

# Semantics-Aware Active Fault Detection in IoT

George J. Stamatakis\*, Nikolaos Pappas†, Alexandros Fragkiadakis\*, Apostolos Traganitis\*

\* Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH)

† Department of Computer and Information Science, Linköping University, Linköping, Sweden

E-mails: {gstam, alfrag, tragani}@ics.forth.gr, nikolaos.pappas@liu.se

**Abstract**—In this work we address a problem of active fault detection in an IoT scenario, whereby a monitor probes a remote device in order to detect faults and acquire fresh information. However, probing can have a significant impact on the IoT network’s energy and communication resources. To address this problem we utilize Age of Information as a measure of the freshness of information at the monitor and adopt a semantics-aware communication approach between the monitor and the remote device. In semantics-aware communications, the processes of generating and transmitting information are treated jointly in order to consider the importance of information and the purpose of communication. We formulate the problem as a Partially Observable Markov Decision Process and propose a computationally efficient stochastic approximation algorithm to approximate the optimal policy. Finally, we present numerical results that exhibit the advantage of our approach compared to a conventional delay-based probing policy.

**Index Terms**—Semantics-aware communications, Internet of Things, Fault Detection, Partially Observable Markov Decision Process

## I. INTRODUCTION

The emergence of massive IoT ecosystems poses new challenges for their maintenance procedures. IoT networks are characterized by software, hardware, and communication protocols’ diversity. Furthermore, they are typically comprised of a large number of devices that are often deployed in remote and harsh environments. In this context, the development of autonomous fault detection procedures is necessary to safely and efficiently operate an IoT network. The majority of fault detection algorithms that have been proposed in the past [1], assume that the system is passively monitored and utilize statistical or machine learning techniques to infer the actual health status of its subsystems. However, a major drawback with passive monitoring is that faults can pass undetected if the faulty and the nominal operation overlap due to measurement and process uncertainties or in cases where control actions mask the influence of faults [2]. To address this problem we make use of an *active* fault detection scheme that utilizes probes to affect the system’s response and thus to increase the probability of detecting certain faults.

With active fault detection special care must be taken so that the extra network traffic due to probing is not detrimental to the system’s performance. Blindly generating and transmitting probes could increase network congestion and prohibit other applications from satisfying their possibly strict real-time constraints. To this end, we adopt a *semantics-aware* [3]–[12] approach to active fault detection which has exhibited its

ability to eliminate the transmission of redundant and uninformative data and thus minimize the induced overhead. Within the context of *semantics-aware* communications the generation and transfer of information across a network are considered jointly in order to take into account the goal or purpose of the communication. What is more, the *importance/significance* of a communication event, i.e., the event of generation and transmission of information, constitutes the decisive criterion of whether it should take place or not. The definition of the importance of a communication event is application-specific, thus, in the context of active fault detection, we define it to be a function of the freshness of information that has been received from the remote device and of the operational status of the communication network and the remote device.

More specifically, in this work, we consider a basic active fault detection scenario for a discrete-time dynamic system that is comprised of a sensor and a monitor. At the beginning of each time slot, the sensor probabilistically generates and transmits status updates to the monitor over an unreliable link while the monitor decides whether or not to probe the sensor for a mandatory transmission of a fresh status update through a separate unreliable link. By the end of each time slot, the monitor may or may not receive a status update either because none was generated at the sensor or due to intermittent faults at the sensor and the wireless links. To detect intermittent faults, the monitor maintains a belief vector, i.e., a probability distribution, over the operational status (healthy or faulty) of the system and a measure of its confidence in this belief vector that is expressed by the entropy of the belief vector. Probing, successfully or unsuccessfully, increases the confidence of the monitor in its belief state, however, it also induces a cost for the monitor that measures the negative impact of probing on the system’s energy and communication resources. Our objective is to find a policy that decides at each time slot whether or not a probe should be sent to the sensor so that it optimally balances the probing cost with the need for fresh information at the monitor.

Our approach to solving this problem is to formulate it as a Partially Observable Markov Decision Process (POMDP) and derive the necessary conditions for probing to result in a reduction of the belief state’s entropy. To the best of our knowledge, this is the first work with this approach. Our analysis indicates that there exist probing cost values such that the optimal policy is of a *threshold* type. In addition, we propose a stochastic approximation algorithm that can compute such a policy and, subsequently, evaluate the derived

policy numerically.

### A. Related work

Fault detection methods can be categorized as passive, reactive, proactive, and active. Passive fault detection methods collect information from the data packets that the wireless sensors exchange as part of their normal operation whereas in reactive and proactive fault detection methods the wireless sensors collect information related to their operational status and subsequently transmit it to the monitor. Finally, in active fault detection methods the monitor probes the wireless sensors for information specific to the fault detection process. In Wireless Sensor Networks (WSNs) the fault detection algorithms being cited in recent works [1], [13] fall in the passive, reactive, and proactive categories, with the majority of them being passive fault detection algorithms. In [14] and [15] the authors adopted an active approach to fault detection in WSNs. However, both of these tools were meant for pre-deployment testing of WSNs software rather than a health status monitoring mechanism.

Unlike these works, we propose an active fault detection method for continuously monitoring the health status of sensors. We believe that autonomous active fault detection methods can successfully complement the passive ones by addressing their limitations. More specifically, passive fault detection methods often fail to detect faults because the faulty and the nominal operation overlap due to measurement and process uncertainties. What is more, network control mechanisms specifically designed to increase the robustness of IoT networks, e.g., by delegating the job of a sensor to neighboring or redundant nodes, often compensate for the performance degradation due to intermittent faults and thus mask their influence rendering them undetectable [2]. Acknowledging the fact that the network overhead due to active fault detection can be prohibitive, we adopted the semantics-aware communication paradigm [3], [4], [6]–[12], [16] which has exhibited its ability to eliminate the transmission of redundant and uninformative data and thus minimize the induced overhead. Active fault detection methods have also been studied in the context of wired networks [17]–[22]. However, the operational conditions of wired networks differ considerably from these of WSNs in terms of protocols, energy, bandwidth, and transmission errors thus the proposed techniques for wired networks cannot be applied in the context of IoT networks.

