

Integrated Push-and-Pull Update Model for Goal-Oriented Effective Communication

Pouya Agheli, *Graduate Student Member, IEEE*, Nikolaos Pappas, *Senior Member, IEEE*,
Petar Popovski, *Fellow, IEEE*, and Marios Kountouris, *Fellow, IEEE*.

Abstract—This paper studies decision-making for goal-oriented effective communication. We consider an end-to-end status update system where a sensing agent (SA) observes a source, generates and transmits updates to an actuation agent (AA), while the AA takes actions to accomplish a goal at the endpoint. We integrate the push- and pull-based update communication models to obtain a *push-and-pull* model, which allows the transmission controller at the SA to decide whether to push an update to the AA and the query controller at the AA to pull updates by initiating queries at specific time instants. To gauge effectiveness, we utilize a *grade of effectiveness* (GoE) metric incorporating the updates' freshness, usefulness, and the timeliness of actions as qualitative attributes. We then derive effect-aware policies to maximize the expected discounted sum of the updates' effectiveness subject to induced costs. The effect-aware policy at the SA considers the potential effectiveness of communicated updates at the endpoint, while at the AA, it accounts for the probabilistic evolution of the source and importance of the generated updates. Our results show the proposed push-and-pull model outperforms models solely based on push- or pull-based updates both in terms of efficiency and effectiveness. Additionally, using effect-aware policies at both agents enhances effectiveness compared to periodic and/or probabilistic, effect-agnostic policies at either or both agents.

Index Terms—Goal-oriented effective communication, status update systems, push-and-pull model, decision-making.

I. INTRODUCTION

The emergence of cyber-physical systems empowered by interactive and networked sensing and actuation/monitoring agents has caused a shift in focus from extreme to sustainable performance. Emerging networks aim to enhance effectiveness within the system while substantially improving resource utilization, energy consumption, and computational efficiency. The key is to strive for a minimalist design, frugal in resources, which can scale effectively rather than causing network over-provisioning. This design philosophy has crystallized into

the goal-oriented and/or semantic communication paradigm, holding the potential to enhance the efficiency of diverse network processes through a parsimonious use of communication and computation resources [2], [3]. From an effectiveness perspective, a message is generated and conveyed by a sender *if* it has the potential to have the *desirable effect* or the *right impact* at the destination, e.g., executing a critical action, for accomplishing a specific goal. This promotes system scalability and efficient resource usage by avoiding the acquisition, processing, and transportation of information that turns out to be ineffective, irrelevant, or useless.

Messages, e.g., in the form of status update packets, are communicated over existing networked intelligent systems mostly using a *push-based* communication model. Therein, packets arriving at the source are sent to the destination based on decisions made by the source, regardless of whether the endpoint has requested or plans to utilize these updates to accomplish a goal. In contrast, in a *pull-based* model, the endpoint decides to trigger and request packet transmissions from the source and controls the time and the type of generated updates [4]–[10]. Nevertheless, this model does not consider the availability of the source to generate updates or the usefulness of those updates. To overcome these limitations, we propose an *integrated push-and-pull* model that involves both agents/sides in the decision-making process, thereby combining push- and pull-based paradigms in a way that mitigates their drawbacks. In either model, decisions at the source or endpoint could influence the effectiveness of communicated updates. Therefore, we can categorize decision policies into *effect-aware* and *effect-agnostic*. Under an effect-aware policy, the source adapts its decisions by taking into account the effects of its communicated packets at the endpoint. Likewise, the endpoint initiates queries based on the evolution of the source and the expected importance of the pulled updates. Under the effect-agnostic policy, however, decisions are made regardless of their consequent effect on the performance.

We illustrate the motivation behind the integrated push-and-pull model under effect-aware policies with a real-world example: *Consider an autonomous mobile robot (AMR) platform, such as Amazon's Proteus. In addition to communicating with a central control system, robotic agents might engage in peer-to-peer interactions to support dynamic obstacle detection, collision avoidance, and cooperative task execution. In this context, timely and effective information exchange becomes critical, particularly in dense environments, such as warehouses. Each robot may serve as a sensing agent, an actuation agent, or both, depending on its current state, assigned*

P. Agheli and M. Kountouris are with the Communication Systems Dept., EURECOM, France, email: agheli@eurecom.fr. M. Kountouris is also with the Dept. of Computer Science and Artificial Intelligence, University of Granada, Spain, email: mariosk@ugr.es. N. Pappas is with the Dept. of Computer and Information Science, Linköping University, Sweden, email: nikolaos.pappas@liu.se. P. Popovski is with the Dept. of Electronic Systems, Aalborg University, Denmark, email: petarp@es.aau.dk. This work has been supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101003431), the Velux Foundation, Denmark, through the Villum Investigator Grant WATER, nr. 37793, the Swedish Research Council (VR), ELLIIT, and the SNS JU projects under the EU's Horizon program Grant Agreement No 101139232(6G-GOALS), No. 101096526 (ETHER), and No. 101192080 (6G-LEADER), and the European Union's Horizon Europe Research and Innovation Programme under the Marie Skłodowska-Curie Grant Projects SOVEREIGN under Agreement 101131481 and ELIXIRION under Grant 101120135. Part of this work is presented in [1].

tasks, and proximity to other robots. A purely push-based communication model risks overwhelming actuation agents with irrelevant or redundant data, or miss critical updates amid competing transmissions. Conversely, a purely pull-based model requires actuation agents to query sensing agents for updates, which can delay the detection or response to urgent events, such as unexpected obstacles. The push-and-pull model offers a balanced alternative by integrating both strategies. Actuation agents retain the flexibility to pull information as needed, while sensing agents can proactively push updates deemed highly relevant to current goals. With effect-aware decision-making, information exchange and robotic actions are prioritized based on contextual relevance and the actuator's readiness to respond. This fosters efficient multiplexing of shared computational resources, and thus energy, across concurrent tasks of a robot, including sensing, actuation, and communication. Furthermore, by continuously monitoring environmental dynamics, actuation agents can adapt the timing of their queries to align with the evolving urgency and impact of the incoming information.

In this work, we investigate a time-slotted end-to-end status update system in which a sensing agent observes an information source and communicates updates/observations in the form of packets with an actuation agent. The actuation agent then takes action based on the updates successfully received as a means of accomplishing a subscribed goal at the endpoint. We develop an integrated push-and-pull model, which allows both agents to make decisions based on their local policies or objectives. In particular, a transmission controller at the sensing agent decides to either send or drop update packets according to their potential usefulness at the endpoint. On the other side, a query controller at the actuation agent also determines the time instants around which the actuator should perform actions in the form of initiating queries. In that sense, effective updates are those that result in the right impact and action at the endpoint. These queries, however, are *not* communicated to the sensing agent. Instead, the actuation agent acknowledges effective updates to the sensing agent. With prior knowledge that the effectiveness of updates depends on the actuator's availability to perform actions, the sensing agent can infer the initiated queries based on those acknowledgments. A time diagram showing processes at both agents is depicted in Fig. 1.

We introduce a metric to measure the effectiveness and significance of updates and derive a class of optimal policies for each agent that make effect-aware decisions to maximize the long-term expected effectiveness of the update packets communicated to achieve the goal, subject to induced costs. These costs may arise from transmitting updates, initiating queries, and the availability of the actuation agent to respond and take action, all of which depend on the communication model. To do so, each agent must first estimate the necessary system parameters to make the right decisions. Our analytical and simulation results show that the integrated push-and-pull model offers higher energy efficiency compared to the push-based model and better effectiveness performance compared to the pull-based one. Moreover, we observe that utilizing effect-aware policies at both agents significantly improves the

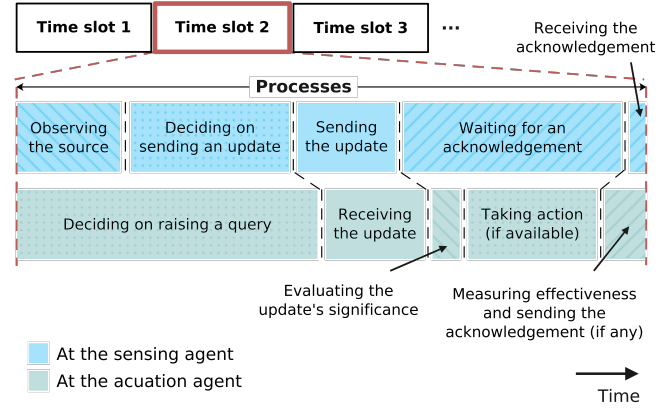


Fig. 1. A timing diagram of processes involving the sensing and actuation agents, illustrating interactions and update communications leading to actions.

system's effectiveness performance in most cases, with a large gap compared to periodic and probabilistic effect-agnostic policies at either or both agents. Accordingly, we demonstrate that the solution for finding an optimal effect-aware policy at each agent converges to a *threshold-based* agent decision framework where the agent can make timely decisions based on an individual lookup map in hand and threshold boundaries computed to satisfy the goal.

A. Related Works

This paper widely broadens prior work on push-based and (query-) pull-based communications by enabling both agents to make decisions in order to maximize the effectiveness of communicated updates in the system. The pull-based communication model has been widely analyzed; see, for example, [4]–[10]. In [4], a metric called effective age of information (EAoI), which comprises the effects of queries and the freshness of updates in the form of the age of information (AoI) [11]–[13], is introduced. Query AoI (QAoI), which is similar to the EAoI, is utilized in [5]–[9]. Following the same concept as QAoI, on-demand AoI was introduced in [10], [14]. Probe-based (query-based) active fault detection, where actuation or monitoring agents adaptively decide to probe sensing agents to detect probable faults at the endpoint, is studied in [15]. Most prior work has employed a pull-based communication model and focuses on the freshness and timeliness of information. In this work, we consider multiple information attributes and propose a *grade of effectiveness* metric to measure the effectiveness of updates, which goes beyond existing metrics such as AoI, EAoI, QAoI, on-demand AoI, age of incorrect information (AoII) [16], and value of information (VoI) [17], [18]. In particular, we focus on the freshness of successfully received updates and the timeliness of performed actions as two attributes of interest at the link level and on the usefulness of the updates to achieve the goal at the source level.

This paper extends our prior work [1], which only considers a pull-based model and an effectiveness metric with two attributes: freshness and usefulness. As such, [1] conveys a special form of the decision problem we solve here. In this work, we generalize the problem to a push-and-pull communication model, considering that the sensing and actuation agents

individually make decisions and converge to a point where they can transmit updates and initiate queries, respectively, to maximize the effectiveness of updates and achieve the desired impact at the endpoint. Importantly, we assume that the source distribution is unknown to the actuation agent, and the sensing agent does not have perfect knowledge of the goal. Therefore, the agents must estimate the required parameters separately. This approach is substantially different from the one in our previous work and other state-of-the-art techniques.

B. Contributions

The main contributions can be briefly outlined as follows.

- We develop an *integrated push-and-pull update communication model*, which allows both agents to take decision-making roles in communicating updates and taking appropriate actions to satisfy the goal, following the paradigm of goal-oriented communications. This makes the system more adaptable from an effectiveness viewpoint compared to the conventional push- and pull-based models.
- We use a *grade of effectiveness* metric to capture the timely impact of communicated updates at the endpoint, which relies on the freshness of successfully communicated updates, the timeliness of actions performed, and the usefulness of those updates in fulfilling the goal. Our approach maps multiple information attributes into a unique metric that measures the impact or effect that each status update packet traveling over the network can have.
- We obtain optimal model-based control policies for agents that make effect-aware decisions to maximize the discounted sum of updates' effectiveness while keeping the induced costs within predefined constraints. To achieve this, we formulate an optimization problem, derive its dual form, and propose an iterative algorithm based on dynamic programming to solve the decision problem separately and independently from each agent's perspective.
- We demonstrate that the integrated push-and-pull model offers higher energy efficiency than the push-based model and better effectiveness performance compared to the pull-based one. We also show that applying effect-aware policies at both agents results in better performance than in the scenarios where one or both agents utilize effect-agnostic policies. We also broaden our results by deriving model-free decisions using *reinforcement learning*. Finally, we provide a lookup map presenting the optimal decisions for each agent that applies the effect-aware policy based on the given solution. This allows the agent to make decisions on time by merely looking up the map with the obtained threshold-based policy for the goal.

