Ensuring a Semantically Effective IoT Network Through Blockchain with Delayed Feedback

Omer Gencay, Alperen Balci, Elif Tugce Ceran, and Elif Uysal
Communication Networks Research Group (CNG),
Electrical and Electronics Eng. Dept, METU, Ankara, Turkey

***Abstract*—In this paper, we propose a delay-tolerant blockchain-enabled low-power wide area network. We consider the scenario of a set of information sources (e.g. sensors, IoT nodes) equipped with transceivers, accessing a common gateway in order to send sampled data of a local process to a group of remote destinations. Each source has an information penalty function, and the gateway earns a reward according to the result of this function. We show that our policy maximizes the reward of the gateway earned from the blockchain in a delayed reward situation.**

***Index Terms*—Delay, Delay-tolerant-networks, Blockchain, Semantic Communication**

I. Introduction

The Internet of Things (IoT) ecosystem facilitates the control and monitoring of various data on a single dashboard. The ease of collecting data from remote sources, however, also leads to large amounts of possibly irrelevant data to store and large data tables that must be searched to extract meaningful data. Consider, for example, a temperature sensor that is sampled every 10 seconds, only to provide the same temperature. Sending the same data over and over again is not only a waste of resources but also a waste of storage space. Moreover, considering that most IoT nodes transmit their data over a shared channel (e.g., a wireless link to an access point), unnecessarily frequent transmissions not only increase the load but also lead to premature exceeding of network capacities. Congestion at the network and transport layers also constitutes a bottleneck. Therefore, for the scalability of IoT both in terms of communication resources and in terms of data storage and retrieval, judicious selection of the data to be transmitted is important.

However, this requires cross-layer operation at the application and MAC layers, where the data generation process and the scheduling of transmissions are coordinated. For example, when there are packet drops at the MAC layer due to high load in the network, it will be meaningless for sources to generate new samples at faster rates, only to have them delayed in the transmitter queues. Hence, in contrast to traditional networking models with exogenous data arrival processes at transmitters, one needs to adopt a "generate-at-will" model where nodes can decide to generate fresh data samples at a rate the network can support.

This work was supported in part by TUBITAK grant 22AG019.

In recent literature, there have been various efforts to extend communication and networking models to a "semantic communication" model [1], [2], [3]. In line with [2], our use of the term semantic refers not to the meaning but to the value of the data with respect to a computation to be performed at the destination. In accordance, examples of metrics measuring semantic value are Age of Incorrect Information (AoII) [4], Value of Information [5], and Peak Age of Information [6].

As the scale of networked systems grow, the network systems tend to become more heterogeneous and produce more privacy-sensitive data. Distributed technologies such as blockchain can be used for secure and encrypted transmissions and logging [3]. Blockchain is a distributed ledger technology that is used to record transactions across many computers so that the records cannot be altered retroactively without the alteration of all subsequent blocks and the collusion of the network. For example, the Helium Network is a wide-area networking system that uses protocol tokens for data transmissions using blockchain [14]. In the Helium system, there is a system that rewards the gateway that sends the data, and its security is ensured by blockchain.

This paper proposes to repurpose the blockchain to provide security and a mechanism to gauge and reward the semantic effectiveness of the data sent by the sources. By rewarding sources that send useful data and penalizing those who send data of no value, unnecessary transmissions are reduced, and the allocation of wireless channel resources to sources sending useful data is maximized.

Hence, the proposed architecture can be described as a blockchain-aided medium access resource allocation scheme. One of the technical issues to be handled is the intrinsic delay in the blockchain computation, which can impact the MAC performance. We construct a multi-armed bandit model that allows analyzing and circumventing the effects of this unavoidable delay. We show that we can obtain performance close to a delay-free environment.