## II. SYSTEM MODEL

We consider the system presented in Figure 1 that is comprised of a sensor that transmits status updates to a monitoring device over the wireless link  $l_{SM}$  and the monitoring device, that is able to probe the sensor for a fresh status update over the wireless link  $l_{MS}$ . Transmissions over the  $l_{MS}$  and  $l_{SM}$  links are subject to failure with failures being independent between the two links. We assume that time is slotted and indexed by  $t \in \mathbb{Z}^+$ . The state of the sensor is modeled as an independent two state time-homogeneous Markov process. Let  $F_t^S \in \{0, 1\}$  be the state of the sensor's Markov process at the beginning of the  $t$ -th time slot. When  $F_t^S$  has a value of 0/1, the  $i$ -th

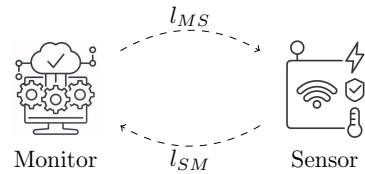


Fig. 1: Basic IoT setup.

sensor's operational status is healthy/faulty. We assume that the sensor will remain in the same state for the duration of a time-slot and, afterwards, it will make a probabilistic transition to another state as dictated by the state transition probability matrix  $P^S$ . Furthermore, at the beginning of each time slot the sensor will generate a status update with probability  $P_g$ , when in a healthy state, while it will not generate a status update when in a faulty state. In this work we assume that  $P_g < 1$ , otherwise the probing system is redundant. In the case of a status update generation the sensor will transmit it over the  $l_{SM}$  link. At the end of the time slot the status update is discarded independently of the outcome of the transmission.

Similarly, we model the health status of the wireless links as two independent two-state time-homogeneous Markov processes. Let  $F_t^{MS}, F_t^{SM} \in \{0, 1\}$  denote the state of the independent Markov processes for the  $l_{MS}$  and  $l_{SM}$  links respectively, at the beginning of the  $t$ -th time slot. When  $F_t^{MS}$  and  $F_t^{SM}$  take a value of 0/1 the operation of the respective wireless link is healthy/faulty. We assume that the wireless links will remain in the same state for the duration of a single time-slot and, subsequently, they will make a transition to another state as dictated by the transition probability matrices  $P^{MS}$  and  $P^{SM}$  respectively. When in a healthy state, the wireless link will forward successfully a status update to the monitor with probability 1, whereas a faulty wireless link will always fail to deliver the status update. Finally, following a successful reception of a probe through the  $l_{MS}$  link, the sensor will generate a fresh status update at the next time slot with probability 1, if it is in a healthy state; and with probability 0 if it is in a faulty state.

In this work we consider the problem of a monitoring agent that must optimally decide, at the beginning of each time slot, whether or not to probe the sensor. As a result of its decision and the dynamics of the system a transition cost is induced on the agent by the end of each time slot. The agent's objective is to minimize the total cost accumulated over a finite time horizon. The transition cost is a function of the agent's confidence in its belief about the joint health status of the sensor and the  $l_{SM}$  link, the staleness of the status updates it has received up to that time slot and a cost value  $c$  associated with the probing action. More specifically, since the agent cannot observe the actual health status of the sensor and of the  $l_{SM}$  link, it maintains a belief vector for their joint health status, i.e., a probability distribution over two possible events. The first event occurs when both of them are healthy and the second event occurs when at least one of them is faulty.

Furthermore, the agent's confidence in the health status belief vector is expressed by its entropy denoted with  $H_t$ . Details regarding the definition of the health status belief vector and its entropy  $H_t$  are presented in Section III.

Furthermore, to characterize the staleness of the status updates received at the monitor we utilize the Age of Information (AoI) metric that has received significant attention in the research community [23]–[31]. AoI was defined in [32] as the time that has elapsed since the generation of the last status update that has been successfully decoded by the destination, i.e.,  $\Delta(t) = t - U(t)$ , where  $U(t)$  is the time-stamp of the last packet received at the destination at time  $t$ . We use  $\Delta_t$ ,  $t = 0, 1, \dots, N$ , to denote the AoI of the sensor at time  $t$ . However, as the time horizon of the optimal probing problem increases  $\Delta_t$  could assume values that would be disproportionately larger than  $H_t$ . To alleviate this problem we will use a normalized value of the AoI which we define as,  $\bar{\Delta}_t = \frac{\Delta_t}{N}$ , where  $N$  is the length of the finite horizon measured in time-slots.

Finally, we define the Value-of-Information (VoI), i.e., a metric that quantifies the importance of receiving a fresh status update at the monitor at time  $t$ , as,

$$V_t = \lambda_1 H_t + \lambda_2 \bar{\Delta}_t, \quad (1)$$

where  $\lambda_1$  and  $\lambda_2$  are weights that determine the relative value of each component of the metric.

Finally, although it seems intuitive that probing will lead to the reduction of both  $\bar{\Delta}_t$  and entropy  $H_t$ , this is not always the case especially for the latter one. While probing makes the generation of a status update mandatory, i.e., it reduces the uncertainty induced in the system due to the probabilistic generation of status updates, one should also consider that probing introduces a new type of uncertainty due to the transmission failures occurring in the  $l_{MS}$  link. As an example consider the case where a probe was sent to the sensor yet no status update arrives at the monitor. It is not certain whether this happened because the probe didn't actually reach the sensor, due to a faulty  $l_{MS}$  link, or because the sensor did not generate a status update, due its faulty state, or because the transmission of the status update failed, due to a faulty  $l_{SM}$  link. It is possible to express in a concise way the necessary conditions for probing to always result in the reduction of entropy [33] for in the system we consider. In this work we assume that these necessary conditions are satisfied.

### III. PROBLEM FORMULATION

In this section we formulate the decision problem presented above as a Partially Observable Markov Decision Process (POMDP) denoted with  $\mathcal{P}$ . A POMDP model with a finite horizon  $N$  is a 7-tuple  $(\mathcal{S}, \mathcal{A}, \mathcal{Z}, P, r, g, g_N)$ , where  $\mathcal{S}$  is a set of states,  $\mathcal{A}$  is a set of actions,  $\mathcal{Z}$  is a set of possible observations,  $P$  is a probability matrix representing the conditional transition probabilities between states,  $r$  represents the observation probabilities,  $g$  is the transition reward function and  $g_N$  is the terminal cost incurred at the last decision stage. In the remaining part of this section we present the

individual elements of  $\mathcal{P}$  and formulate the corresponding dynamic program based on a belief state formulation [34].

**State Space ( $\mathcal{S}$ ):** At the beginning of the  $t$ -th time-slot the health state of the system is represented by the column vector,  $s_t = [F_t^{MS}, F_t^S, F_t^{SM}]^T$  where,  $F_t^{MS}$ ,  $F_t^S$  and  $F_t^{SM}$  were defined in Section II.