Notations: \mathbb{R} , \mathbb{R}_0^+ , and \mathbb{N} indicate the sets of real, non-negative real, and natural numbers, respectively. $\mathbb{E}[\cdot]$ denotes the expectation operator, $|\cdot|$ depicts the absolute value operator, $\mathbb{1}\{\cdot\}$ is the indicator function, $\mathcal{O}(\cdot)$ denotes growth rate of a function, and $\lceil \cdot \rceil$ is the ceiling function.

II. SYSTEM MODEL

We consider an end-to-end communication system in which a *sensing agent* (SA) sends messages in a time-slotted manner to an *actuation agent* (AA) as a means of taking effective action at the endpoint and satisfy a *subscribed goal*. Specifically, the SA observes a source and generates *status update packets* in each time slot, and a transmission controller decides whether to transmit that observation or not, following a specific policy. We assume that the source has finite-dimensional realizations and that the observation at the n -th, $\forall n \in \mathbb{N}$, time slot is assigned a rank of importance v_n from a finite set $\mathcal{V} = \{v_i | i \in \mathcal{I}\}$, with $\mathcal{I} = \{1, 2, \dots, |\mathcal{V}|\}$, based on its significance or usefulness for satisfying the goal, measured or judged at the source level.¹ The elements of \mathcal{V} are independent and identically distributed (i.i.d.) with probability $p_i = p_{\nu}(v_i)$ for the i -th outcome, where $p_{\nu}(\cdot)$ denotes a given probability mass function (pmf).²

The AA is assisted by a query controller that decides to initiate queries and to pull new updates according to a certain policy. A packet received at the AA has a satisfactory or sufficient impact at the endpoint if that update achieves a minimum effectiveness level, subject to the latest query initiated and the AA's availability to act on it. An effective update communication is followed by an *acknowledgment of effectiveness* (E-ACK) signal sent from the AA to the SA to report the effective update communication. We assume all transmissions and E-ACK feedback occur over packet erasure channels (PECs), with p_e and p'_e being the erasure probabilities in the forward communication and the acknowledgment links, respectively. Therefore, an E-ACK may not be received at the SA due to either *ineffective* update communication or erasure in the acknowledgment (backward) channel. With this interpretation, channel errors lead to a graceful degradation of the proposed scheme. An initiated query does *not* necessarily need to be shared with the SA. As discussed in Section III-C, the SA can deduce the initiation of a query or the availability of the AA to take action from a successful E-ACK, given prior knowledge that an update can be effective *only if* it arrives within the period during which the AA is available to act.

In this model, we consider the goal to be subscribed at the endpoint, with the AA fully aware of it. In contrast, the SA does not initially know the goal but learns which updates could be useful to accomplish the goal based on the received E-ACK and the significance of observations. Meanwhile, the AA is not aware of the evolution of the source or the likely importance of observations, attempting to approximate it based on arrival updates. Consequently, the agents might use different criteria to measure the usefulness of the updates and may need to adjust their criteria or valuation frameworks to account for possible goal changes over time. Finally, we assume that update acquisition, potential communication, and waiting time for receiving an E-ACK occur within *one* slot.

¹To determine the usefulness of an update, we can use the same *metavalue* approach proposed in [19, Section III-A].

²A more elaborate model could consider the importance of a realization dependent on the most recently generated update at the SA. This implies that a less important update increases the likelihood of a more significant update occurring later, which can be captured by utilizing a learning algorithm.

A. Communication Model

The following three strategies can be employed for effective communication of status updates.

1) *Push-based*: Under this model, the SA pushes its updates to the AA, taken for instance based on the source evolution, without considering whether the AA has requested them or is available to take any action upon receipt. This bypasses the query controller, enabling the SA to directly influence actions at the AA side.

2) *Pull-based*: In this model, the query controller plays a central role in the generation of update arrivals at the AA by pulling those updates from the SA. Here, the AA can *only* take action when queries are initiated. However, this model excludes the SA from generating and sending updates.

3) *Push-and-pull*: This model arises from integrating the push- and pull-based models so that the transmission and query controllers individually decide to transmit updates and send queries, respectively. Thereby, the AA is provided with a level of flexibility where it is also able to take some actions *beyond* query instances within a *limited* time. As a result, the effectiveness of an update packet depends on both agents' decisions. Dismissing the decision of either agent transforms the push-and-pull model into the push- or pull-based model.

B. Agent Decision Policies

We propose that the agents can adhere to the following decision policies, namely *effect-agnostic* and *effect-aware*, for transmitting updates or initiating queries to satisfy the goal.

1) *Effect-agnostic*: This policy employs a predetermined schedule or random process (e.g., Poisson, binomial, or Markov [5]–[9]) to send updates (initiate queries) from (by) the SA (AA), without considering their impact at the endpoint. We define a *controlled update transmission (query) rate* specifying the expected constant number of updates (queries) to be communicated (initiated) within a period. Also, as the effect-agnostic policy does not account for what might be happening in the other agent during the decision time, an *aleatoric* uncertainty is associated with random updates (queries).

2) *Effect-aware*: The effect-aware policy takes into consideration the impacts of both agents' decisions at the endpoint. In this regard, the SA (AA) predicts the effectiveness status at the endpoint offered by a sent update that is potentially received at the AA (the usefulness of a possible update at the source). Then, based on this prediction, the agent attempts to adapt transmission (query) instants and send (pull) updates in the right slots. This policy comes with an *epistemic* uncertainty because decisions are made according to probabilistic estimations, not accurate knowledge. However, such uncertainty can be decreased using learning or prediction techniques.

III. EFFECTIVENESS ANALYSIS METRICS

To achieve the right effect at the endpoint, an update packet that is successfully received at the AA has to satisfy a set of qualitative attributes, captured by the metrics as follows.

A. Grade of Effectiveness Metric

We introduce a *grade of effectiveness* (GoE) metric that comprises several qualitative attributes and characterizes the amount of impact an update makes at the endpoint. Mathematically speaking, the GoE metric is modeled via a composite function $\text{GoE}_n = (f \circ g)(\mathcal{I}_n)$ for the n -th time slot. Here, $g : \mathbb{R}^x \rightarrow \mathbb{R}^y, x \geq y$, is a (nonlinear) function of $x \in \mathbb{N}$ information attributes $\mathcal{I}_n \in \mathbb{R}^x$, and $f : \mathbb{R}^y \rightarrow \mathbb{R}$ is a context-aware function.³ The particular forms of functions f and g could vary according to different subscribed goals and their relevant requirements.

In this paper, without loss of generality, we consider *freshness* of updates and *timeliness* of actions as the main contextual attributes. The first comes in the form of the AoI metric, which is denoted by $\Delta_n \in \mathbb{N}$. The second is measured from the action's *lateness*, denoted by $\Theta_n \in \mathbb{N}$. Thereby, we can formulate the GoE metric as follows

$$\text{GoE}_n = f_g(g_\Delta(\hat{v}_n, \Delta_n), g_\Theta(\Theta_n); g_c(C_n)) \quad (1)$$

where $C_n \in \mathbb{R}_0^+$ represents the overall cost incurred at the n -th time slot. Also, $g_\Delta : \hat{\mathcal{V}} \times \mathbb{N} \rightarrow \mathbb{R}_0^+$, $g_\Theta : \mathbb{N} \rightarrow \mathbb{R}_0^+$, and $g_c : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ are *penalty* functions, and $f_g : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ is a non-decreasing *utility* function. Moreover, g_Δ , g_Θ , and g_c are non-increasing with respect to (w.r.t.) Δ_n , Θ_n , and C_n , respectively, while g_Δ is non-decreasing w.r.t. \hat{v}_n . Here, \hat{v}_n is the usefulness of the received update from the endpoint's viewpoint at the n -th slot. Thus, we assume that \hat{v}_n belongs to the set $\hat{\mathcal{V}} = \{0\} \cup \{\hat{v}_j \mid \hat{v}_j > 0, j \in \mathcal{J}\}$ with i.i.d. elements, where $\mathcal{J} = \{1, 2, \dots, |\hat{\mathcal{V}}| - 1\}$, the j -th element has probability $q_j = p_{\hat{v}}(\hat{v}_j)$, and $p_{\hat{v}}(\cdot)$ is a pmf derived in Section V-A. Since the packet is sent over a PEC, $\hat{v}_n = 0$ if it is erased or the update ends up being useless at the endpoint.

1) *AoI*: Measuring the freshness of correctly received updates at the AA within a query slot, the AoI is defined as $\Delta_n = n - u(n)$, where $\Delta_0 = 1$ and $u(n)$ is the slot index of the latest successful update, which is given by

$$u(n) = \max \{m \mid m \leq n, \beta_m(1 - \epsilon_m) = 1\} \quad (2)$$

with $\epsilon_m \in \{0, 1\}$ being the channel erasure at the m -th slot. In addition, $\beta_m \in \{0, 1\}$ indicates the query controller's decision, where $\beta_m = 1$ means pulling the update; otherwise, $\beta_m = 0$.

2) *Action lateness*: The lateness of an action performed at the n -th time slot in relevance to a query initiated at the n' -th slot, $n' \leq n$, is calculated as follows

$$\Theta_n = (1 - \beta_n)(n - n'), \quad (3)$$

which is valid for $\Theta_n < \Theta_{\max}$. Herein, Θ_{\max} shows the *width of action window* within which the AA can act on each query based on update arrivals from the SA. Outside the dedicated action window for the SA, the AA might undertake other tasks or communicate with other agents. Employing the push-based, pull-based, and push-and-pull update communication models, we have $\Theta_{\max} = \infty$, $\Theta_{\max} = 1$, and $\Theta_{\max} > 1$, respectively. Fig. 2 shows the action and idle windows for different models.

³The GoE metric in this form can be seen as a special case of the semantics of information (SoI) metric introduced in [2], [19], [20].

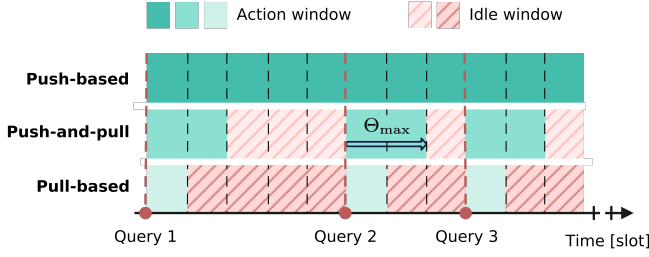


Fig. 2. The outline of the action and idle windows in different models.

A wider action window enables higher flexibility during heavy action loads at the cost of longer actuation availability.

B. Special Forms of the GoE

The GoE metric's formulation in (1) can simply turn into the QAOI and the VoI metrics as special cases. In this regard, we obtain a penalty function of the QAOI such that $\text{GoE}_n = g_\Delta(\Delta_n)$ if we set $\Theta_{\max} = 1$, assume linear $g_\Theta(\cdot)$, and overlook updates' usefulness and cost. In addition, by removing the concepts of query and time, hence the freshness and timeliness in the GoE's definition, we arrive at a utility function of the VoI, i.e., $\text{GoE}_n = f_g(\hat{v}_n; g_c(C_n))$.

