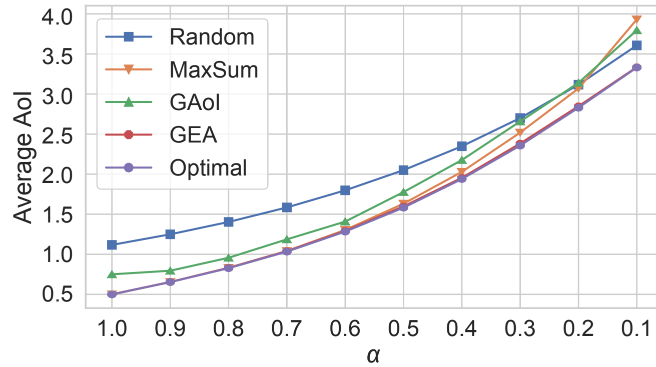

Fig. 3. A scenario with four channels, one device on each, and transmission probabilities $[1, \alpha, \alpha, \alpha]$. The monitor scans two channels simultaneously. The GEA policy performs close to optimal.

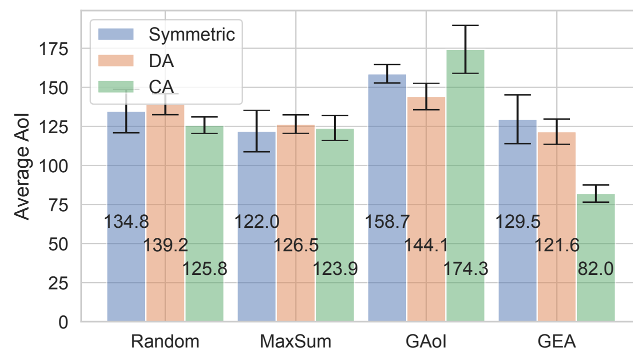

Fig. 4. A monitor scanning three out of 10 channels simultaneously. There are two devices on each channel. The GEA policy performs by far the best under asymmetric channel conditions (CA), while under other conditions, GEA and MaxSum perform slightly better than Random and much better than GAoI.

B. Trace-based simulation

The second experimental setup is based on real-world traces provided by Gvozdenovic et al. [3], [4]. The traces were obtained by recording Zigbee and Bluetooth LE traffic in a controlled environment for 10 minutes each. The resulting PCAP files are converted into anonymized time series. This measurement campaign resulted in 48 trace files containing at least two devices, and 28 trace files containing 6 or more devices. In contrast to the MDP model, running the simulation model allows for larger channel and device configurations, as the entire state transition model does not have to be computed.

We consider 10 channels with two devices on each, and a monitor simultaneously scanning three channels at a time. We construct three scenarios:

Symmetric: 10 traces are randomly selected from the available trace files without replacement and used as channels.

DA: 20 traces are randomly selected from the available traces without replacement. For each channel, the device with the slowest and fastest transmission rate are selected (these statistics are contained in the trace files). The RAR for these device pairs is computed for each trace, and the 10 traces with the highest ratio are used as channels.

CA: 20 traces are randomly selected from the available traces without replacement. Traces are sorted by the mean device transmission rate (contained in the trace files). The five traces with the lowest and highest mean transmission rate respectively are selected as the 10 channels.

Each of these scenarios is repeated independently 20 times, and the results averaged.

These trace-based results are summarized in Figure 4. The GEA policy outperforms other policies significantly under channel asymmetry, i.e., when devices on some channels update at a significantly slower rate than devices on other channels. This is the case in multi-protocol scanning, where transmission rates across observed protocols can span orders of magnitude [3]. In the other scenarios, MaxSum and GEA have a slight advantage over random channel selection, whereas GAoI performs noticeably worse. This is a result of GAoI selecting individual channels containing the highest-AoI devices, then randomly selecting tuples containing that channel, whereas MaxSum and GEA compute a tuple selection based on the the highest scoring group of channels.

VI. Conclusion

We presented a new perspective on AoI optimization for a passive monitor which cannot influence the arbitrary data sources it is monitoring, relying purely on channel switching policies. We formulated this problem using MDPs in order to comparatively study different single and multi-channel scanning policies. We further compared policy performance using real IoT device traffic.

While the MDP can be solved in small instances, it becomes intractable as the number of devices increases. To address this, we proposed and evaluated several policies, using both synthetic and real traffic traces. We showed that a policy, coined Greedy Expected Area (GEA), performs well in many of the scenarios. In particular, under channel asymmetry, GEA significantly outperforms all other discussed policies. For small instance, we also showed that GEA performs near the optimal solution solved using the MDP. We also formally proved the optimality of GEA, GAoI, and MaxSum in the special case of a symmetric network with a single device per channel and single channel monitoring.