**Actions ( $\mathcal{A}$ ):** The set of actions available to the agent is denoted with  $\mathcal{A} = \{0, 1\}$ , where 0 represents the no-probe action and 1 indicates the probe action. The action taken by the agent at the beginning of the  $t$ -th time slot is denoted with  $a_t \in \mathcal{A}$ .

**Random variables:** The state of the system presented in Fig. 1 will change stochastically at the beginning of each time slot. The transition to the new state is governed by the action taken by the monitor, the transition probability matrices  $P^{MS}$ ,  $P^S$  and  $P^{SM}$ , and the following random variables. The random variable  $W_t^g \in \{0, 1\}$  that represents the random event of a status update generation at the  $t$ -th time slot. If a status update is generated by the sensor at  $t$  then  $W_t^g$  takes the value 1 and if the sensor does not generate a status update at  $t$  then  $W_t^g$  takes the value 0. We have the following conditional distribution for  $W_t^g$

$$P[W^g = 0 | F^S, a] = \begin{cases} 1 - P_g, & \text{if } a = 0 \text{ and } F^S = 0, \\ 1, & \text{if } a = 0 \text{ and } F^S = 1, \\ 0, & \text{if } a = 1 \text{ and } F^S = 0, \\ 1, & \text{if } a = 1 \text{ and } F^S = 1 \end{cases}$$

and  $P[W^g = 1 | F^S, a] = 1 - P[W^g = 0 | F^S, a]$ , where we omitted the time index since the distribution is assumed to remain constant over time.

The random variable  $W_t^{MS} \in \{0, 1\}$  represents the random event of a successful transmission over the  $MS$  link during the  $t$ -th time slot. A value of 0 indicates an unsuccessful transmission over the link and a value of 1 indicates a successful transmission. The conditional probability distribution for  $W_t^{MS}$  is given by,

$$P[W^{MS} = 0 | F^{MS}, a] = \begin{cases} 1, & \text{if } a = 0, \\ 0, & \text{if } a = 1 \text{ and } F^{MS} = 0, \\ 1, & \text{if } a = 1 \text{ and } F^{MS} = 1 \end{cases}$$

and  $P[W^{MS} = 1 | F^{MS}, a] = 1 - P[W^{MS} = 0 | F^{MS}, a]$ , where again we omitted the time index  $t$ . Finally, the random variable  $W_t^{SM} \in \{0, 1\}$ , represents the random event of a successful transmission over the  $SM$  link during the  $t$ -th time slot. A value of 0 indicates an unsuccessful transmission over the link and a value of 1 indicates a successful transmission. The conditional probability distribution for  $W^{SM}$  is given by,

$$P[W^{SM} = 0 | W^g, F^{SM}] = \begin{cases} 1, & \text{if } W^g = 0, \\ 0, & \text{if } W^g = 1 \text{ and } F^{SM} = 0, \\ 1, & \text{if } W^g = 1 \text{ and } F^{SM} = 1, \end{cases}$$

and  $P[W^{SM} = 1 | W^g, F^{SM}] = 1 - P[W^{SM} = 0 | W^g, F^{SM}]$ .

**Transition probabilities ( $P$ ):** Let  $m$  be an index over the set of the three subsystems presented in Fig. 1, i.e.,

$m \in \{MS, S, SM\}$ , then the transition probability matrices  $P^{MS}$ ,  $P^S$  and  $P^{SM}$  can be defined as follows,  $P^m = \begin{bmatrix} p_{00}^m & p_{01}^m \\ p_{10}^m & p_{11}^m \end{bmatrix}$ , where  $p_{00}^m$  represents the probability to make a transition from a healthy state (0) to a healthy state (0) for subsystem  $m$ . Transition probabilities  $p_{01}^m$ ,  $p_{10}^m$ , and  $p_{11}^m$  are defined in a similar way. Furthermore, we introduce the shorthand notation  $s = [s_0, s_1, s_2]$  and  $s' = [s'_0, s'_1, s'_2]$  for states  $s_t = [F_t^{MS}, F_t^S, F_t^{SM}]^T$  and  $s_{t+1}$  respectively so that the conditional probability distribution of state  $s'$  given the current state  $s$  can be expressed as,  $P[s_{t+1} = s' | s_t = s] = p_{s_0 s'_0}^{MS} \cdot p_{s_1 s'_1}^S \cdot p_{s_2 s'_2}^{SM}$ .

**Observations ( $Z$ ):** At the beginning of each time slot the agent observes whether a status update arrived or not. Let  $z_t \in \{0, 1\}$  denote the observation made at the  $t$ -th time slot, with 0 representing the event that no status update was received and 1 representing the event that a status update was received. We define  $r_s(a, z)$  as the probability to make observation  $z$  at the  $t$ -th time slot, i.e.,  $z_t = z$ , given that the system is in state  $s$ , i.e.,  $s_t = s$ , and the preceding action was  $a$ , i.e.,  $a_{t-1} = a$ . Thus we have,  $r_s(a, z) = P[z_t = z | s_t = s, a_{t-1} = a]$ . By utilizing the conditional probability distributions presented in Section III we derived [33] the observation probabilities for all possible combinations of states and controls and present them in Table I.

TABLE I: Observation probabilities  $r_s(a, z)$ .

$i$	$F_t^{MS}$	$F_t^S$	$F_t^{SM}$	$a_{t-1} = 0$		$a_{t-1} = 1$	
				$z_t = 0$	$z_t = 1$	$z_t = 0$	$z_t = 1$
0	0	0	0	$1 - P_g$	$P_g$	0	1
1	0	0	1	1	0	1	0
2	0	1	0	1	0	1	0
3	0	1	1	1	0	1	0
4	1	0	0	$1 - P_g$	$P_g$	$1 - P_g$	$P_g$
5	1	0	1	1	0	1	0
6	1	1	0	1	0	1	0
7	1	1	1	1	0	1	0

The evolution of the AoI value over time depends on the observation made by the agent and,

$$\Delta_t = \begin{cases} 1, & \text{if } z_t = 1 \\ \min\{N, \Delta_t + 1\}, & \text{if } z_t = 0 \end{cases} \quad (2)$$

where  $N$  is the finite time horizon of the optimization problem.

**Transition cost function:** At the end of each time slot, the agent is charged with a cost that depends on the VoI and the action taken by the agent as follows,  $g_t = c \cdot \mathbb{1}_{\{a_t=1\}} + V_t$ , where,  $\mathbb{1}_{\{a_t=1\}}$  is the indicator function which takes a value of 1 when the probe action was taken by the agent and a value of zero otherwise, and  $V_t$  is computed using Equation (1). Parameter  $c$  is the cost value associated with probing and quantifies the consumption of system resources for the generation and transmission of a probe.