C. Effectiveness Indicator

An update at the n -th time slot is considered effective at the system level if its GoE_n is higher than a *target effectiveness grade*, which is called GoE_{tgt} and is necessary to satisfy the goal. Let us define E_n as an *effectiveness indicator* at the n -th time slot. Thus, we can write

$$E_n = \mathbb{1}\{\text{GoE}_n \geq \text{GoE}_{\text{tgt}} \wedge \Theta_n < \Theta_{\max}\}. \quad (4)$$

The second condition in (4) appears from (3). According to (4), an update could be effective *only if* it arrives within the action window of the AA. Hence, a consequent E-ACK shared with the SA can imply the initiation of a query or the availability of the AA to take action. Given the values of Δ_n and Θ_n , and by inserting (1) into (4), we reach a *target usefulness* level v_{tgt} as the importance threshold that the update should exceed to be considered effective. In this case, if $\Theta_n < \Theta_{\max}$, we have

$$v_{\text{tgt}} = \min \left\{ \hat{v}_j \mid \hat{v}_j \in \hat{\mathcal{V}}, f_g(g_\Delta(\hat{v}_j, \Delta_n), g_\Theta(\Theta_n)) \geq \text{GoE}_{\text{tgt}} \right\}; \quad (5)$$

otherwise, $v_{\text{tgt}} = \max\{\hat{v}_j \mid \hat{v}_j \in \hat{\mathcal{V}}\}$. In (5), v_{tgt} can be computed using exhaustive search.

IV. MODEL-BASED AGENT DECISIONS

In this part, we first formulate a decision problem for effect-aware policies, cast it as a *constrained Markov decision process* (CMDP) [21], and then solve it based on its dual form.

A. Problem Formulation

The objective is to maximize the expected *discounted* sum of the updates' effectiveness in fulfilling the subscribed goal,

where each agent individually derives its decision policy subject to the relevant imposed cost by looking into the problem from its own perspective. Let us define π_α^* and π_β^* as the classes of optimal policies for transmission and query controls, respectively. Therefore, we can formulate the decision problem solved at each agent as follows⁴

$$\begin{aligned} \mathcal{P}_1 : \quad & \max_{\pi_\gamma} \limsup_{N \rightarrow \infty} \frac{1}{N} \mathbb{E} \left[\sum_{n=1}^N \lambda^n E_n \mid E_0 \right] \\ \text{s.t.} \quad & \limsup_{N \rightarrow \infty} \frac{1}{N} \mathbb{E} \left[\sum_{n=1}^N \lambda^n c_\gamma(\gamma_n) \right] \leq C_{\gamma, \max} \end{aligned} \quad (6)$$

where $\lambda \in [0, 1]$ indicates a discount factor, and $\gamma \in \{\alpha, \beta\}$ is replaced with α and β for the update transmission and query decision problems, respectively, at the SA and the AA. Herein, $\gamma_n \in \{0, 1\}$ denotes the decision at the relevant agent, $c_\gamma : \{0, 1\} \rightarrow \mathbb{R}_0^+$ is a non-decreasing cost function, and $C_{\gamma, \max}$ shows the maximum discounted cost.

For either update communication model introduced in Section II-A, optimal decisions at the agent(s) following the effect-aware policy, i.e., π_α^* and/or π_β^* , are obtained by solving \mathcal{P}_1 in (6). However, for every agent that employs an effect-agnostic policy, with regard to Section II-B, there is a pre-defined/given set of decisions denoted by $\tilde{\pi}_\alpha$ or $\tilde{\pi}_\beta$ such that $\pi_\alpha = \tilde{\pi}_\alpha$ or $\pi_\beta = \tilde{\pi}_\beta$, respectively.

B. CMDP Modeling

We cast \mathcal{P}_1 from (6) into an *infinite-horizon* CMDP denoted by a tuple $(\mathcal{S}_\gamma, \mathcal{A}_\gamma, P_\gamma, r_\gamma)$ with components that are defined via the agent that solves the decision problem.

1) *Modeling at the SA*: The CMDP at the SA is modeled according to the following components:

States – The state of the system $S_{\alpha, n}$ at the n -th slot from the SA's perspective is depicted by a tuple (v_n, \hat{E}_n) in which v_n is the update's usefulness, and $\hat{E}_n \in \{0, 1\}$ shows the E-ACK arrival status at the SA after passing the PEC, as defined in Section II. Herein, we have $\hat{E}_n = 0$ in case $E_n = 0$ or the acknowledgment signal is erased; otherwise, $\hat{E}_n = 1$. In this regard, $S_{\alpha, n}$ belongs to a finite and countable state space \mathcal{S}_α with $|\mathcal{S}_\alpha| = 2 \cdot |\mathcal{V}|$ elements.

Actions – We denote α_n the decision for update communication at the n -th slot, which is a member of an action space $\mathcal{A}_\alpha = \{0, 1\}$. In this space, 0 stands for discarding the update, and 1 indicates transmitting the update.

Transition probabilities – The transition probability from the current state $S_{\alpha, n}$ to the future state $S_{\alpha, n+1}$ via taking the action α_n is written by

$$\begin{aligned} p_\alpha(S_{\alpha, n}, \alpha_n, S_{\alpha, n+1}) &= \Pr((v_{n+1}, \hat{E}_{n+1}) \mid (v_n, \hat{E}_n), \alpha_n) \\ &= p_\nu(v_{n+1}) \Pr(\hat{E}_{n+1} \mid v_n, \alpha_n) \end{aligned} \quad (7)$$

since \hat{E}_{n+1} and \hat{E}_n are independent, and \hat{E}_n is independent of $v_n, \forall n$. We can derive the conditional probability in (7) as

⁴In an ideal scenario, where both agents have full knowledge of the goals and the source's evolution, the problem could be approached in a centralized manner, allowing for the joint derivation of policies for both agents. However, this falls outside the scope of this work.

- $\Pr(\hat{E}_{n+1} = 0 | v_n, \alpha_n) = \Pr(\hat{v}_{\text{tgt}} > \alpha_n v_n) = 1 - P_{\hat{v}_{\text{tgt}}}(\alpha_n v_n),$
- $\Pr(\hat{E}_{n+1} = 1 | v_n, \alpha_n) = P_{\hat{v}_{\text{tgt}}}(\alpha_n v_n),$

where \hat{v}_{tgt} is a *mapped* target usefulness that the SA considers, $P_{\hat{v}_{\text{tgt}}}(\hat{v}_{\text{tgt}}) = \sum_{\hat{v}'_{\text{tgt}} \leq \hat{v}_{\text{tgt}}} p_{\hat{v}_{\text{tgt}}}(\hat{v}'_{\text{tgt}})$ indicates its cumulative distribution function (CDF), and $p_{\hat{v}_{\text{tgt}}}(\cdot)$ shows the pmf derived in Section V-B.

Rewards – The immediate reward of moving from the state $S_{\alpha,n}$ to the state $S_{\alpha,n+1}$ under the action α_n is equal to $r_\alpha(S_{\alpha,n}, \alpha_n, S_{\alpha,n+1}) = \hat{E}_{n+1}$ where it relies on the E-ACK status in the future state.

Despite possible erasures over the acknowledgment link, the reward defined in this model fits into the decision problem in (6), where the corresponding objective becomes maximizing the expected discounted sum of E-ACK arrivals. In this sense, \hat{E}_n at the SA resembles E_n at the AA plus noise in the form of the E-ACK erasure.

2) *Modeling at the AA:* For modeling the problem at the AA, we have the components as follows:

States – We represent the state $S_{\beta,n}$ at the n -th time slot using a tuple $(\hat{v}_n, \Delta_n, \Theta_n)$, where \hat{v}_n is the usefulness of the received update from the perspective of the endpoint, Δ_n is the AoI, and Θ_n denotes the action lateness, as modeled in Section III. Without loss of generality, we assume the values of Δ_n and Θ_n are truncated by the maximum values notated as Δ_{\max} and Θ_{\max} , respectively, such that the conditions

$$g_\Delta(\hat{v}_n, \Delta_{\max}-1) \leq (1 + \varepsilon_\Delta)g_\Delta(\hat{v}_n, \Delta_{\max}), \quad (8)$$

for $\hat{v}_n \in \hat{\mathcal{V}}$, and

$$g_\Theta(\Theta_{\max}-1) \leq (1 + \varepsilon_\Theta)g_\Theta(\Theta_{\max}) \quad (9)$$

are met with the relevant accuracy ε_Δ and ε_Θ . Given this, at the AA, $S_{\beta,n}$ is a member of a finite and countable space \mathcal{S}_β having $|\mathcal{S}_\beta| = \Delta_{\max} \cdot \Theta_{\max} \cdot |\hat{\mathcal{V}}|$ states.

Actions – As already mentioned, β_n shows the decision of initiating a query at the n -th time slot and gets values from an action space $\mathcal{A}_\beta = \{0, 1\}$. Here, 0 and 1 depict refusing and confirming to initiate a query, respectively.

Transition probabilities – The transition probability from the current state $S_{\beta,n}$ to the future state $S_{\beta,n+1}$ under the action β_n is modeled as

$$p_\beta(S_{\beta,n}, \beta_n, S_{\beta,n+1}) = \Pr((\hat{v}_{n+1}, \Delta_{n+1}, \Theta_{n+1}) | (\hat{v}_n, \Delta_n, \Theta_n), \beta_n). \quad (10)$$

According to (10), we can write:

- $\Pr((\hat{v}_j, \min\{\Delta_n + 1, \Delta_{\max}\}, \min\{\Theta_n + 1, \Theta_{\max}\}) | (\hat{v}_j, \Delta_n, \Theta_n), \beta_n) = 1 - \beta_n,$
- $\Pr((\hat{v}_j, \min\{\Delta_n + 1, \Delta_{\max}\}, 1) | (\hat{v}_j, \Delta_n, \Theta_n), \beta_n) = \beta_n p_\epsilon,$
- $\Pr((\hat{v}_{j'}, 1, 1) | (\hat{v}_j, \Delta_n, \Theta_n), \beta_n) = \beta_n (1 - p_\epsilon) q_{j'},$

with $\hat{v}_j, \hat{v}_{j'} \in \hat{\mathcal{V}}$. For the rest of the transitions, we have $p_\beta(S_{\beta,n}, \beta_n, S_{\beta,n+1}) = 0$. As stated earlier, $q_j = p_{\hat{v}}(\hat{v}_j)$ with the pmf $p_{\hat{v}}(\cdot)$ derived in Section V-A.

Rewards – Arriving at the state $S_{\beta,n+1}$ from the state $S_{\beta,n}$ by taking the action β_n , is rewarded based on the effectiveness

level provided at the future state such that

$$r_\beta(S_{\beta,n}, \beta_n, S_{\beta,n+1}) = E_{n+1} = \mathbb{1}\{f_g(g_\Delta(\hat{v}_{n+1}, \Delta_{n+1}), g_\Theta(\Theta_{n+1})) \geq \text{GoE}_{\text{tgt}}\} \times \mathbb{1}\{\Theta_{n+1} < \Theta_{\max}\} \quad (11)$$

by the use of (1) and (4).

3) *Independence of the initial state:* Before we delve into the dual problem and solve it, we state and prove two propositions to show that the expected discounted sum of effectiveness in (6) is the same for all initial states.