The traces used in this work have been made available to the research community [4], so that researchers can propose and compare other algorithms. Insight from these works may lead to new discoveries on policies which perform well in large constellations of channels and devices, which has implications for monitoring large and crowded bands, such as the 2.4 GHz band, across multiple channels and IoT protocols.

Acknowledgements

This research was supported in part by the US National Science Foundation under grants CNS-1717858, CNS-1908087, CCF-2006628, EECS-2128517, and by an Ignition Award from Boston University.

References

[1] Fortune Business Insights, "Software Defined Radio Market Size, Share & COVID-19 Impact Analysis," Tech. Rep., 2021. [Online]. Available: https://www.fortunebusinessinsights.com/software-defined-radios-market-102524

[2] Global Industry Analysts Inc., "Global Software Defined Radio (SDR) Market to Reach $23.3 Billion by 2026," 2021. [Online]. Available: https://www.prnewswire.com/news-releases/global-software-defined-radio-sdr-market-to-reach-23-3-billion-by-2026–301350042.html

[3] S. Gvozdenovic, J. K. Becker, J. Mikulskis, and D. Starobinski, "Multi-Protocol IoT Network Reconnaissance," in *2022 IEEE Conference on Communications and Network Security (CNS)*.

[4] S. Gvozdenovic, J. K. Becker, J. Mikulskis, and D. Starobinski. (2022) IoT-Scan Traces. [Online]. Available: https://github.com/nislab/iot-scan

[5] R. D. Yates, Y. Sun, D. Richard Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of Information: An Introduction and Survey," *IEEE J. Sel. Areas Commun.*, pp. 1–30, 2021.

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

[7] V. Tripathi and S. Moharir, "Age of Information in Multi-Source Systems," in *IEEE Global Telecommun. Conf.*, 2017, pp. 1–6.

[8] B. Sombabu and S. Moharir, "Age-of-Information Based Scheduling for Multi-Channel Systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4439–4448, 2020.

[9] B. Zhou and W. Saad, "Joint status sampling and updating for minimizing age of information in the internet of things," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7468–7482, 2019.

[10] ——, "Minimum age of information in the internet of things with non-uniform status packet sizes," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1933–1947, 2020.

[11] Z. Chen, N. Pappas, E. Bjornson, and E. G. Larsson, "Age of Information in a Multiple Access Channel with Heterogeneous Traffic and an Energy Harvesting Node," *INFOCOM Workshops*, pp. 662–667, 2019.

[12] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, 2018.

[13] Y. Gu, H. Chen, C. Zhai, Y. Li, and B. Vucetic, "Minimizing Age of Information in Cognitive Radio-Based IoT Systems: Underlay or Overlay?" *IEEE Internet Things J.*, vol. 6, no. 6, Dec. 2019.

[14] S. Leng and A. Yener, "Age of Information Minimization for Wireless Ad Hoc Networks: A Deep Reinforcement Learning Approach," in *IEEE Global Telecommun. Conf.* IEEE, Dec. 2019, pp. 1–6.

[15] ——, "Minimizing Age of Information for an Energy Harvesting Cognitive Radio," in *IEEE Wireless Commun. Netw. Conf.*, vol. 2019-April, no. 2, Apr. 2019, pp. 427–439.

[16] Y. Zou, K. T. Kim, X. Lin, and M. Chiang, "Minimizing Age-of-Information in Heterogeneous Multi-Channel Systems," in *ACM MobiHoc '21*. ACM, Jul. 2021, pp. 11–20.

[17] M. Alsliety and D. N. Aloi, "Signal processing choices and challenges for SDR in Telematics," in *Int. Symp. Signal Process. Appl.*, no. 2. IEEE, Feb. 2007, pp. 1–4.

[18] L. Choong, "Multi-Channel IEEE 802.15.4 Packet Capture Using Software Defined Radio," *UCLA Networked & Embedded Sensing Lab,*, vol. 03, pp. 1–20, 2009.

[19] I. Kadota, M. S. Rahman, and E. Modiano, "WiFresh: Age-of-Information from Theory to Implementation," *Int. Conf. Comput. Commun. Netw.*, pp. 1–13, 2021.

[20] E. U. Atay, I. Kadota, and E. Modiano, "Aging bandits: Regret analysis and order-optimal learning algorithm for wireless networks with stochastic arrivals," *arXiv*, 2020.

[21] D. Bertsekas, *Dynamic programming and optimal control: Volume I.* Athena Scientific, 2012, vol. 1.

[22] I. Chadès, G. Chapron, M.-J. Cros, F. Garcia, and R. Sabbadin, "MDPtoolbox: a multi-platform toolbox to solve stochastic dynamic programming problems," *Ecography*, vol. 37, no. 9, pp. 916–920, Sep. 2014.