**Belief State:** At each time slot  $t$  the agent maintains a belief state  $P_t$ , i.e., a probability distribution over all possible system states,  $P_t = [p_t^0, \dots, p_t^7]^T$ . Starting from an arbitrarily initialized belief state  $P_0$  the agent updates its belief about the

actual state of the system at the beginning of each time slot as follows,

$$p_{t+1}^j = \frac{\sum_{i=0}^7 p_t^i \cdot p_{ij} \cdot r_j(a, z)}{\sum_{s=0}^7 \sum_{i=0}^7 p_t^i \cdot p_{is} \cdot r_s(a, z)}, \quad (3)$$

where  $p_{ij} = P[s_{t+1} = j | s_t = i]$ . In the literature  $p_{ij}$  is usually a function of the action selected at time  $t$ , i.e.,  $p_{ij}(a_t)$ , however, in our case the actions taken by the agent do not affect the system's state. In any case, the action taken by the agent affects the observation  $z_{t+1}$  made by the agent and thus directly affects the evolution of the belief state over time. As mentioned in Section I, based on  $P_t$  the agent forms the *health status belief vector*  $P_t^h$  that represents our belief regarding the health status of the sub-system comprised of the sensor and the  $l_{SM}$  link. We have,  $P_t^h = [p_t^h, p_t^f]$ , where  $p_t^h$  and  $p_t^f$  represent, respectively, the probabilities for the sub-system to be in a healthy or faulty state. We define  $p_t^h = p_t^0 + p_t^4$ , since states with index 0 and 4 in Table I are the only states where both the sensor and the  $l_{SM}$  link are in a healthy state. Correspondingly, we define  $p_t^f = \sum_{i \neq 0, 4} p_t^i$ ,  $i = 0, \dots, 7$ . It holds that  $P_t^h$  is a probability distribution since  $p_t^h$  and  $p_t^f$  are computed over complementary subsets of the system's state space and  $P_t$  is a probability distribution. Finally, the health status entropy is computed as  $H_t = -[p_t^h \cdot \log_2(p_t^h) + p_t^f \cdot \log_2(p_t^f)]$ .

For the agent to have all the information necessary to proceed with the decision process it must also keep the value of the AoI as part of its state, thus we augment the belief state with the value of AoI and define the following representation of the current state, i.e.,  $x_t = [P_t, \Delta_t]$ , and define  $X$  to be the set of all states.

**Dynamic program of  $\mathcal{P}$ :** By utilizing the belief state formulation and for a finite horizon  $N$  the optimal policy  $\pi^* : X \rightarrow \mathcal{A}$  can be obtained by solving the following dynamic program,

$$J_t(x_t) = \min_{a_t \in \{0, 1\}} [g_t + \sum_z \sum_s \sum_i p_t^i p_{is} r_s(a_t, z) J_{t+1}(x_{t+1})] \quad (4)$$

for all  $x_t \in X$  and  $t = 0, 1, \dots, N$ , where  $x_{t+1} = [P_{t+1}^{a,z}, \Delta_{t+1}^z]$ ,  $z \in \{0, 1\}$ ,  $s, i \in \{0, 1, \dots, 7\}$  and the terminal cost is given by  $J_N(x_N) = g_N$ . It is known that for (4) there do exist optimal stationary policies [34], i.e.,  $\pi^* = \{\pi_0^*, \pi_1^*, \dots, \pi_{N-1}^*\}$ . However, since the state space  $X$  is uncountable the recursion in (4) does not translate into a practical algorithm. Nevertheless, based on (4) we proved that the optimal policy is of a threshold type [33] and thus it can be computed efficiently.

#### IV. ANALYSIS

In this section we present structural results for the optimal policy of the POMDP  $\mathcal{P}$  defined in the previous section. We represent the belief state at the  $(t+1)$ -th time slot as  $P_{t+1}^{a,z}$  where  $a$  is the action that was taken at the previous time-slot  $t$ , i.e.,  $a_t$ , and  $z$  is the observation made at  $(t+1)$ , i.e.,  $z_{t+1}$ . Furthermore, in this work we assume that POMDP  $\mathcal{P}$  satisfies the following two assumptions:

**Assumption 1.** Let  $x_t = [P_t, \bar{\Delta}_t]$  and  $x_t^+ = [P_t^+, \bar{\Delta}_t^+]$  be states such that  $H(P_t^{h,+}) \geq H(P_t^h)$  then  $H(P_{t+1}^{h,a,z,+}) \geq H(P_{t+1}^{h,a,z})$ ,  $a, z \in \{0, 1\}$ .

Assumption 1 states that, given the action at  $t$  and the observation at  $t+1$ , if the system starts in a belief state with higher health status entropy, i.e.,  $H(P_t^{h,+}) \geq H(P_t^h)$ , then it will make a transition to a state with higher health status entropy, i.e.,  $H(P_{t+1}^{h,a,z,+}) \geq H(P_{t+1}^{h,a,z})$ .

**Assumption 2.** Let  $I_S = \{0, 1, \dots, 7\}$  and  $i \in I_S$  be the index of the system's state  $s_t = [i_0, i_1, i_2]^T$  at time  $t = 0, 1, \dots, N$ , where  $i_0 = F^{MS}$ ,  $i_1 = F^S$ , and  $i_2 = F^{SM}$  (see Table I). Furthermore, let  $p_{i_1 0}^S, p_{i_2 0}^{SM}$  be the probabilities for the sensor  $S$  and the link  $l_{SM}$  to make a transition from health status  $i_1$  and  $i_2$ , respectively, to a healthy status (indicated by 0) at  $t+1$ . We assume that for the POMDP  $\mathcal{P}$  the following inequality is true,

$$\sum_{i \in I_S} p_t^i [p_{i_1 0}^S p_{i_2 0}^{SM} (2 - P_g) - 1] \leq 0, \quad t = 0, 1, \dots, N. \quad (5)$$

Assumption 2 expresses the necessary conditions and system's parametrization for the probing action to always result in a lower health status entropy compared to the no probe action. It may seem intuitive that probing reduces entropy, since it makes the generation of a status update from the sensor mandatory, i.e., it reduces the uncertainty induced in the system due to the probabilistic generation of status updates from the sensor, however, one should also consider that probing introduces a new type of uncertainty in the system due to the transmission failures occurring in the  $MS$  link. As an example consider the case where a probe was sent to the sensor yet no status update was received by the monitor. It is not certain whether this happened because the probe didn't actually reach the sensor, due to a faulty  $MS$  link, or because the sensor, or the  $SM$  link, or both were in a faulty state. Assumption 2 expresses the effect of faults occurring at the  $MS$  link along with that of parameters  $p_{i_1 0}^S, p_{i_2 0}^{SM}$  and  $P_g$  on the resulting health status entropy [33] and it is utilized in the proof of the following lemma in order to ensure that the probe action will always result in the same or reduced health status entropy compared to the no-probe action for a given observation  $z$  at time  $t+1$ .