Proposition 1. *The CMDP modeled at the SA satisfies the accessibility condition.*

Proof. Given the transition probabilities defined in (7), every state $S_{\alpha,m} \in \mathcal{S}_\alpha$, $m \leq N$, is accessible or reachable from the state $S_{\alpha,n}$ in finite steps with a non-zero probability, following the policy π_α . Therefore, the accessibility condition holds for the CMDP modeled at the SA [22, Definition 4.2.1].

Proposition 2. *The modeled CMDP at the AA meets the weak accessibility condition.*

Proof. We divide the state space \mathcal{S}_β into two disjoint spaces of \mathcal{T}_a and $\mathcal{T}_b = \mathcal{S}_\beta - \mathcal{T}_a$, where \mathcal{T}_a consists of all the states whose $\Delta_n = 1$, i.e., $\mathcal{T}_a = \{S_{\beta,n} | S_{\beta,n} = (\hat{v}_j, 1, \Theta_n), \forall \hat{v}_j \in \hat{\mathcal{V}}, \Theta_n = 1, 2, \dots, \Theta_{\max}\}$. Thus, \mathcal{T}_b includes the rest of the states with $\Delta_n \geq 2$. With regard to the transition probabilities derived in (10), all states of \mathcal{T}_b are transient under any policy, while every state of an arbitrary pair of two states in \mathcal{T}_a is accessible from the other state. Accordingly, the weak accessibility condition in the modeled CMDP at the AA is satisfied according to [22, Definition 4.2.2].

Given Propositions 1 and 2, we can show that the expected effectiveness obtained by \mathcal{P}_1 in (6) is the same for all initial states [22, Proposition 4.2.3]. In this regard, E_n , $\forall n$, is independent of E_0 for either model, thus we arrive at the following decision problem:

$$\begin{aligned} \mathcal{P}_2 : \max_{\pi_\gamma} \limsup_{N \rightarrow \infty} \frac{1}{N} \mathbb{E} \left[\sum_{n=1}^N \lambda^n E_n \right] \\ \text{s.t.} \quad \limsup_{N \rightarrow \infty} \frac{1}{N} \mathbb{E} \left[\sum_{n=1}^N \lambda^n c_\gamma(\gamma_n) \right] \leq C_{\gamma, \max} \end{aligned} \quad (12)$$

for $\gamma = \{\alpha, \beta\}$. Applying Propositions 1 and 2 confirms that there exist stationary optimal policies π_α^* and π_β^* for \mathcal{P}_2 solved at the SA and the AA, respectively, where both policies are *unichain* [22, Proposition 4.2.6].

C. Dual Problem

To solve the decision problem \mathcal{P}_2 given in (12), we first define an unconstrained form for the problem via dualizing the constraint. Then, we propose an algorithm to compute the decision policies for both agents.

The unconstrained form of the problem is derived by writing the Lagrange function $\mathcal{L}(\mu; \pi_\gamma)$ as below

$$\mathcal{L}(\mu; \pi_\gamma) = \limsup_{N \rightarrow \infty} \frac{1}{N} \mathbb{E} \left[\sum_{n=1}^N \lambda^n (E_n - \mu c_\gamma(\gamma_n)) \right] + \mu C_{\gamma, \max} \quad (13)$$

with $\mu \geq 0$ being the Lagrange multiplier. According to (13), we arrive at the following dual problem to be solved:

$$\mathcal{P}_3 : \inf_{\mu \geq 0} \max_{\pi_\gamma} \underbrace{\mathcal{L}(\mu; \pi_\gamma)}_{:= h_\gamma(\mu)} \quad (14)$$

where $h_\gamma(\mu) = \mathcal{L}(\mu; \pi_{\gamma, \mu}^*)$ is the Lagrange dual function with $\pi_{\gamma, \mu}^* : \mathcal{S}_\gamma \rightarrow \mathcal{A}_\gamma$ denoting a stationary μ -optimal policy, which is obtained as

$$\pi_{\gamma, \mu}^* = \arg \max_{\pi_\gamma} \mathcal{L}(\mu; \pi_\gamma) \quad (15)$$

for μ derived in the dual problem \mathcal{P}_3 . As the dimension of the state space \mathcal{S}_γ is finite for both defined models, the growth condition is met [23]. Furthermore, the immediate reward and the induced cost are bounded according to Section IV. Given these satisfied conditions, from [23, Corollary 12.2], we can assert that \mathcal{P}_2 and \mathcal{P}_3 converge to the same expected values such that we can write

$$\inf_{\mu \geq 0} \max_{\pi_\gamma} \mathcal{L}(\mu; \pi_\gamma) = \inf_{\mu \geq 0} \mathcal{L}(\mu; \pi_\gamma^*) = \max_{\pi_\gamma} \mathcal{L}(\mu^*; \pi_\gamma) \quad (16)$$

for some policy class π_γ^* . Given the conditions satisfied, there exist non-negative optimal values for the Lagrange multiplier μ^* under Slater's condition such that (16) holds [23, Theorem 12.8], establishing the strong duality between \mathcal{P}_2 and \mathcal{P}_3 .

We can now proceed to derive the optimal policies at the SA and the AA from the decision problem \mathcal{P}_3 by applying an *iterative algorithm* in line with the dynamic programming approach based on (13)–(15) [14].

D. Iterative Algorithm

The iterative algorithm is given in Algorithm 1 and consists of two *inner* and *outer* loops. The inner loop is for computing the μ -optimal policy, i.e., $\pi_{\gamma, \mu}^*$, using the *value iteration* method. Over the outer loop, the optimal Lagrange multiplier μ^* is derived via the *bisection search* method.

1) *Computing $\pi_{\gamma, \mu}^*$* : Applying the value iteration method, the decision policy is iteratively improved given μ from the outer loop (bisection search). Thus, $\pi_{\gamma, \mu}(s) \in \mathcal{A}_\gamma, \forall s \in \mathcal{S}_\gamma$, is updated such that it maximizes the *expected utility* (value) $V_k^{\pi_{\gamma, \mu}}(s)$ at the k -th, $\forall k \in \mathbb{N}$, iteration, which is obtained as

$$V_k^{\pi_{\gamma, \mu}}(s) = \mathbb{E}[r_k + \lambda r_{k+1} + \lambda^2 r_{k+2} + \dots | s_k = s] \approx \mathbb{E}[r_k + \lambda V_{k-1}^{\pi_{\gamma, \mu}} | s_k = s] \quad (17)$$

where s_k denotes the state at the k -th iteration, and r_k is the corresponding reward at that state. The approximation in (17) appears after bootstrapping the rest of the discounted sum of the rewards by the value estimate $V_{k-1}^{\pi_{\gamma, \mu}}$. Under the form of the

Algorithm 1: Solution for deriving π_γ^* and μ^*

Input: Given parameters $N \gg 1, C_{\gamma, \max}, \eta, \varepsilon_\mu$, CMDP's state space, i.e., \mathcal{S}_γ , and action space, i.e., \mathcal{A}_γ . The form of the cost function $c_\gamma(\cdot)$. Initial values $l \leftarrow 0, \mu_-^{(0)} \leftarrow 0, \mu_+^{(0)} \leftarrow 0, \mu_+^{(0)} > 0, \pi_{\gamma, \mu_-^{(0)}} \leftarrow 0$, and $\pi_{\gamma, \mu_+^{(0)}} \leftarrow 0$.

- 1 Derive $\pi_{\gamma, \mu}^*(s), \forall s \in \mathcal{S}_\gamma$, via running `policy` ($\mu^{(0)}$).
 - 2 **if** $\mathbb{E}[\sum_{n=1}^N c_\gamma(\gamma_n)] \leq NC_{\gamma, \max}$ **then goto** 11.
 - 3 **while** $|\mu_+^{(l)} - \mu_-^{(l)}| \geq \varepsilon_\mu$ **do**
 - Step l:* ▷ Outer loop (Bisection search)
 - 4 set $l \leftarrow l + 1, \mu_-^{(l)} \leftarrow \mu_-^{(l-1)}$, and $\mu_+^{(l)} \leftarrow \mu_+^{(l-1)}$.
 - 5 Update $\mu^{(l)} \leftarrow \frac{\mu_-^{(l)} + \mu_+^{(l)}}{2}$.
 - 6 Improve $\pi_{\gamma, \mu}^* \leftarrow \text{policy}(\mu^{(l)})$.
 - 7 **if** $\mathbb{E}[\sum_{n=1}^N c_\gamma(\gamma_n)] \geq NC_{\gamma, \max}$ **then**
 - 8 $\mu_-^{(l)} \leftarrow \mu^{(l)}$, and $\pi_{\gamma, \mu_-^{(l)}} \leftarrow \text{policy}(\mu_-^{(l)})$.
 - 9 **else** $\mu_+^{(l)} \leftarrow \mu^{(l)}$, and $\pi_{\gamma, \mu_+^{(l)}} \leftarrow \text{policy}(\mu_+^{(l)})$.
 - 10 **if** $\mathbb{E}[\sum_{n=1}^N c_\gamma(\gamma_n)] < NC_{\gamma, \max}$ **then**
 - $\pi_{\gamma, \mu}^*(s) \leftarrow \eta \pi_{\gamma, \mu_-^{(l)}}(s) + (1 - \eta) \pi_{\gamma, \mu_+^{(l)}}(s), \forall s \in \mathcal{S}_\gamma$.
 - 11 **return** $\mu^* = \mu^{(l)}$ and $\pi_\gamma^*(s) = \pi_{\gamma, \mu}^*(s), \forall s \in \mathcal{S}_\gamma$.
-

Function `policy` (μ):

Input: Known parameters from the outer loop.

Initial values $k \leftarrow 1, \pi_{\gamma, \mu}(s) \leftarrow 0$, and

$V_k^{\pi_{\gamma, \mu}}(s) \leftarrow 0, \forall s \in \mathcal{S}_\gamma$.

Iteration k: ▷ Inner loop (Value iteration)

- 12 **for** state $s \in \mathcal{S}_\gamma$ **do**
 - 13 compute $V_k^{\pi_{\gamma, \mu}}(s)$ from (18).
 - 14 Improve $\pi_{\gamma, \mu}(s)$ according to (19) and (20).
 - 15 **if** $\text{sp}(V_k^{\pi_{\gamma, \mu}} - V_{k-1}^{\pi_{\gamma, \mu}}) \geq \varepsilon_\pi$ **as in** (22) **then**
 - 16 step up $k \leftarrow k + 1$, and **goto** 12.
 - 17 **return** $\pi_{\gamma, \mu}^*(s) = \pi_{\gamma, \mu}(s), \forall s \in \mathcal{S}_\gamma$.
-

value iteration for the unichain policy MDPs [24], the optimal value function is derived from Bellman's equation [25], as

$$V_k^{\pi_{\gamma, \mu}}(s) = \max_{\gamma \in \mathcal{A}_\gamma} \sum_{s' \in \mathcal{S}_\gamma} p_\gamma(s, \gamma, s') [r_{\gamma, \mu}(s, \gamma, s') + \lambda V_{k-1}^{\pi_{\gamma, \mu}}(s')] \quad (18)$$

for the state $s \in \mathcal{S}_\gamma$. Consequently, the decision policy in that state is improved by

$$\pi_{\gamma, \mu}(s) \in \arg \max_{\gamma \in \mathcal{A}_\gamma} \sum_{s' \in \mathcal{S}_\gamma} p_\gamma(s, \gamma, s') [r_{\gamma, \mu}(s, \gamma, s') + \lambda V_{k-1}^{\pi_{\gamma, \mu}}(s')]. \quad (19)$$

In (18) and (19), we define a *net* reward function as

$$r_{\gamma, \mu}(s, \gamma, s') = r_\gamma(s, \gamma, s') - \mu c_\gamma(\gamma), \quad (20)$$

which takes into account the cost caused by the action taken.