The rest of the paper is organized as follows. In Section II, we describe the system components and define the problem formulation. In Section III, we propose delay-tolerant scheduling policies to maximize the average expected semantic reward. In Section IV, we compare the proposed policy with base-line policies and evaluate the effect of blockchain delay. Finally, in

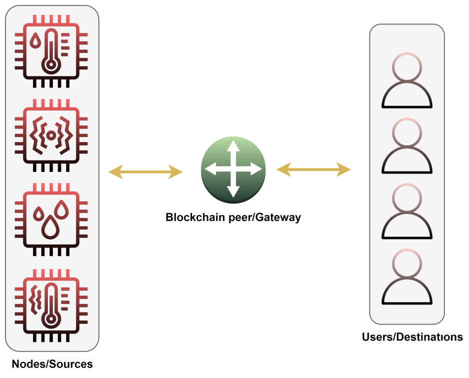

Fig. 1. System Model for Blockchain-Enabled LP-WAN Network

Section V, we conclude the paper.

II. System Design

We consider a blockchain-enabled network architecture with Low-power wide-area network (LP-WAN) access network scenario where the sources send update-based data to subscribers through the gateways and blockchain components see Figure 1. A number of sources are distributed in a large area and connected to a common gateway. A gateway manages the connection between destinations and sources as a member of the blockchain. It aims to earn a reward by securing and providing informative data to the users in the network. We will focus on a single gateway for the rest of the paper, but the model can be extended to include multiple (competing) gateways.

We will refer to the sources as sensors and the remote destinations as *users*. The users are interested in the data produced by the sources to exploit the data in various applications. We assume that the communication is pull-based, whereby the gateway pulls status updates from sensors by applying a proper scheduling algorithm. The blockchain is used for two main purposes: validation and reward, but it causes significant end-to-end delays. Since blockchain has an overhead, the system model does not cover real-time applications.

Gateways identified as untrustworthy are labeled with a low reputation parameter. Based on this, the gateway either keeps earning rewards or faces a temporary ban. Figure 2 illustrates the validation and rewarding processes between gateways. Validator gateways have secret keys for threshold values and information penalty functions to decrypt and validate the transactions. Transactions can be considered authorization of the gateway to transfer the updated data from sources to the network server. If a transaction is successful, updated data is sent to the network server; otherwise, it's not transferred.

The fee that the gateway earns to continue serving the network is referred to as a reward. The gateway earns a reward if it pulls data from the sources that have valuable data. The value of the data is measured by a semantic metric (i.e. AoI, AoII, PAoI).

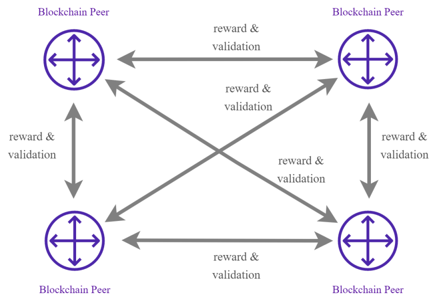

Fig. 2. Validation and Rewarding Processes Between Gateways

There are N sensors where each sensor $i \in \{1, ..., N\}$ has a reward function $R_i(t)$ as follows:

$$R_i(t) = \begin{cases} 1, & \text{if semantic metric is satisfied} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Let $p_i \in (0, 1)$ be each source's independent, identically distributed reward probability. It is assumed that p_i is not known apriori by the gateway.

A. Problem Definition

Let policy $\pi \in \Pi$ denote the scheduling policy employed by the gateway. The maximization can be shown in (2).

$$\max_{\pi \in \Pi} \quad \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} \mathbb{E}[R_i(t)] \tag{2}$$

$$\text{s.t.} \quad \textstyle\sum_{i=1}^{N} u_i(t) \leq k, \forall t, \tag{3}$$

where k denotes the maximum number of sources that can be utilized at each t. Gateway claims a reward for each source after scheduling it. This transmission may receive a reward after being processed by the blockchain network, and the gateway will learn about the reward with a delay. Since the rewards are independent and identically distributed, this problem is considered as a multi-armed bandit (MAB) problem with delayed feedback.

III. The Delay Tolerant Scheduling Policy

In this section, we investigate the proposed delay-tolerant scheduling policy. We assumed the reward probability of each arm is i.i.d. and did not change with the time during the operation. The reward of selected arms is obtained after some delay. Due to the blockchain features, delay follows a Poisson distribution with a fixed mean.