**Lemma 1.** Let  $P_{t+1}^{0,z}$  and  $P_{t+1}^{1,z}$  be the belief states of  $\mathcal{P}$  at the  $(t+1)$ -th time slot when  $a_t = 0$  and 1, respectively, and let  $P_{t+1}^{h,0,z}, P_{t+1}^{h,1,z}$  be their corresponding health status belief vectors, then, if Assumption 2 is satisfied, it holds that,  $H(P_{t+1}^{h,0,z}) \geq H(P_{t+1}^{h,1,z})$ ,  $z \in \{0, 1\}$ .

The proof is given in [33]. Next, in Lemma 2, we show that the expected cost-to-go from decision stage  $t$  up to  $N$  is an increasing function of the health status entropy.

**Lemma 2.** Let  $x_t^+ = [P_t^+, \bar{\Delta}_t^+]$  and  $x_t^- = [P_t^-, \bar{\Delta}_t^-]$  be states such that  $H(P_t^{h,+}) \geq H(P_t^{h,-})$  and  $J_t(\cdot)$  be the dynamic program of  $\mathcal{P}$  then for  $t = 1, \dots, N$ , it holds that  $J_t(P_t^+, \bar{\Delta}_t^+) \geq J_t(P_t^-, \bar{\Delta}_t^-)$ .

The proof of Lemma 2 is given in [33]. In Lemma 3 we state a similar property for the expected cost-to-go when the value of AoI increases.

**Lemma 3.** Let  $\bar{\Delta}_t^+$  and  $\bar{\Delta}_t^-$  be normalized AoI values such that  $\bar{\Delta}_t^+ \geq \bar{\Delta}_t^-$  and  $J_t(\cdot)$  be the cost-to-go function in the dynamic program (4) then for  $t = 0, 1, \dots, N-1$ , it holds that  $J_t(P_t, \bar{\Delta}_t^+) \geq J_t(P_t, \bar{\Delta}_t^-)$ .

Lemma 4 states properties of the cost-to-go function  $J_t(\cdot)$  that are necessary to establish the structural properties of the optimal policy in Theorem 1.

**Lemma 4.** Let  $J_t(x_t)$  be the value of the dynamic program of  $\mathcal{P}$  at  $x_t = [P_t, \bar{\Delta}_t]$  then  $J_t(x_t)$  is piece-wise linear, increasing, and concave with respect to  $H(P_t^h)$  and  $\bar{\Delta}$  for  $t = 1, \dots, N$ .

The proof of Lemma 4 is given in [33]. Finally, in Theorem 1 we show that there exist configurations of POMDP  $\mathcal{P}$  such that the optimal policy is of a threshold type.

**Theorem 1.** At each decision stage  $t = 0, 1, \dots, N-1$  there exists a positive probing cost  $c$  such that the probing action is optimal for state  $x_t^T = [P_t, \bar{\Delta}_t]$  and for all states  $x_t^+ = [P_t^+, \bar{\Delta}_t^+]$  with  $H(P_t^{h,+}) \geq H(P_t^h)$  and  $\bar{\Delta}_t^+ \geq \bar{\Delta}_t$ .

The proof of Theorem 1 is given in [33].

## V. OPTIMAL POLICY APPROXIMATION

According to [33], given a proper probing cost  $c$ , the optimal policy  $\pi^*$  for the finite horizon POMDP  $\mathcal{P}$  is of a threshold type. Actually  $\pi^*$  is comprised of different threshold values at *each* decision stage  $t = 0, 1, \dots, N$ . More specifically, let  $\theta_t^{H,*}$  and  $\theta_t^{\Delta,*}$  be the optimal threshold values for the health status entropy and the normalized AoI at stage  $t$ , then the optimal policy can be expressed as  $\pi^* = \{[\theta_0^{H,*}, \theta_0^{\Delta,*}], [\theta_1^{H,*}, \theta_1^{\Delta,*}], \dots, [\theta_{N-1}^{H,*}, \theta_{N-1}^{\Delta,*}]\}$ . Computing  $\theta_k^* = [\theta_k^{H,*}, \theta_k^{\Delta,*}]^T$  for  $t = 0, 1, \dots, N$  can be a computationally demanding task, especially if one considers large time horizons. To address this problem we approximate the optimal policy  $\pi^*$  with a *single* threshold and utilize a Policy Gradient algorithm, namely, the Simultaneous Perturbation Stochastic Approximation (SPSA) Algorithm [35] in order to find it.

The SPSA algorithm appears in Algorithm 1 and operates by generating a sequence of threshold estimates,  $\theta_k = [\theta_k^H, \theta_k^\Delta]^T$ ,  $k = 1, 2, \dots, K$  that converges to a local minimum, i.e., an approximation of the best single threshold policy for POMDP  $\mathcal{P}$ . The SPSA algorithm picks a single random direction  $\omega_k$  along which the derivative is evaluated at each step  $k$ , i.e.,  $\omega_k^H$  and  $\omega_k^\Delta$  are independently generated according to a Bernoulli distribution as presented in line 4 of Algorithm 1. Subsequently, in line 5 the algorithm generates the threshold vectors  $\theta_k^+$  and  $\theta_k^-$ , which are bounded element-wise in the interval  $[0, 1]$ , i.e.,  $\mathbf{0}$  and  $\mathbf{1}$  in line 5 are column vectors whose elements are all zeros and ones respectively.  $\theta_k^\Delta$  is also bounded in  $[0, 1]$  since we assumed a normalized value for the AoI, and, this is also true for  $\theta_k^H$  since the maximum health status entropy occurs for  $P_t^h = [0.5, 0.5]$  which evaluates to 1. In line 6 the estimates  $\hat{J}(\theta^+)$  and  $\hat{J}(\theta^-)$  are computed by

simulating  $M_s$  times the POMDP  $\mathcal{P}$  under the corresponding single threshold policy. Finally, the gradient is estimated in line 7, where  $\odot$  represents an element-by-element division, and  $\theta_k$  is updated in line 8. Since the SPSA algorithm converges to local optima it is necessary to try several initial conditions  $\theta_0$ .