The value iteration stops running at the k -th iteration once the following convergence criterion is met [24]:

$$\text{sp}(V_k^{\pi_{\gamma, \mu}} - V_{k-1}^{\pi_{\gamma, \mu}}) < \varepsilon_\pi \quad (21)$$

where $\varepsilon_\pi > 0$ is the desired convergence accuracy, and $\text{sp}(\cdot)$ indicates a *span* function $\mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ given as

$$\text{sp}(V_{k'}^{\pi_{\gamma,\mu}}) = \max_{s \in \mathcal{S}} V_{k'}^{\pi_{\gamma,\mu}}(s) - \min_{s \in \mathcal{S}} V_{k'}^{\pi_{\gamma,\mu}}(s) \quad (22)$$

by using the span seminorm for the arbitrary k' -th iteration [24, Section 6.6.1]. As the decision policies are unichain and have aperiodic transition matrices, the criterion in (21) is satisfied after finite iterations for any value of $\lambda \in [0, 1]$ [24, Theorem 8.5.4].

2) *Computing μ^** : We leverage the bisection search method to compute the optimal Lagrange multiplier over multiple steps in the outer loop based on the derived $\pi_{\gamma,\mu}^*$ from the inner loop. Starting with an initial interval $[\mu_-^{(0)}, \mu_+^{(0)}]$ such that $h_\gamma(\mu_-^{(0)})h_\gamma(\mu_+^{(0)}) < 0$, the value of the multiplier at the l -th, $\forall l \in \mathbb{N}$, step is improved by $\mu^{(l)} = \frac{\mu_-^{(l)} + \mu_+^{(l)}}{2}$. As shown in Algorithm 1, at each step, the value of either $\mu_-^{(l)}$ or $\mu_+^{(l)}$ and the corresponding decision policy $\pi_{\gamma,\mu_-^{(l)}}$ or $\pi_{\gamma,\mu_+^{(l)}}$, respectively, are updated according to the cost constraint in (12) until a stopping criterion $|\mu_+^{(l)} - \mu_-^{(l)}| < \varepsilon_\mu$ is reached with the accuracy ε_μ . Considering (13) and (14), $h_\gamma(\mu)$ is a non-increasing function of μ in the unsettled region where the cost constraint has not yet been met. In this regard, the bisection method searches for the smallest Lagrange multiplier that guarantees the constraint. Also, one can show that $h_\gamma(\mu)$ denotes a Lipschitz continuous function with the Lipschitz constant as below

$$\left| C_{\gamma,\max} - \limsup_{N \rightarrow \infty} \frac{1}{N} \mathbb{E} \left[\sum_{n=1}^N \lambda^n c_\gamma(\gamma_n) \right] \right|.$$

Therefore, the bisection search converges to the optimal value of μ within finite steps [26, pp. 294].

After the outer loop stops running, we obtain a stationary *deterministic* decision policy as $\pi_\gamma^* = \pi_{\gamma,\mu}$ if the following condition holds:

$$\limsup_{N \rightarrow \infty} \frac{1}{N} \mathbb{E} \left[\sum_{n=1}^N \lambda^n c_\gamma(\gamma_n) \right] = C_{\gamma,\max}. \quad (23)$$

Otherwise, the derived policy becomes *randomized* stationary in the shape of mixing two deterministic policies $\pi_{\gamma,\mu_-^{(l)}}$ and $\pi_{\gamma,\mu_+^{(l)}}$ with probability $\eta \in [0, 1]$ [27], where

$$\pi_{\gamma,\mu_-^{(l)}} = \lim_{\mu \rightarrow \mu_-^{(l)}} \pi_{\gamma,\mu}, \quad \pi_{\gamma,\mu_+^{(l)}} = \lim_{\mu \rightarrow \mu_+^{(l)}} \pi_{\gamma,\mu}$$

in the l -th step, after which the bisection search terminates. Hence, we can write

$$\pi_\gamma^* \leftarrow \eta \pi_{\gamma,\mu_-^{(l)}} + (1 - \eta) \pi_{\gamma,\mu_+^{(l)}}, \quad (24)$$

which implies that the decision policy is randomly chosen as $\pi_\gamma^* = \pi_{\gamma,\mu_-^{(l)}}$ and $\pi_\gamma^* = \pi_{\gamma,\mu_+^{(l)}}$ with probabilities η and $1 - \eta$, respectively. In (24), η is computed such that the condition in (23) is maintained.

3) *Complexity analysis*: The value iteration approach in the inner loop is *polynomial* with $\mathcal{O}(|\mathcal{A}_\gamma||\mathcal{S}_\gamma|^2)$ arithmetic operations at each iteration. Besides, the bisection search takes



Fig. 3. Time partitioning of the estimation and decision horizons.

$\lceil \log_2(\frac{\mu_+^{(0)}}{\varepsilon_\mu}) \rceil$ steps to reach the optimal Lagrange multiplier within the tolerance of ε_μ , given the derived policy. Here, $\mu_+^{(0)}$ is the upper bound of the initial interval for the multiplier, with the lower bound fixed at zero. Thereby, the overall complexity of Algorithm 1, in terms of the number of arithmetic operations across both loops, can be calculated as follows

$$\mathcal{O} \left(\frac{2|\mathcal{S}_\gamma|^2}{1 - \lambda} \log \left(\frac{1}{1 - \lambda} \right) \log \left(\frac{\mu_+^{(0)}}{\varepsilon_\mu} \right) \right)$$

based on the fixed $\lambda < 1$ (see [28] and [29]). The algorithm's complexity increases as the state space expands, the initial interval for the multiplier widens, $\lambda \rightarrow 1$, and $\varepsilon_\mu \rightarrow 0$.

V. MONTE CARLO PROBABILITY DISTRIBUTION ESTIMATION

In this section, we leverage the *Monte Carlo* estimation method to statistically compute the estimated pmfs of the received updates' usefulness from the endpoint's perspective at the AA, and the mapped target usefulness at the SA. To this end, we consider a time interval in the format of an *estimation horizon* (E-horizon), followed by a *decision horizon* (D-horizon), as illustrated in Fig. 3. The E-horizon is exclusively reserved for the estimation processes and has a length of M time slots, which is sufficiently large to enable an accurate estimation. The D-horizon represents the long-term time horizon with a sufficiently large length of $N \gg 1$ slots, as defined in Section IV, during which the agents find and apply their (model) CMDP-based decision policies.

Within the E-horizon, the SA does not make any decisions. Instead, it focuses on communicating updates at the highest possible rate while adhering to cost constraints. Once receiving these updates, the AA measures their usefulness, stores them in memory, and sends E-ACK signals for effective updates. The SA logs whether the E-ACK has been successfully received or not at every slot of the E-horizon. Finally, both agents perform their estimations by employing the received E-ACK signals at the SA and the measured updates' usefulness at the AA.

A. Usefulness Probability of Received Updates

Picking the j -th, $\forall j \in \mathcal{J}$, outcome from the set $\hat{\mathcal{V}}$ (defined in Section III-A) that corresponds to the received update's usefulness from the endpoint's perspective at the m -th slot of the E-horizon, i.e., \hat{v}_m , the relevant estimated probability of that outcome is given by

$$q_j = p_{\hat{v}}(\hat{v}_j) = \frac{1}{M} \sum_{m=1}^M \mathbb{1}\{\hat{v}_m = \hat{v}_j\}. \quad (25)$$

B. Probability of the Mapped Target Usefulness

We assume that the mapped target usefulness, i.e., \hat{v}_{tgt} , is a member of the set $\hat{\mathcal{V}}_{\text{tgt}} = \{\vartheta_j | j \in \mathcal{J}_{\text{tgt}}\}$ with i.i.d. elements, where $\mathcal{J}_{\text{tgt}} = \{1, 2, \dots, |\hat{\mathcal{V}}_{\text{tgt}}|\}$, and the probability of the j -th element is equal to $p_{\hat{v}_{\text{tgt}}}(\hat{v}_{\text{tgt}} = \vartheta_j)$. Herein, as mentioned earlier, $p_{\hat{v}_{\text{tgt}}}(\cdot)$ is the estimated pmf of the mapped target usefulness and obtained by

$$p_{\hat{v}_{\text{tgt}}}(\vartheta_j) = \sum_{e \in \{0,1\}} p_{\hat{v}_{\text{tgt}}}(\vartheta_j | \hat{E} = e) \Pr(\hat{E} = e) \quad (26)$$

where we find the probability of successfully receiving E-ACK, i.e., $e = 1$, or not, i.e., $e = 0$, as follows

$$\Pr(\hat{E} = e) = \frac{1}{M} \sum_{m=1}^M \mathbb{1}\{\hat{E}_m = e\} \quad (27)$$

where \hat{E}_m indicates the E-ACK arrival status at the m -th slot of the E-horizon. Furthermore, to derive the conditional probability in (26), we first consider the successful arrivals of E-ACK signals such that

$$p_{\hat{v}_{\text{tgt}}}(\vartheta_j | \hat{E} = 1) = \frac{\sum_{i \in \mathcal{I}} p_{\nu | \hat{E}=1}(\nu_i | \nu_i \geq \vartheta_j)}{\sum_{j \in \mathcal{J}_{\text{tgt}}} \sum_{i \in \mathcal{I}} p_{\nu | \hat{E}=1}(\nu_i | \nu_i \geq \vartheta_j)}. \quad (28)$$

Then, we have

$$p_{\hat{v}_{\text{tgt}}}(\vartheta_j | \hat{E} = 0) = \frac{\sum_{i \in \mathcal{I}} p_{\nu | \hat{E}=0}(\nu_i | \nu_i < \vartheta_j)}{\sum_{j \in \mathcal{J}_{\text{tgt}}} \sum_{i \in \mathcal{I}} p_{\nu | \hat{E}=0}(\nu_i | \nu_i < \vartheta_j)}. \quad (29)$$

The pmfs $p_{\nu | \hat{E}=1}(\cdot)$ and $p_{\nu | \hat{E}=0}(\cdot)$ in (28) and (29) are associated with an observation's importance rank given the successful and unsuccessful communication of the E-ACK, respectively. In this regard, by applying Bayes' theorem, we can derive the following formula:

$$p_{\nu | \hat{E}=e}(\nu_i) = \frac{\frac{1}{M} \sum_{m=1}^M \mathbb{1}\{v_m = \nu_i \wedge \hat{E}_m = e\}}{\Pr(\hat{E} = e)}. \quad (30)$$

VI. SIMULATION RESULTS

In this section, we present simulation results that corroborate our analysis and assess the performance gains in terms of effectiveness achieved by applying different update models and agent decision policies in end-to-end status update systems.