The maximization problem with unknown reward probabilities can be considered a stochastic MAB problem where each source represents an arm, and the gateway represents the agent. To solve this problem, we will define some parameters. Each source has a mean reward denoted by $\mathbb{E}[R_i(t)]$. The proposed policy tries to guess the number of k sources with a maximum mean reward from the sources set.

To choose arms or decide when to exploit, two well-known techniques are Upper Confidence Bound (UCB) and Thompson Sampling (TS) [8], [11].

A. Upper Confidence Bound

Since the gateway does not have a priori knowledge of any source i, we exploit the upper confidence bound (UCB) to estimate the empirical reward as shown in (4).

$$\hat{\mu}_{i,t} = \frac{\sum_{\tau=1}^{t} (R_i(\tau))}{n_{i,t}} \tag{4}$$

where $n_{i,t}$ indicates the number of times source i is scheduled up to time t and $R_i(\tau)$ is the gained reward at round τ. The calculation of the UCB value for each source is given in (5).

$$UCB_{i,t} := \hat{\mu}_{i,t} + \sqrt{\frac{2\ln(t)}{n_{i,t}}} \tag{5}$$

Using the term $\sqrt{\frac{2\ln(t)}{n_{i,t}}}$ avoids exploiting the same source when it gives high $\hat{\mu}_{i,t}$ value and represents the uncertainty. As $n_{i,t}$ increases, uncertainty decreases and the system becomes more certain about the estimated value of the arm. Otherwise, the corresponding arm becomes a candidate for exploration. Since the empirical reward needs initial values, all sources are scheduled once to observe the reward. The UCB bound increases for every time slot t, preventing the gateway from always scheduling the same source. In each round, the arm with the highest UCB is selected for pulling.

B. Thomson Sampling

The TS algorithm uses a Bayesian approach to model the reward distribution of the sources. The sources give reward with Bernoulli distribution, and p_i is the success probability for source i. To model the prior distribution of p_i, Beta distribution [12] is adopted and $\hat{\nu}_i$ is the prior estimate of p_i, and Γ is the Gamma function. (6) tries to converge the true p_i with the help of Beta distribution.

$$f(\hat{\nu}_i(t); \alpha_i, \beta_i) = \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \hat{\nu}_i(t)^{\alpha_i - 1}(1 - \hat{\nu}_i(t))^{\beta_i - 1} \tag{6}$$

It is a function of α_i and β_i representing the success and failure counts of source i, where $\alpha_i > 1, \beta_i > 1$. At each iteration, the posterior Beta distribution of each source is calculated, and the source that gives the max value for the next round will be chosen. Simply, the posterior can be $Beta(\alpha_i+1, \beta_i)$ or $Beta(\alpha_i, \beta_i+1)$. This way, TS balances the exploration and exploitation processes based on posterior Beta distributions. Each successful reward will increase α_i, and each unsuccessful reward will increase β. The TS initially assumes $\alpha_i = 1, \beta_i = 1$ to make the selection probability of each source uniform in the first place.

C. Delay Tolerant UCB and TS with Reward Buffers

While UCB and TS algorithms demonstrate effective convergence in scenarios with perfect feedback, their performance experiences a decline when applied to systems with delayed feedback, such as a blockchain network. In the paper [9], the algorithm for delayed feedback systems considers fixed feedback delay. The extended algorithm is given in [10] as a black-box algorithm. In [10], there is a black-box algorithm that aims to increase the performance of the MAB algorithms even if there is a delay in the feedback. An arm is considered ”free” if it has already received reward information after prior prediction and ”busy” if it awaits feedback and arms are selected from the ”free” ones. Using Thomson sampling and Upper Confidence Bound as a selection algorithm, the proposed algorithms are given as Algorithm 1 and Algorithm 2. Unlike the other two papers [9] [10], two buffers are used to store the delays and reward histories. This means the same node can be scheduled even if its last feedback has not been realized yet.