---

**Algorithm 1** Policy gradient algorithm for probing control

---

- 1: Initialize threshold  $\theta_0 = [\theta_0^H, \theta_0^\Delta]$  and  $\gamma, A, \eta, \beta, \zeta$
  - 2: **for**  $k = 1$  to  $K$  **do**
  - 3:  $\gamma_k = \frac{\gamma}{(k+A)^\beta}$  and  $\eta_k = \frac{\zeta}{k^\zeta}$
  - 4: Randomly set  $\omega_k^H, \omega_k^\Delta$  to the equiprobable values  $\{-1, 1\}$  and define  $\omega_k = [\omega_k^H, \omega_k^\Delta]^T$
  - 5:  $\theta_k^+ = \min\{\mathbf{1}, \max\{\mathbf{0}, \theta_{k-1} + \eta_k \cdot \omega_k\}\}$  and  $\theta_k^- = \min\{\mathbf{1}, \max\{\mathbf{0}, \theta_{k-1} - \eta_k \cdot \omega_k\}\}$
  - 6:  $y_k^+ = \hat{J}(\theta_k^+), y_k^- = \hat{J}(\theta_k^-)$
  - 7:  $e_k = (y_k^+ - y_k^-) \odot (2 \cdot c_k \cdot \omega_k)$
  - 8:  $\theta_k = \theta_{k-1} - \gamma_k \hat{e}$
  - 9: **end for**
- 

## VI. NUMERICAL RESULTS

In this section, we evaluate numerically the cost efficiency of the single threshold probing policy and provide comparative results with a delay based probing policy that is often used in practice. The delay based policy will probe the sensor whenever the time that has elapsed since the last arrival of a status update at the monitor exceeds a certain threshold. We also note here that for the system we consider in this work, the delay and AoI metrics coincide. This holds because the sensor does not buffer status updates, and the status update generation scheme is fixed. Thus, the results we present in this section exhibit that AoI fails to capture the semantics of information with the exception of timeliness.

For the first scenario we consider, the system was configured as follows,  $c = 1, \lambda_1 = 1, \lambda_2 = 1, P_g = 0.1$  and the transition probability kernels were set as,  $P^{MS} = \begin{bmatrix} 0.9 & 0.1 \\ 0.9 & 0.1 \end{bmatrix}, P^S = \begin{bmatrix} 0.9 & 0.1 \\ 0.9 & 0.1 \end{bmatrix}$ , and  $P^{SM} = \begin{bmatrix} 0.9 & 0.2 \\ 1 - p_{SM}^{11} & p_{SM}^{11} \end{bmatrix}$ , where  $p_{SM}^{11} = 0.1, 0.2, \dots, 0.9$ . Furthermore, we set the parameters of the SPSA algorithm as follows,  $\eta = 1, \gamma = 10^{-3}, A = 1, \beta = 1$ , and  $\zeta = 1$ . We derived the threshold policy by executing  $K = 20$  iterations of the SPSA algorithm. At each iteration  $k = 1, 2, \dots, K$  we calculated each of  $y_k^+$  and  $y_k^-$  as the sample average of 100 Monte-Carlo simulations. Each Monte-Carlo simulation had a time horizon of  $N = 100$  time slots and during that period the system was controlled by the single threshold policy defined by  $\theta_k^+$ , in the case of  $y_k^+$ , and  $\theta_k^-$ , in the case of  $y_k^-$ , as presented in Algorithm 1. Subsequently, we used the threshold  $\theta_K$  to evaluate the efficiency of the derived threshold policy. More specifically, for all policies appearing in Figure 2, we calculated the average cost  $\hat{J}_0$  as the sample average over  $M = 2000$  Monte-Carlo simulations of the system while it was being controlled by the corresponding policy over a period

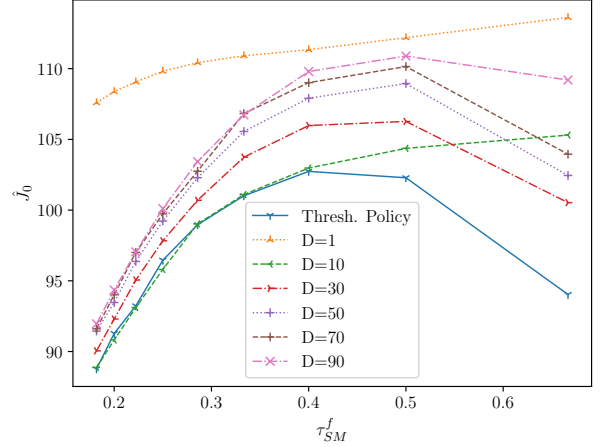


Fig. 2:  $\hat{J}_0(\cdot)$  vs  $\tau_{SM}^f$  for a horizon of 100 time slots.

of  $N$  time slots, i.e.,  $\hat{J}_0 = \frac{1}{M} \sum_{m=1}^M \sum_{t=0}^N g_t$ . Finally, for each Monte-Carlo simulation we set randomly the initial health status for the sensor and the  $l_{MS}$  and  $l_{SM}$  links.

In Figure 2 we present the evolution of  $\hat{J}_0$  with respect to the steady state probability of link  $l_{SM}$  to be in a faulty state,  $\tau_{SM}^f = \frac{p_{SM}^{01}}{1 - p_{SM}^{11} + p_{SM}^{01}}$ . We utilize  $\tau_{SM}^f$  instead of  $p_{MS}^{11}$  because it assumes a more intuitive interpretation, i.e., it expresses the expected time that link  $l_{SM}$  would spend in the faulty state over a large time horizon. In Figure 2 we consider multiple delay policies that will probe the sensor when a status update arrival has been delayed for more than  $D$  time slots. The results presented in Figure 2 indicate that the threshold based policy achieved a lower cost  $\hat{J}_0$  compared to the delay based policies. In order to provide insights into this result we have to point out the behavior of the two extreme delay policies, i.e., those with  $D$  equal to 1 and 90. The first one probed the sensor more often than the other policies since  $D$  assumed its smallest value, while the delay policy with  $D = 90$  practically never probed the sensor since 90 is almost equal to the entire time horizon of 100 time slots.