A. Setup and Assumptions

We study the performance over 5×10^5 time slots, which includes the E-horizon and the D-horizon with 1×10^5 and 4×10^5 slots, respectively. To model the Markovian effect-agnostic policy, we consider a Markov chain with two states, 0 and 1. We assume that the self-transition probability of state 0 is 0.9, while the one for state 1 relies on the controlled update transmission or query rate. Without loss of generality, we assume that the outcome spaces for the usefulness of generated updates, i.e., \mathcal{V} , the usefulness of received updates, i.e., $\hat{\mathcal{V}}$, and the mapped target usefulness, i.e., $\hat{\mathcal{V}}_{\text{tgt}}$, are bounded within the span $[0, 1]$. For simplicity, we divide each space into discrete levels based on the number of its elements in ascending order,

TABLE I
PARAMETERS FOR SIMULATION RESULTS

Name	Symbol	Value
E-horizon length	—	1×10^5 [slot]
D-horizon length	N	4×10^5 [slot]
Erasure probability in update channel	p_ϵ	0.2
Erasure probability in acknowledgment link	p'_ϵ	0.1
Length of generated update usefulness space	$ \mathcal{V} $	10
Length of received update usefulness space	$ \hat{\mathcal{V}} $	11
Length of mapped target usefulness space	$ \hat{\mathcal{V}}_{\text{tgt}} $	11
Shape parameters for usefulness distribution	a	0.3
	b	0.3
Maximum truncated AoI	Δ_{\max}	10 [slot]
Action window width in pull-based model	Θ_{\max}	1 [slot]
Action window width in push-and-pull model		5 [slot]
Action window width in push-based model		10 [slot]
Update transmission cost at n -th slot	$C_{n,1}$	0.1
Query initiation cost at n -th slot	$C_{n,2}$	0.1
Actuation availability cost at n -th slot	$C_{n,3}$	0.01
Maximum discounted cost in decision problem	$C_{\gamma, \max}$	0.08
Target effectiveness grade	GoE_{tgt}	0.6
Discount factor in CMDP	λ	0.75
Convergence accuracy in Algorithm 1	ε_μ	10^{-4}
	ε_π	10^{-4}
Mixing probability in bisection method	η	0.5
Controlled transmission rate (effect-agnostic)	—	0.8
Controlled query rate (effect-agnostic)	—	0.8

where every level shows a randomized value. We also consider that the i -th outcome of the set \mathcal{V} notated as ν_i , $i \in \mathcal{I}$, occurs following a *beta-binomial* distribution with pmf

$$p_\nu(\nu_i) = \binom{|\mathcal{V}| - 1}{i - 1} \frac{\text{Beta}(i - 1 + a, |\mathcal{V}| - i + b)}{\text{Beta}(a, b)} \quad (31)$$

where $\text{Beta}(\cdot, \cdot)$ is the beta function, and $a = 0.3$ and $b = 0.3$ are shape parameters.

Moreover, we plot the figures based on the following form of the GoE metric⁵, which comes from the general formulation proposed in (1):

$$\text{GoE}_n = \frac{\hat{v}_n}{\Delta_n \Theta_n} - \alpha_n C_{n,1} - \beta_n C_{n,2} - C_{n,3} \quad (32)$$

for the n -th time slot. Herein, $C_{n,1}$ is the communication cost, $C_{n,2}$ denotes the query cost, and $C_{n,3}$ indicates the actuation availability cost at the AA which depends on the update communication model. However, the cost function in the decision problem's constraint is assumed to be $c_\gamma(\gamma_n) = \gamma_n$, $\forall n \in \mathbb{N}$. The parameters used in the simulations are summarized in Table I. In the legends of the plotted figures, we depict the policies at the agents in the form of a tuple, with the first and second elements referring to the policy applied in the SA and the AA, respectively. For an effect-aware policy, we use the notation "E-aware," while for an effect-agnostic policy, only

⁵The analysis can be easily extended to any other form of the GoE metric.

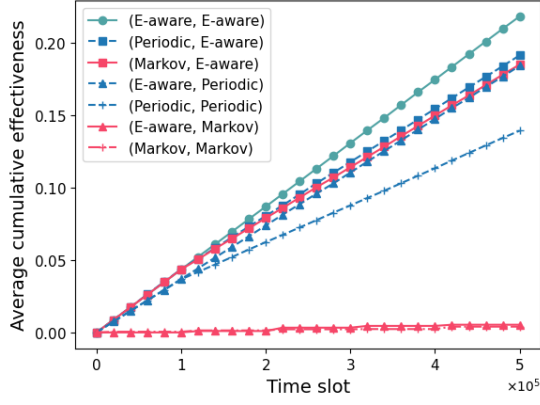


Fig. 4. Evolution of the average effectiveness accumulated over time in the push-and-pull model.

the modeling process is mentioned in the legend.

B. Results and Discussion

Fig. 4 illustrates the evolution of the average cumulative effectiveness over time for the push-and-pull model and different agent decision policies. As shown, applying the effect-aware policy in both agents offers the highest effectiveness, where the gap between its performance and the other policies' performance increases gradually as time passes. Nevertheless, using the other effect-agnostic policies at either or both agents diminishes the effectiveness performance of the system by at least 12% or 36%, respectively. Suppose the SA and the AA apply the effect-aware and Markovian effect-agnostic policies sequentially. In that case, the offered effectiveness is even lower than the scenario in which both agents use periodic effect-agnostic policies. One of the reasons is that the Markovian query initiation misleads the SA in estimating the mapped target usefulness of updates. It is also worth mentioning that throughout the E-horizon from slot 1 to 1×10^5 , the scenarios where the AA initiates effect-aware queries demonstrate the same performance and are better than the other scenarios. However, in the D-horizon, a performance gap appears and evolves depending on the applied policy at the SA. Therefore, by making effect-aware decisions at both agents based on estimations in the E-horizon and CMDP-based policies in the D-horizon, performance can be significantly improved.

The bar chart in Fig. 5 shows the average rate of update transmissions from the SA and the consequent actions performed at the AA based on the initiated queries that result in the system's effectiveness, as depicted in Fig. 4. Interestingly, the scenario where both agents apply the effect-aware policy has lower transmission and action rates compared to the other scenarios, except the ones with Markovian effect-agnostic query policies. However, the number of actions that can be taken for those scenarios with Markovian queries is limited since most updates are received outside the action windows. Figs. 4 and 5 show that using effect-aware policies at both the SA and the AA not only brings the highest effectiveness but also needs lower update transmissions by an average of 11%, saving resources compared to the scenarios that have

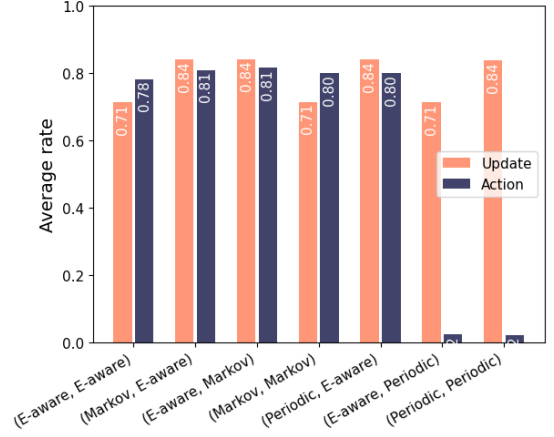


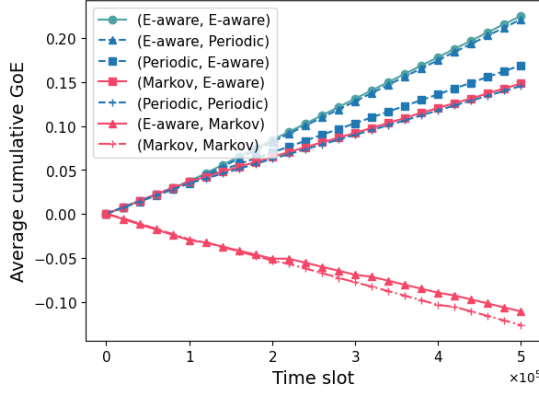
Fig. 5. Average update transmission and query rates for different decision policies in the push-and-pull model.

comparable performances. We also note that although effect-aware and periodic query decisions with effect-aware update transmission have almost the same action rate, the effect-aware case leads to more desirable effects or appropriate actions at the endpoint. This results in around 16% higher average cumulative effectiveness, referring to Fig. 4.

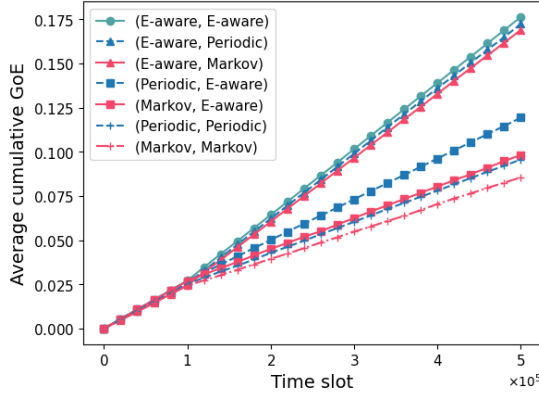
Figs. 6(a), 6(b), and 6(c) present the average cumulative GoE provided in the system over 5×10^5 time slots for the push-and-pull, push- and pull-based communication models, respectively. The corresponding effectiveness at the endpoint for the primary model can be found in Fig. 4. The plots demonstrate that when both agents decide based on effect-aware policies, regardless of the update communication model, the highest offered GoE of the system is reached. However, in other scenarios, the performance of some policies exceeds that of others, depending on the update model. For instance, having effect-aware update transmission and Markovian queries shows a 2.52 times higher average GoE for the push-based model than the push-and-pull one. This is because the push-based model has a larger action window.

Besides, comparing Figs. 6(a) and 6(b), the reason that the provided GoE by the effect-aware decisions at both agents is 28% lower for the push-based model compared to the push-and-pull is that the AA has to be available longer, which causes a higher cost. Also, with the pull-based model as in Fig. 6(c), the average GoE within a period is less than the average cost since the AA is only available to act at query instants, significantly reducing the average GoE despite the high update transmission rate. In the pull-based model, however, applying the CMDP-based update transmission decisions at the SA can address this issue with a 16% higher average GoE.

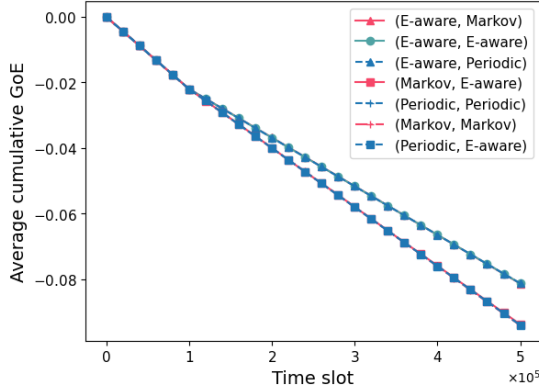
The trade-off between the average effectiveness and the width of the action window, i.e., Θ_{\max} , is shown in Fig. 7 for different decision policies. We see that the system cannot offer notable effectiveness with $\Theta_{\max} = 1$, i.e., under the pull-based model. However, by expanding the action window width from $\Theta_{\max} = 1$ to 10 [slot], the average effectiveness boosts from its lowest to the highest possible value. Since all policies already reach their best performance before $\Theta_{\max} = 10$, which



(a)



(b)



(c)

Fig. 6. Temporal evolution of the average cumulative GoE for the (a) push-and-pull, (b) push-based, and (c) pull-based models.

indicates the push-based model, we can conclude that the push-and-pull model with a flexible action window is more advantageous than the push-based one with a very large fixed window. In addition, the scenario where both agents use effect-aware policies outperforms the others for $\Theta_{\max} > 1$. It is worth mentioning that performance converges to the highest level for sufficiently large widths in the scenarios where the AA initiates Markovian queries. As depicted in Fig. 4, the average cumulative effectiveness of these scenarios visibly rises for $\Theta_{\max} \geq 9$.

