Optimizing Age of Information on Real-Life TCP/IP Connections through Reinforcement Learning

Egemen Sert*, Canberk Sönmez†, Sajjad Baghaee‡, Elif Uysal-Biyikoglu§
*†‡§Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey
*egemen.sert@metu.edu.tr , †canberk.sonmez@metu.edu.tr, ‡sajjad@baghaee.com, §uelif@metu.edu.tr

***Abstract*—Age of Information (AoI) has emerged as a performance metric capturing the freshness of data for status-update based applications (*e.g.*, remote monitoring) as a more suitable alternative to classical network performance indicators such as throughput or delay. Optimizing AoI often requires distinctly novel and sometimes counter-intuitive networking policies that adapt the rate of update transmissions to the randomness in network resources. However, almost all previous work on AoI to data has been theoretical, assuming idealized networking models, and known delay and service time distributions. It is difficult to obtain these statistics and optimize for them in a real-life network as there are many interacting phenomena in different networking layers (*e.g.*, consider an end-to-end IoT application running over the Internet). With this work we introduce a deep reinforcement learning-based approach that can learn to minimize the AoI with no prior assumptions about network topology. After evaluating the learning model on an emulated network, we have shown that the method can be scaled up to any realistic network with unknown delay distribution.**

***Keywords*—Age of Information (AoI), Reinforcement learning (RL), Deep Q Networks, AoI on TCP/IP.**

I. Introduction

In status update-based applications (*e.g.*, machine-type communication such as industrial manufacturing, telerobotics, IoT for smart cities, environmental monitoring, social network applications) the key performance metric is the freshness of data. A majority of these applications transmits updated contents sufficiently frequently to a destination for status monitoring. It means that the latest status of these contents is more interesting in destination side. For instance, in asset or individual tracking applications [22], [23], the most recent information provides client a way to estimate the location of the object. Therefore, the "freshness" of data can be used as a performance indicator parameter. For quantifying the freshness of information, the concept of AoI was introduced in [1]. The metric of AoI is defined as the time elapsed since the newest data was generated.

In recent years, reducing the AoI and keeping the information fresh, becomes one of the fascinating research topics among the researchers. Queuing model and packet management scheme are considered in [2] to minimize age. Age-optimal generation of update packets of single-hop networks was analyzed in [3], [4]. In [5] for multiserver single-hop systems when service times are exponential, it is shown that a preemptive Last Generated First Served (LGFS) policy can optimize the age, throughput, and delay simultaneously, while the packet generation times, arrival times, and queue buffer size are arbitrary. The study of [6] shows in multihop networks when the packet transmission time over the network links are exponentially distributed, preemptive Last Generated First Served (LGFS) policy provides the smaller age-of-information at all nodes in a stochastic ordering sense. [7], [8] are focused on AoI for push-based communication systems where sensors decide when to send an update to the destination.The sub-optimality of the throughput and delay optimal update policies with respect to age under transmission scheduling setup is shown in [9]. A real-time sampling problem of the Wiener process is solved in [10], where the authors showed optimal sampling problem can reduce to an age-of-information optimization problem. Authors in [11] minimizing the AoI for a sensor network under energy harvesting constraints with unit capacity under a memoryless energy arrival process. Furthermore, in an energy harvesting transmitter with any battery capacity, the AoI minimization is shown in [12] while authors explicitly characterized the threshold structure.

To the best of our knowledge most of the studies of the AoI are related to theoretical approaches and [13] is the only work that utilizes RL on minimizing the average AoI. They formulated the AoI minimization problem as a constrained Markov Decision Process and solved it by using Value Iteration and SARSA algorithms, on discrete state space (ages defined as integers). In our study, a deep RL based algorithm is proposed to present a realization and provide a practical solution to minimize AoI in continuous state-space.

In this study, a real-world networking protocol, TCP/IP, is used with the AoI concept for communication. The experimentation of learning algorithm is conducted with a network emulator called CORE [14] on a client-server pair. This emulator is also used for testing theoretical work on AoI concept by [15].

For the first time, In this study, the non-monotonic relation between AoI and sampling rate is observed for a channel with buffer over TCP/IP protocol in emulation (Fig. 1 & 2). This motivated us to start the investigation of this phenomenon.

II. Problem Formulation

Status age is defined as $\Delta(t) = t - U(t)$, where it is function of time t. $U(t)$ is the generation time (i.e. timestamp) of the freshness data that the destination has received by time t. In other words, age ($\Delta(t)$) is the time elapsed since the newest data was generated. A sample variation of age $\Delta(t)$ with the sawtooth pattern, for a source as a function of time t, at the destination is shown in Fig.3, where at time $t = 0$ observation has begun. It is assumed the queue of the destination is empty, while the initial age is Δ_0 (i.e. $\Delta(0) = \Delta_0$). The source status updates are generated at time t_1, t_1, $\cdots$,t_n and received at time $U(t_1), U(t_2), \cdots, U(t_n)$ respectively. In the absence of any updates, the source status age at the destination

978-1-5386-1501-0/18/$31.00

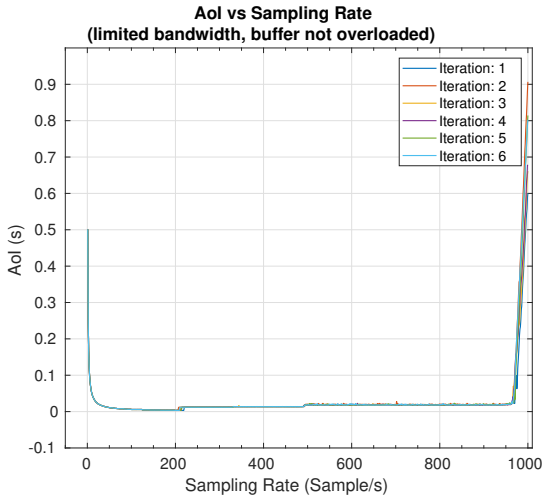

Fig. 1: AoI vs. sampling rate, from 0 to 1000 samples/s (6 iterations)