From Figure 2 we see that when  $\tau_{SM}^f$  was less than 0.20 the delay policy with  $D = 90$  and the threshold policy had similar cost efficiency. This means that probing was rarely needed for that range of  $\tau_{SM}^f$  values. When  $\tau_{SM}^f$  lied in the range between 0.20 and 0.30 the cost induced by the delay policy with  $D = 90$  increased with a higher rate compared to all other policies. This indicates that probing became necessary in order to reduce cost  $\hat{J}_0$ . This is evident also by the fact that the delay based policy with  $D = 10$  performed closer to the threshold policy within this range of  $\tau_{SM}^f$  values. When  $\tau_{SM}^f$  took values in the range between 0.30 and 0.5 all policies saw an increment in  $\hat{J}_0$ . For this range of  $\tau_{SM}^f$  values, the increased value of  $p_{SM}^{11}$ , resulted in extended periods during which the  $l_{SM}$  link was in a faulty state. As a consequence, the delay of status update arrivals increased and all delay based policies engaged in persistently probing the sensor whenever



delay exceed their threshold  $D$ . The persistence in probing is explained by the fact that the  $l_{SM}$  link was in a faulty state for long periods and the generated status updates could *not* reach the monitor and decrease delay below  $D$ . The effect of persistent probing is particularly evident in the abrupt increase of  $\hat{J}_0$  for the delay policies with the smaller values of  $D$ , i.e.,  $D = 1$  and  $10$ . In contrast, the threshold based policy was able to avoid unnecessary probing by utilizing the health status entropy and thus defer probing while it was confident that  $l_{SM}$  link was in a faulty state. Finally, when  $\tau_{SM}^f$  is larger than  $0.5$  we observe a reduction in the induced cost  $\hat{J}_0$  for all policies except for the delay based policies with  $D = 1$  and  $D = 10$ . The observed reduction in  $\hat{J}_0$  is mainly due to the uniform reduction in the cost induced by the health status entropy. More specifically, for large values  $\tau_{SM}^f$ , i.e., for large values of  $p_{SM}^{11}$  and  $p_{SM}^{01}$ , the monitor was confident that the  $l_{SM}$  link was in a faulty state, and this resulted in a reduced cost due to health status entropy. However, despite the reduction of  $\hat{J}_0$  values for all policies the effect of persistent probing is still evident and especially so for the delay policies with  $D = 1$  and  $10$ .

In order to provide further insight on the effect of health status entropy on  $\hat{J}_0$  we modified the previous scenario by setting,  $P^{MS} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$ ,  $P^S = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$ . By setting the

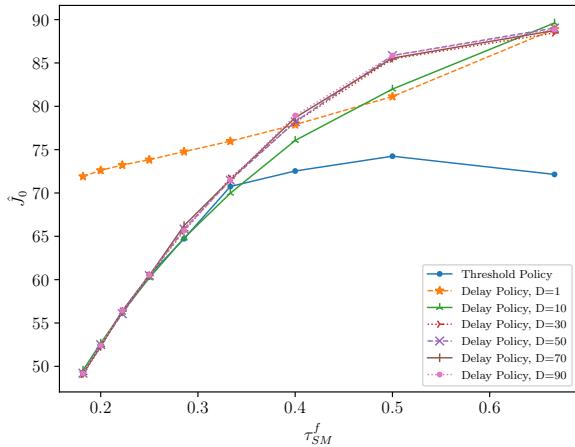


Fig. 3:  $J_0(\cdot)$  vs.  $\tau_{SM}^f$  when only the  $l_{SM}$  link is subject to failures.

matrices  $P^{MS}$  and  $P^S$  to the values presented above both the link  $l_{MS}$  and the sensor  $S$  would never enter a faulty state and, even if they were randomly initialized to a faulty state they would return to the healthy state with probability 1 in the next time slot. Thus, the system can now be in one of two states, i.e., the states with index  $i = 0$  and  $i = 1$  respectively in Table I. This comes in contrast to the eight possible states of the previous scenario and results in a uniformly reduced value for the health status entropy across all policies and the whole range of  $\tau_{SM}^f$  values. Now, in Figure 3 we do not observe the

significant reduction of  $\hat{J}_0$  when  $\tau_{SM}^f$  assumes values greater than  $0.5$  that we observed in Figure 2. This is because, the health status entropy cost is uniformly lower and the increment of  $\tau_{SM}^f$  has a much less significant effect on its value compared to the increment of normalized AoI and probing costs.

Finally, in Figure 4 we present the reduction of  $\hat{J}_0(\cdot)$  for all policies *relative* to the cost  $\hat{J}_0^{90}$  of the delay policy with  $D = 90$  for an increasing time horizon  $N$ . We modified the

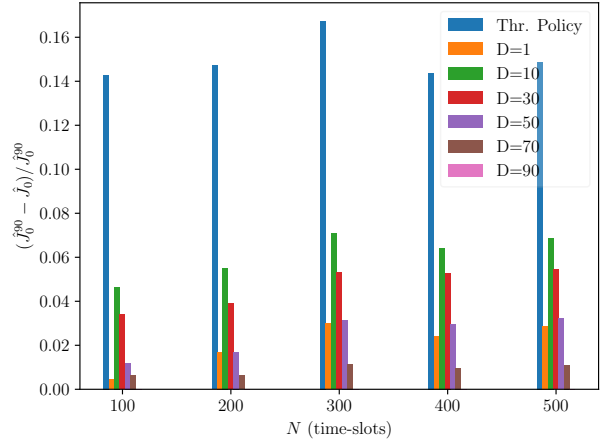


Fig. 4: Relative reduction in  $\hat{J}_0(\cdot)$  vs. time horizon  $N$ .

basic system setup of Figure 2 by setting  $p_{SM}^{11} = 0.9$  and  $\lambda_2 = \frac{N}{100}$  so that the normalized AoI doesn't become negligible as the time horizon increases. By setting  $\lambda_2 = \frac{N}{100}$  we had a normalized AoI cost of  $\bar{\Delta} = \frac{\Delta}{100}$ , which was analogous to that of the basic scenario for all values of  $N$ . Figure 4 depicts that the approximately optimal threshold policy achieves an almost constant reduction rate of 16% across all experiments indicating that an increasing horizon  $N$  is not detrimental to its performance.

## VII. CONCLUSIONS

In this work, we address the problem of deriving an efficient policy for probing sensors in IoT networks. We adopted a semantics-aware communications paradigm, formulated the problem as a POMDP and utilized a computationally efficient algorithm to derive an approximately optimal policy. Finally, we presented numerical results that exhibit significant cost reductions compared to conventional delay based policies.