In this paper, we adopt the black-box algorithm by adding one more buffer for storing both delay and reward for each pull for each arm, and sampling rewards uniformly from the finite size reward buffer instead of applying the first in first out (FIFO) scheme. $\hat{R}_i(\tau)$ represents empirical reward of each source i.

Algorithm 1 UCB algorithm with reward buffer

```
Set delay value.
for each node do
    Create an empty reward buffer.
    Create a delay buffer and set all delays to -1.
    Create a selection counter $n_i(t)$ for each node
    and set all counters to 0.
    Create an average reward $\hat{\mu}_i(t)$ for each node
    and set all average rewards to 0.
end for
while # of node selection lower than mean delay do
    Select k node for each round from the beginning
    to end, respectively
    Update delay buffer $D[i] = d$
end while
for each round $\tau$ do
    Sample reward $\hat{R}_i(\tau)$ uniformly from reward buffer
    Select $k$ nodes that maximizes $\hat{\mu}_i(t) + \sqrt{\frac{2ln(t)}{n_i(t)}}$
    Update $n_i(t)$ and $\hat{\mu}_i(t)$ for selected nodes
    if there is any node that has 0 in delay buffer then
        Delete oldest reward
        Add new reward $\hat{R}_i(t)$ to the reward buffer
        Update empirical reward
        Update delay buffer to -1
    end if
    Update delay buffer
end for
```

In algorithm 1, after the delay value is set, empty reward buffers, delay buffers, selection buffers, and, average reward for each node are initialized and set to 0. In order to distinguish nodes whose reward time has arrived, the delay buffer has been assigned as -1 instead of 0. For all nodes, the empirical reward is 0 at the beginning. Each node is selected at least a number of delay rounds to give an initial value to each node's empirical reward. The following process is a simple UCB algorithm with a few differences: Select k nodes that maximize equation 5. Sample reward estimate $\hat{R}_i(\tau)$ uniformly from reward buffer of i. Update $n_i(t)$ and $\hat{\mu_i}(t)$ for the selected nodes. If there is any node whose reward time has arrived, the new reward is added to the FIFO reward buffer, and the delay buffer of the node is updated to -1.

Algorithm 2 TS algorithm with reward buffer

```
for each node do
    Create an empty reward buffer.
    Create a delay buffer and set all delays to -1.
    Set $\alpha_i$ and $\beta_i$ values to 1 for all $i \in \{1, \ldots, N\}$.
end for
while # of node selection lower than mean delay do
    Select k node for each round, respectively
    Update delay buffer
end while
for each round do
    Sample reward $\hat{R}_i(t)$ uniformly from reward buffer
    If $\hat{R}_i(t) = 1$, then $\alpha_i = \alpha_i + 1$, else $\beta_i = \beta_i + 1$
    Select $k$ nodes that maximizes $\hat{\nu_i}(t)$
    Update delay buffer by decreasing all delays 1 round
    if there is any node that has 0 in delay buffer then
        Delete oldest reward
        sample $\hat{\nu_i}(t)$ from the $Beta(\alpha_i + 1, \beta_i + 1)$
        distribution.
        Add new reward to the reward buffer
        Update delay buffer to -1
    end if
    Update delay buffer
end for
```

Thompson Sampling with reward buffer is implemented in algorithm 2. At the beginning, empty reward and delay buffers are created for each node. The values α and β are set to 1 for all nodes. Similar to algorithm 1, each node is scheduled at least delay times and delay buffer is updated. For each round, sample reward $\hat{R}_i(t) = 1$ from the reward buffer, update TS statistics $alpha$ and $beta$. Check if there is a reward arriving as the round passes. If a new reward is arrived, delete the oldest reward for the selected node. Sample $\hat{\nu_i}(t)$ from the $Beta(\alpha_i + 1, \beta_i + 1)$ distribution, add new reward to the reward buffer and set delay buffer to -1.