The interplay between the average effectiveness and the

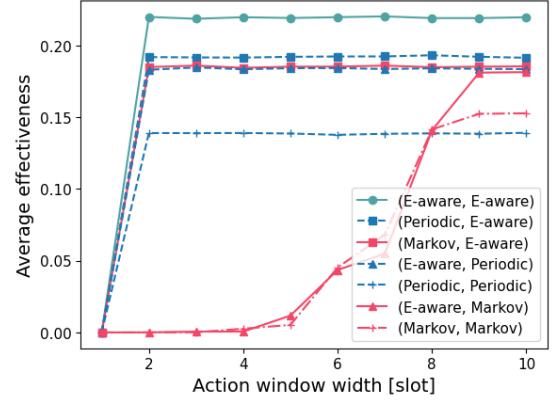


Fig. 7. Average effectiveness versus the action window width in the push-and-pull communication model.

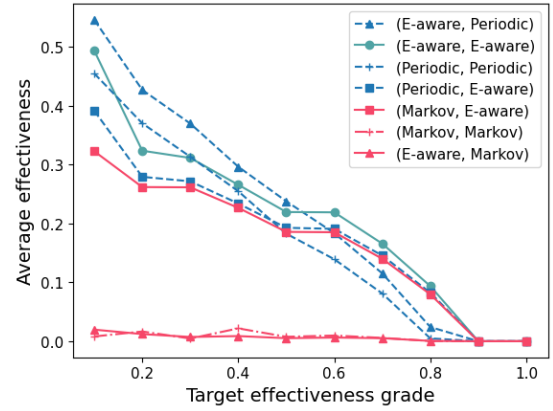


Fig. 8. Average effectiveness as a function of the target effectiveness grade in the push-and-pull model.

target effectiveness grade, i.e., GoE_{tgt} from Section III-C, is depicted in Fig. 8 for the push-and-pull model and different agent decision policies. Evidently, the average effectiveness under all policies decreases gradually with the increase of the target effectiveness grade and converges to zero for $\text{GoE}_{\text{tgt}} \geq 0.9$. Using effect-aware policies at both agents offers the highest effectiveness for medium-to-large target grades, i.e., $\text{GoE}_{\text{tgt}} \geq 0.52$ here. However, for lower target grades, the effect-aware and periodic effect-agnostic decisions at the SA and the AA, respectively, result in better performance. This comes at the cost of higher transmission and action rates. Thus, a trade-off emerges between incurred cost and offered effectiveness, where different policies can be applied depending on the chosen cost criterion and the target effectiveness grade.

Fig. 9 depicts the average effectiveness obtained in the system through 5×10^5 time slots versus the controlled update transmission rate for the push-and-pull model and various agent decision policies. Concerning Section II-B, this controlled rate is related to the effect-agnostic update transmission policies at the SA and denotes the expected number of updates to be communicated within the specified period. Therefore, the performance of the other policies should remain fixed for different controlled transmission rates. Fig. 9 reveals that increasing the controlled update rate increases the offered

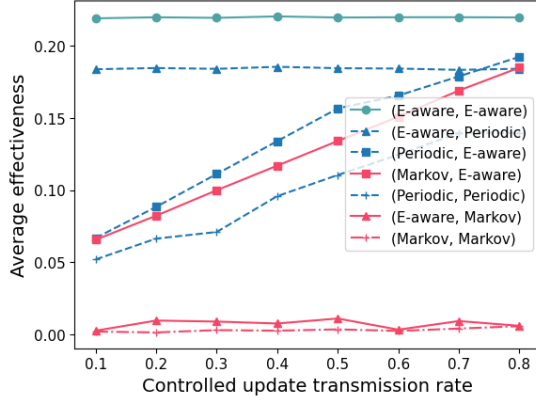


Fig. 9. Comparison between different policies with variable update transmission rates but fixed query rates for effect-agnostic policies.

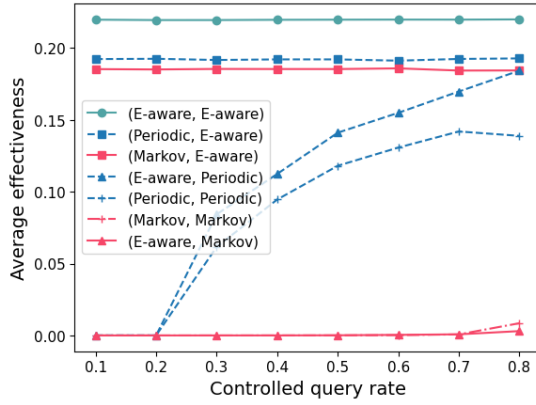


Fig. 10. Comparison between different policies with variable query rates but fixed update transmission rates for effect-agnostic policies.

effectiveness when the SA applies the effect-agnostic policies. However, even at the highest possible rate, subject to the maximum discounted cost, using effect-aware policies at both agents is necessary to ensure the highest average effectiveness, regardless of the controlled transmission rate. As an illustrative example, when the AA initiates effect-aware queries, the effectiveness drops by an average of 38% (43%) if the SA transmits updates based on the periodic (Markovian) effect-agnostic policy instead of using the effect-aware one.

In Fig. 10, the plot shows the same trend as Fig. 9, but this time it focuses on the changes in the average effectiveness versus the controlled query rate. Herein, the controlled query rate is dedicated to the scenarios where the AA operates under effect-agnostic policies. The results indicate that the average effectiveness rises with the increase of the controlled query rate for the periodic and Markovian policies. However, the increase in effectiveness is not significant for the latter one. Despite this, even with the highest rates, the effectiveness offered in the course of effect-aware queries is still higher than that of initiating effect-agnostic queries. Therefore, the highest average effectiveness is achieved when both agents make effect-aware decisions, as depicted in Figs. 9 and 10.

In the context of the decision-making problem \mathcal{P}_2 in (12), altering the maximum discounted cost, i.e., $C_{\gamma, \max}$, can impact

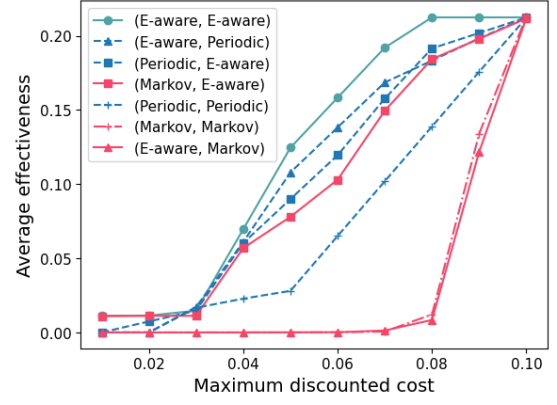


Fig. 11. Average effectiveness achieved through different policies versus the maximum discounted cost in the push-and-pull model.

the decisions made by each agent. To study this, we have plotted Fig. 11 for the push-and-pull model under different decision policies. The figure shows that the stricter the cost constraint, the lower the average effectiveness, irrespective of the decision policy. Also, for all cost constraints, the scenario in which both agents use effect-aware policy yields the best performance, whereas the other policies outperform each other under different cost constraints. Due to CMDP-based decisions, the gap between the effectiveness performance of the best scenario and that of the others increases as the constraint decreases until $C_{\gamma, \max} = 0.1$, where the SA can transmit all updates, and the AA can initiate queries without restrictions.

According to the CMDP modeled at either agent, model-based effect-aware policies heavily rely on the conditions of the update and/or E-ACK channels. To evaluate this impact, Fig. 12 illustrates the average effectiveness as a function of the channel erasure probability. For clarity, we assume $p_e = p'_e$, where p_e and p'_e denote the erasure probabilities in the forward communication and the acknowledgment links, respectively. At lower erasure probabilities, particularly when $p_e \leq 0.52$, employing effect-aware policies at both agents yields the highest average effectiveness. However, as channel conditions worsen, the performance of effect-aware decision-making gradually declines. Notably, when $p_e \geq 0.66$, the effectiveness of pulling updates using effect-aware policies drops to its lowest levels.

Afterward, we compare the performance provided by model-based agent decisions discussed in Section IV with that of *model-free* decisions. The latter is based on reinforcement learning (RL), where each agent separately learns to make decisions through direct interaction with the environment. This learning process is modeled under the state (here, observation) spaces, action sets, and rewards according to Section IV-B, without relying on the construction of a predefined model. To derive model-free decisions, we employ a *deep Q-network* (DQN) and parameterize an approximate value function for every agent within the E-horizon through a multilayer perceptron (MLP), assisted with the experience replay mechanism. The neural network consists of two hidden layers, each with 64 neurons, and is trained using the adaptive moment estimation

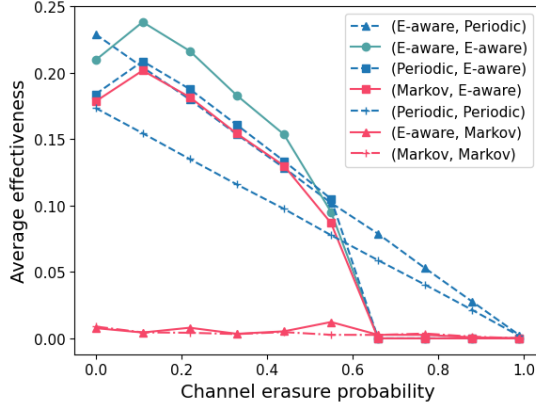


Fig. 12. Interplay between average effectiveness and erasure probability for both the update and E-ACK channels in the push-and-pull model.

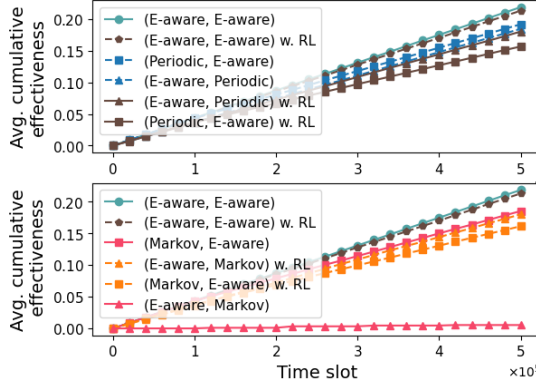


Fig. 13. Temporal evolution of average cumulative effectiveness under model-based and model-free (RL-based) policies in the push-and-pull model.

(Adam) optimizer. The default values for the RL setting are taken from [30], except for the learning rate that is 10^{-4} , and the discount factor is assumed to be 0.75, as aligned with Table I. In this regard, Fig. 13 depicts the evolution of the average cumulative effectiveness over time for the push-and-pull model. The graph compares the performance achieved by model-based agent decisions with that of model-free (RL-based) ones. As shown, the model-based approach outperforms the model-free approach across all decision policies. However, the performance gap is less significant when both agents apply effect-aware policies. Considering the CMDP modeled at both agents, several factors may contribute to the higher effectiveness of the model-based approach, such as the relatively small state space, especially for the modeling at the SA.

Additionally, the length of the E-horizon potentially influences the performance and complexity of both decision-making approaches. To analyze this impact, Fig. 14 demonstrates how the E-horizon length affects the average effectiveness of effect-agnostic and effect-aware policies under both model-based and model-free approaches for the push-and-pull model. The results show that increasing the E-horizon length enhances performance, reaching its peak at an optimal E-horizon length. This optimal length value varies depending on the policy and approach employed. For instance, when both

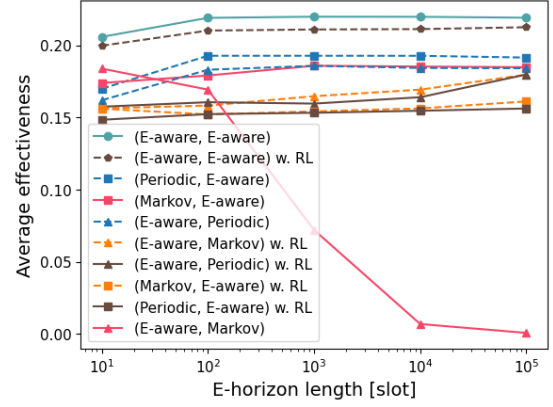


Fig. 14. Impact of E-horizon length on the effectiveness of model-based and model-free (RL-based) policies in the push-and-pull model.

agents employ effect-aware policies, achieving the highest effectiveness requires approximately $10^{2.9}$ and $10^{3.7}$ slots for the model-based and model-free approaches, respectively, with convergence accuracy specified in Table I. In this context, the required E-horizon length serves as a measure of the complexity of the corresponding algorithm for each approach. For the above example, the model-based algorithm (Algorithm 1) achieves a complexity that is 6.31 times lower compared to the model-free DQN algorithm.