## ACKNOWLEDGMENT

This research has been financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T1EDK-00070). The work of N. Pappas has been supported by the Center for Industrial Information Technology (CENIIT), the Swedish Research Council (VR), and the Excellence Center at Linköping-Lund in Information Technology (ELLIIT).

## REFERENCES

- [1] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, and M. Mukherjee, "A survey on fault diagnosis in wireless sensor networks," *IEEE Access*, 2018.
- [2] S. L. Campbell and R. Nikoukhan, *Auxiliary signal design for failure detection*. Princeton University Press, 2015.
- [3] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Communications Magazine*, vol. 59, no. 6, pp. 96–102, 2021.
- [4] P. Popovski, O. Simeone, F. Boccardi, D. Gündüz, and O. Sahin, "Semantic-effectiveness filtering and control for post-5G wireless connectivity," *Journal of the Indian Institute of Science*, vol. 100, no. 2, 2020.
- [5] B. Güler, A. Yener, and A. Swami, "The semantic communication game," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 787–802, 2018.
- [6] E. Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Computer Networks*, p. 107930, 2021.
- [7] N. Pappas and M. Kountouris, "Goal-oriented communication for real-time tracking in autonomous systems," in *IEEE International Conference on Autonomous Systems (ICAS)*, 2021.
- [8] Q. Lan, D. Wen, Z. Zhang, Q. Zeng, X. Chen, P. Popovski, and K. Huang, "What is semantic communication? a view on conveying meaning in the era of machine intelligence," *Journal of Communications and Information Networks*, 2021.
- [9] J. Dommel, D. Wieruch, Z. Utkovski, and S. Stańczak, "A semantics-aware communication scheme to estimate the empirical measure of a quantity of interest via multiple access fading channels," in *IEEE Statistical Signal Processing Workshop 2021*.
- [10] X. Luo, H.-H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wireless Communications*, pp. 1–10, 2022.
- [11] M. Kalfa, M. Gok, A. Atalik, B. Tegin, T. M. Duman, and O. Arikan, "Towards goal-oriented semantic signal processing: Applications and future challenges," *Digit. Signal Process.*, vol. 119, 2021.
- [12] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, T. Soleymani, B. Soret, and K. H. Johansson, "Semantic communications in networked systems: A data significance perspective," 2022.
- [13] G. Stamatakis, N. Pappas, A. Fragkiadakis, and A. Traganitis, "Autonomous maintenance in IoT networks via aoi-driven deep reinforcement learning," in *IEEE INFOCOM Workshops*, 2021.
- [14] M. M. H. Khan, H. K. Le, H. Ahmadi, T. F. Abdelzaher, and J. Han, "Dustminer: troubleshooting interactive complexity bugs in sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008.
- [15] J. Yang, M. L. Soffa, L. Selavo, and K. Whitehouse, "Clairvoyant: a comprehensive source-level debugger for wireless sensor networks," in *Proceedings of the 5th international conference on Embedded networked sensor systems*, 2007.
- [16] G. Stamatakis, N. Pappas, and A. Traganitis, "Control of status updates for energy harvesting devices that monitor processes with alarms," in *IEEE Globecom Workshops*, 2019.
- [17] D. Jeswani, M. Natu, and R. K. Ghosh, "Adaptive monitoring: application of probing to adapt passive monitoring," *Journal of Network and Systems Management*, vol. 23, no. 4, 2015.
- [18] A. Tayal, N. Hubballi, M. Natu, and V. Sadaphal, "Congestion-aware probe selection for fault detection in networks," in *10th International Conference on Communication Systems & Networks (COMSNETS)*, 2018.
- [19] A. Tayal, N. Sharma, N. Hubballi, and M. Natu, "Traffic dynamics-aware probe selection for fault detection in networks," *Journal of Network and Systems Management*, vol. 28, no. 4, 2020.
- [20] Z. Hu, L. Zhu, C. Ardi, E. Katz-Bassett, H. V. Madhyastha, J. Heidemann, and M. Yu, "The need for end-to-end evaluation of cloud availability," in *International Conference on Passive and Active Network Measurement*. Springer, 2014.
- [21] L. Quan, J. Heidemann, and Y. Pradkin, "Detecting internet outages with precise active probing (extended)," *USC/Information Sciences Institute, Tech. Rep.*, 2012.
- [22] D. Jeswani, M. Natu, and R. K. Ghosh, "Adaptive monitoring: A framework to adapt passive monitoring using probing," in *8th International Conference on network and service management (CNSM) and Workshop on systems virtualization management (SVM)*, 2012.
- [23] A. Kosta, N. Pappas, and V. Angelakis, "Age of information: A new concept, metric, and tool," *Foundations and Trends in Networking*, vol. 12, no. 3, pp. 162–259, 2017.
- [24] Y. Sun, I. Kadota, R. Talak, and E. Modiano, "Age of information: A new metric for information freshness," *Synthesis Lectures on Communication Networks*, vol. 12, no. 2, pp. 1–224, 2019.
- [25] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.
- [26] M. Moltafet, M. Leinonen, M. Codreanu, and N. Pappas, "Power minimization for age of information constrained dynamic control in wireless sensor networks," *IEEE Transactions on Communications*, vol. 70, no. 1, pp. 419–432, 2022.
- [27] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "The cost of delay in status updates and their value: Non-linear ageing," *IEEE Transactions on Communications*, 2020.
- [28] X. Wang, W. Lin, C. Xu, X. Sun, and X. Chen, "Age of changed information: Content-aware status updating in the internet of things," *IEEE Transactions on Communications*, 2022.
- [29] B. Zhou and W. Saad, "Joint status sampling and updating for minimizing age of information in the internet of things," *IEEE Transactions on Communications*, 2019.
- [30] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas, "A reinforcement learning framework for optimizing age of information in rf-powered communication systems," *IEEE Transactions on Communications*, 2020.
- [31] P. D. Mankar, Z. Chen, M. A. Abd-Elmagid, N. Pappas, and H. S. Dhillon, "Throughput and age of information in a cellular-based iot network," *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 8248–8263, 2021.
- [32] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *IEEE INFOCOM*, March 2012.
- [33] G. Stamatakis, N. Pappas, A. Fragkiadakis, and A. Traganitis, "Semantics-aware active fault detection in status updating systems," *arXiv preprint arXiv:2202.00923*, 2022.
- [34] V. Krishnamurthy, *Partially observed Markov decision processes*. Cambridge University Press, 2016.
- [35] J. C. Spall, *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons, 2005, vol. 65.