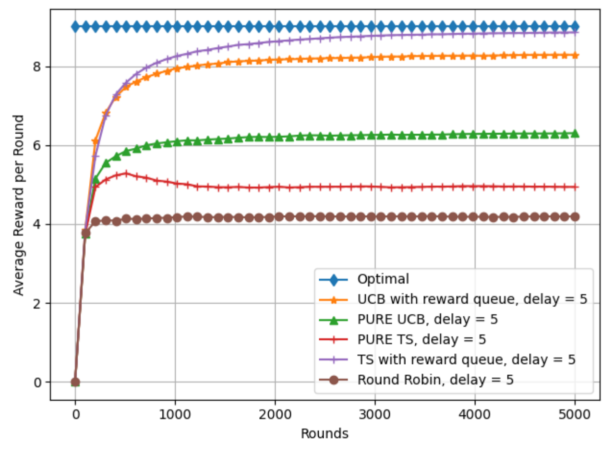

Fig. 3. Average reward per round vs. number of rounds

IV. Numerical Results and Discussion

In this section, we provide numerical results that put emphasis on the effects of the system dynamics on the expected mean reward. We evaluated the performance of pure UCB, UCB with reward buffer, pure TS, TS with reward buffer, round robin, and optimal algorithms. Round Robin is a scheduling algorithm that allocates resources in an equitable manner to a set of tasks, allowing each task to execute before moving on to the next. The Round Robin algorithm was implemented to conduct a comparative analysis with our proposed algorithms. The optimal algorithm epitomizes an exemplary decision-making framework wherein complete knowledge of the entire spectrum of system rewards is presupposed, enabling the systematic identification and enactment of the most advantageous choices. The reward buffer size is set to 10, and the delay buffer size is set to 20. For the results, the blockchain delay is considered in two cases: fixed delay and a Poisson distributed delay with a mean of 10 minutes [13].

In Figure 3, the cases that are using the reward buffer converge to the optimal case after the queue starts to be full. For the optimal case, it is considered that the nodes that have the biggest reward probabilities have been selected for all the rounds. The results show that the algorithms that have reward buffers are about to converge to the optimal case even if they have fixed delays.

Figure 4 shows the mean reward per round of a number of nodes. The number of nodes varies from 10 to 500, and the number of nodes to select (k) is fixed at 10. The UCB and TS algorithms try to learn the expected rewards of the nodes over time and select the node with the highest expected reward. The figure shows that the UCB algorithm performs the best, followed by the TS algorithm.

The delay performances of the algorithms are shown in Figure 5. With no delay, UCB and TS algorithms provide an almost optimal reward per round. As the delay increases, the