C. Lookup Maps for Agent Decisions

As the modeled CMDPs in Section IV-B have finite states, we can depict optimal model-based decisions derived in Algorithm 1 via a multi-dimensional *lookup map* for each decision-making agent. Figs. 15 and 16 illustrate the maps for the SA and the AA, respectively, under different maximum discounted costs. The number of dimensions in a map relies on the number of elements constructing every state of the relevant CMDP, with each dimension assigned to one element. With the lookup map in hand, an agent can make optimal decisions at each slot based on its current state. When comparing the same maps for different maximum discounted costs, the observation emerges that the more stringent the cost constraint is, the narrower the agent decision boundaries become. Thus, the maps could vary by changes in the parameters' values or goals with different target effectiveness grades. It is noteworthy that in Fig. 16, the maps with $\Theta_n = 1$ represent the pull-based model, while the push-and-pull model converges to the push-based model with $\Theta_n \geq 3$ and $\Theta_n \geq 2$ in Figs. 16(a) and 16(b), respectively.

We compute an optimal threshold for each element of the state as a decision criterion, given the values of the other elements. Let us consider Ω_α and Ω_β as the decision criteria at the SA and AA, respectively. To derive the optimal decision α_n^* at the n -th slot, there are two alternative ways to define the criterion:

- Ω_α is a threshold for the index of the update's importance rank, i.e., $i \in \mathcal{I}$, for $v_n = \nu_i, \forall \nu_i \in \mathcal{V}$, given the E-ACK, i.e., \hat{E}_n . Thus, we have

$$\alpha_n^* = \mathbb{1}\{i \geq \Omega_\alpha(\hat{E}_n) | \hat{E}_n\}. \quad (33)$$

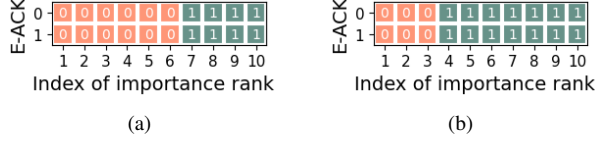


Fig. 15. Lookup maps for decision-making at the SA with (a) $C_{\alpha,\max} = 0.06$ and (b) $C_{\alpha,\max} = 0.08$.

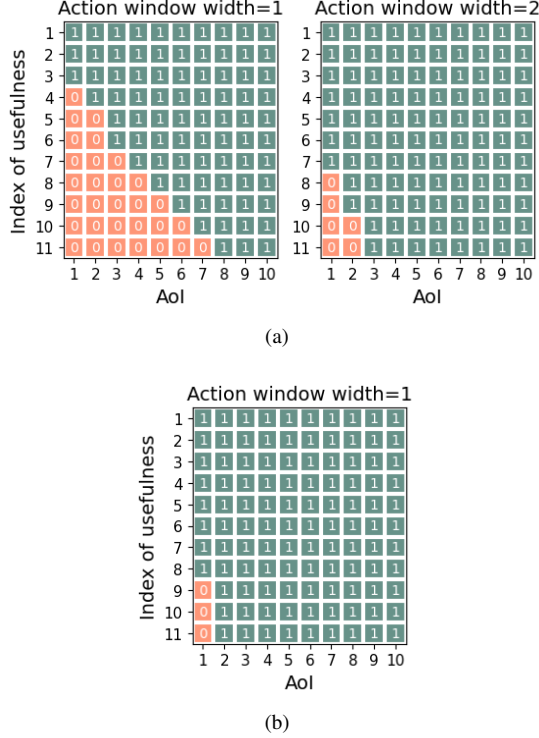


Fig. 16. Lookup maps for decision-making at the AA with (a) $C_{\beta,\max} = 0.06$ and (b) $C_{\beta,\max} = 0.08$.

■ Ω_α shows a threshold for the E-ACK given the importance rank of the update, such that

$$\alpha_n^* = \mathbb{1}\{\hat{E}_n \geq \Omega_\alpha(v_n) \mid v_n\}. \quad (34)$$

For instance in Fig. 15(a), $\Omega_\alpha(\hat{E}_n = 0) = \Omega_\alpha(\hat{E}_n = 1) = 7$, $\forall n$. Also, $\Omega_\alpha(v_n = v_i) > 1$ for $i \leq 6$, while $\Omega_\alpha(v_n = v_i) = 0$ for $i > 6$. Applying the same approach, we find the decision criterion Ω_β to obtain β_n^* from three different viewpoints.

VII. CONCLUSION

We investigated decision-making for enhancing the effectiveness of updates communicated in the end-to-end status update system based on a push-and-pull communication model. To this end, we considered that the SA observes an information source and generates updates, and its transmission controller decides whether to send them to the AA or not. On the other hand, the AA is responsible for acting based on the updates that are successfully received and the queries initiated to accomplish a subscribed goal at the endpoint. After defining the GoE metric, we formulated the decision problem of finding optimal effect-aware policies that maximize the expected discounted sum of the update's effectiveness subject

to cost constraints. Using the dual problem, we cast it to a CMDP solved separately for each agent based on different model components and proposed an iterative algorithm to obtain the decision policies. Our results established that the push-and-pull model, on average, outperforms the push- and pull-based models in terms of energy efficiency and effectiveness, respectively. Furthermore, effect-aware policies at both agents significantly enhance the effectiveness of updates with a considerable difference in comparison to that of periodic and probabilistic effect-agnostic policies used at either or both agents. Finally, we proposed a threshold-based decision policy complemented by a tailored lookup map for each agent that employs effect-aware policies. Future works could explore multi-SA scenarios with varying sensing abilities and the realizations' time-dependent importance.

REFERENCES

- [1] P. Agheli, N. Pappas, P. Popovski, and M. Kountouris, "Effective communication: When to pull updates?" in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2024, pp. 183–188.
- [2] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Commun. Mag.*, vol. 59, no. 6, pp. 96–102, 2021.
- [3] P. Popovski, O. Simeone, F. Boccardi, D. Gündüz, and O. Sahin, "Semantic-effectiveness filtering and control for post-5G wireless connectivity," *J. Indian Inst. Sci.*, vol. 100, no. 2, pp. 435–443, 2020.
- [4] B. Yin, S. Zhang, Y. Cheng, L. X. Cai, Z. Jiang, S. Zhou, and Z. Niu, "Only those requested count: Proactive scheduling policies for minimizing effective age-of-information," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2019, pp. 109–117.
- [5] F. Li, Y. Sang, Z. Liu, B. Li, H. Wu, and B. Ji, "Waiting but not aging: Optimizing information freshness under the pull model," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 465–478, 2021.
- [6] J. Holm, A. E. Kalør, F. Chiariotti, B. Soret, S. K. Jensen, T. B. Pedersen, and P. Popovski, "Freshness on demand: Optimizing age of information for the query process," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2021.
- [7] O. T. Yavascan, E. T. Ceran, Z. Cakir, E. Uysal, and O. Kaya, "When to pull data for minimum age penalty," in *Proc. IEEE Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, 2021, pp. 1–8.
- [8] F. Chiariotti, J. Holm, A. E. Kalør, B. Soret, S. K. Jensen, T. B. Pedersen, and P. Popovski, "Query age of information: Freshness in pull-based communication," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1606–1622, 2022.
- [9] M. E. Ildiz, O. T. Yavascan, E. Uysal, and O. T. Kartal, "Query age of information: Optimizing AoI at the right time," in *Proc. IEEE Int. Sym. Inf. Theory (ISIT)*, 2022, pp. 144–149.
- [10] M. Hatami, M. Leinonen, and M. Codreanu, "AoI minimization in status update control with energy harvesting sensors," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8335–8351, 2021.
- [11] A. Kosta, N. Pappas, and V. Angelakis, "Age of information: A new concept, metric, and tool," *Found. Trends Netw.*, vol. 12, no. 3, 2017.
- [12] 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, 2021.
- [13] N. Pappas, M. A. Abd-Elmagid, B. Zhou, W. Saad, and H. S. Dhillon, *Age of Information: Foundations and Applications*. Cambr. Univ. Press, 2023.
- [14] M. Hatami, M. Leinonen, Z. Chen, N. Pappas, and M. Codreanu, "On-demand AoI minimization in resource-constrained cache-enabled IoT networks with energy harvesting sensors," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7446–7463, 2022.
- [15] G. Stamatakis, N. Pappas, A. Fragkiadakis, N. Petroulakis, and A. Tragantitis, "Semantics-aware active fault detection in status updating systems," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 1182–1196, 2024.
- [16] A. Maatouk, M. Assaad, and A. Ephremides, "The age of incorrect information: An enabler of semantics-empowered communication," *IEEE Trans. Wireless Commun.*, vol. 22, no. 4, pp. 2621–2635, 2022.
- [17] R. L. Stratonovich, "On the value of information," *Izv. USSR Acad. Sci. Tech. Cybern.*, no. 5, 1965.
- [18] R. A. Howard, "Information value theory," *IEEE Trans. Syst. Cybern.*, vol. 2, no. 1, 1966.

- [19] P. Agheli, N. Pappas, and M. Kountouris, "Semantic filtering and source coding in distributed wireless monitoring systems," *IEEE Trans. Commun.*, vol. 72, no. 6, pp. 3290–3304, 2024.
- [20] N. Pappas and M. Kountouris, "Goal-oriented communication for real-time tracking in autonomous systems," in *Proc. IEEE Int. Conf. Autonomous Syst. (ICAS)*, 2021, pp. 1–5.
- [21] J. Luo and N. Pappas, "Semantic-aware remote estimation of multiple Markov sources under constraints," *IEEE Trans. Commun. (Early Access)*, 2025.
- [22] D. Bertsekas, *Dynamic programming and optimal control*. Athena Sci., 2007, vol. 2.
- [23] E. Altman, *Constrained Markov decision processes*. CRC press, 1999.
- [24] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. J. Wiley & Sons, 2014.
- [25] R. Bellman, "On the theory of dynamic programming," *Proc. Natl. Acad. Sci.*, vol. 38, no. 8, pp. 716–719, 1952.
- [26] G. Wood, *Bisection global optimization methods*, C. A. Floudas and P. M. Pardalos, Eds. Springer Sci. Bus. Media, 2009.
- [27] F. J. Beutler and K. W. Ross, "Optimal policies for controlled Markov chains with a constraint," *J. Math. Anal. Appl.*, vol. 112, no. 1, pp. 236–252, 1985.
- [28] P. Tseng, "Solving H -horizon, stationary markov decision problems in time proportional to $\log(H)$," *Oper. Res. Lett.*, vol. 9, no. 5, pp. 287–297, 1990.
- [29] M. L. Littman, T. L. Dean, and L. P. Kaelbling, "On the complexity of solving Markov decision problems," *Proc. Conf. Uncertainty Artif. Intell. (UAI)*, pp. 394–402, 1995.
- [30] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.