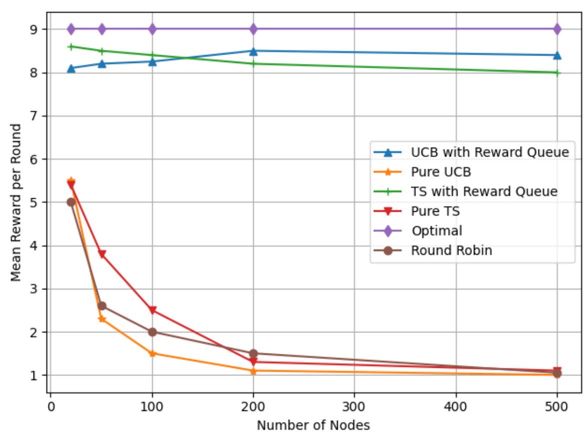

Fig. 4. Mean reward per round vs. number of arms with k = 10

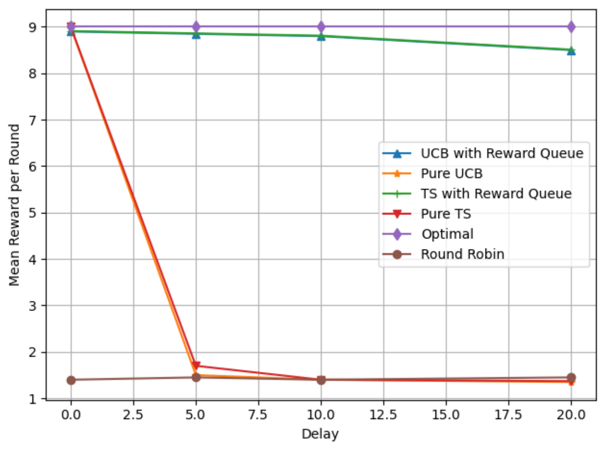

Fig. 5. Mean reward per round vs. delay with k = 10, N = 100

proposed delay-tolerant UCB and TS algorithms still perform very close to optimal. However, the benchmark algorithms that do not have reward and delay queues drastically decrease the reward of the round-robin policy.

V. Conclusion

In this paper, we study a delay-tolerant blockchain network. The introduced reward buffer algorithm converges to the optimal case even in the presence of a delay in the blockchain network. The exploration and exploitation processes are modeled as an MAB problem. The delay introduced by the blockchain is investigated for fixed delay or stochastic delay cases. Other semantic metrics for the network server and the blockchain can be considered for future work. Moreover, the work in this paper can be extended by addressing the energy consumption, overhead, and complexity challenges of the blockchain structure and exploiting the security, transparency, and decentralization advantages. In future research, the goal is to investigate the impact of variable reward delays within the advanced blockchain architecture on the system by considering addressed aspects.

Acknowledgement

We thank Onur Kaya, Alper Yegin, and Mutlu Ahmetoglu for their useful discussions.

References

[1] E. Calvanese Strinati and S. Barbarossa, “6G networks: Beyond Shannon towards semantic and goal-oriented communications,” Computer Networks, vol. 190, p. 107930, 2021.
[2] E. Uysal et al., “Semantic Communications in Networked Systems: A Data Significance Perspective,” IEEE Network, vol. 36, no. 4, pp. 233–240, Jul. 2022.
[3] L. Hang and D.-H. Kim, “Design and Implementation of an Integrated IoT Blockchain Platform for Sensing Data Integrity,” Sensors, vol. 19, no. 10, p. 2228, May 2019,
[4] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, “The Age of Incorrect Information: A New Performance Metric for Status Updates,” IEEE/ACM Transactions on Networking, vol. 28, no. 5, pp. 2215–2228, Oct. 2020.
[5] R. A. Howard, ”Information Value Theory,” in IEEE Trans. on Sys. Sci. and Cybernetics, vol. 2, no. 1, pp. 22-26, Aug. 1966.
[6] M. Costa, M. Codreanu, and A. Ephremides, “Age of information with packet management,” 2014 IEEE International Symposium on Information Theory, 2014.
[7] W. R. Thompson, “On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples,” Biometrika, vol. 25, no. 3/4, p. 285, Dec. 1933.
[8] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, “The Nonstochastic Multiarmed Bandit Problem,” SIAM Journal on Computing, vol. 32, no. 1, pp. 48–77, Jan. 2002.
[9] M. J. Weinberger and E. Ordentlich, “On delayed prediction of individual sequences,” IEEE Transactions on Information Theory, vol. 48, no. 7, pp. 1959–1976, Jul. 2002.
[10] P. Joulani, A. Gyorgy, C. Szepesvari. Online Learning under Delayed Feedback. International Conference on Machine Learning, Atlanta, Georgia, USA, 2013.
[11] W. R. Thompson, “On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples,” Biometrika, vol. 25, no. 3/4, p. 285, Dec. 1933.
[12] N. L. Johnson, Continuous univariate distributions / 2. New York: Wiley, 1995.
[13] Ghassan O. Karame, Elli Androulaki, and Srdjan Capkun. Double-spending fast payments in bitcoin. In Proceedings of the ACM Conference on Computer and Comms. Security. Association for Computing Machinery, New York, NY, USA, 906–917, 2012.
[14] A. Haleem, A. Allen, A. Thompson, M. Nijdam, R. Garg. 2012. Helium, A Decentralized Wireless Network. Helium Systems, Inc. Release 0.4.2 (2018-11-14).