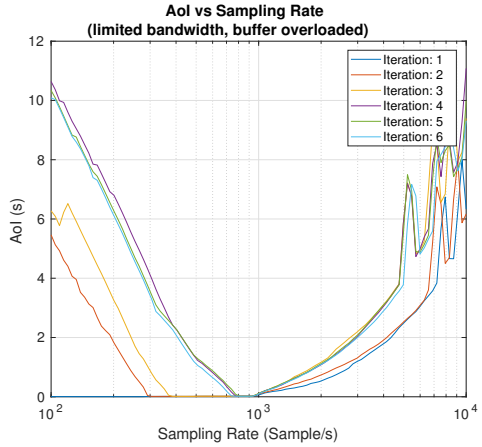

Fig. 2: AoI vs. sampling rate, from 100 to 10000 samples/s (6 iterations)

increases linearly in time and is decreases just after an update is received. AoI is defined by (1) and shown by the area under the age graph (Fig.3), where normalized by time T.

$$\overline{\Delta} = \frac{1}{T}\int_0^T \Delta(t)dt \tag{1}$$

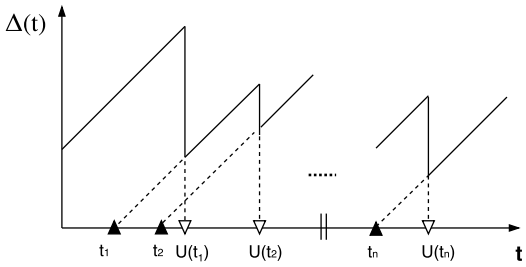

Fig. 3: Sample path of the age process $\Delta(t)$

In this study to minimize the average AoI in a network with unknown delay distribution, a Markov Decision Process (MDP) has been formulated where the transition probabilities are unknown. The formulation of the MDP is as follows:

- $s_t = \Delta_t$, age at time t is the state at time t
- $a_t = \{p, r\}$, the action at time t (pause and resume)

Due to the Markov Property, it can be stated that the next state is merely dependent on the current action and the state, with transition probability $p(s_{t+1}|s_t, a_t)$. Since the transition probabilities are not known, a formulation is required to learn the underlying transitions. Hence, to solve the problem, RL is utilized, where the goal of the RL is to find the optimal action that maximizes the expected cumulative reward $r(s, a)$ over the trajectory distribution $p_\theta(\tau)$ as in (2). The trajectory is defined as the state-action pair $\tau = (s, a)$.

$$\theta^* = \arg\max_\theta \ \mathbb{E}_{\tau\sim p_\theta(\tau)}\Big[\Sigma_t r(s_t, a_t)\Big] \tag{2}$$

To orient the problem towards RL, it's assumed to have infinite horizon where the optimization process is potentially infinite. The goal of the infinite horizon RL is given in (3).

$$\theta^* = \arg\max_\theta \Big(\limsup_{T\to\infty} \ \frac{1}{T}\ \mathbb{E}_{\tau\sim p_\theta(\tau)}\Big[\Sigma_{t=1}^T r(s_t, a_t)\Big]\Big) \tag{3}$$

The Reinforcement Learning literature evolves around the reward concept where the environment informs the agent how well the taken action is given the current state; however, in the AoI concept, the goal is to minimize the age to have fresh data. Therefore, (3) is modified to comply with the goal by introducing a cost function as $c(s_t, a_t) = -r(s_t, a_t) = -\Delta_t$. Consequently, the task of the paper is composed in the (4)

$$\theta^* = \arg\min_\theta \Big(\limsup_{T\to\infty} \ \frac{1}{T}\ \mathbb{E}_{(s_t,a_t)\sim p_\theta(s_t,a_t)}\Big[\Sigma_{t=1}^T \Delta_t\Big]\Big) \tag{4}$$

To solve the given problem (4), the popular algorithm, deep Q network (DQN), introduced by [16] is employed. The algorithm tries to estimate the expected reward (Q-Value) of each state's possible actions, using neural networks. Albeit, since the goal is to minimize the cost rather than maximizing the reward, the algorithm is modified such that it minimizes the introduced cost function $c(s_t, a_t) = 1 - \exp\{-\Delta_t\}$, thus the Algorithm 1 is obtained. In the algorithm, a neural network is utilized to map states to each possible action's Q-value, as conducted in [16]. The Q-values represent the vector of expected costs if any of the actions a is taken in state s; moreover, $\arg\min_a \ \hat{Q}_\phi(s)$ represents the action that minimizes the expected cost $\mathbb{E}[c(s, a)]$ at state s. Although taking the action with the minimum Q-value at each step, the agent is forced to explore by utilizing the ϵ_{greedy} approach [17]. Hence at each time step the action is taken as the one with the minimum Q-value with probability $1-\epsilon$ or a random action with probability ϵ. It is known that Deep Q-Learning may diverge if not the conditions, as shown in [18], are satisfied, thus stabilizing approaches such as experience replay [19] and Double DQN [20] are introduced.

Algorithm 1 Double Deep Q Network with Experience Replay

$buffer$ ▷ empty stack of length 2000
$\eta \Leftarrow 10^{-3}$ ▷ learning rate
$\gamma \leftarrow 0.99$ ▷ discount factor
$\epsilon \leftarrow 1.0$ ▷ exploration factor
$\beta \leftarrow 0.995$ ▷ exploration decay factor
$\tau \leftarrow 0.125$ ▷ rolling average factor
$environment$ ▷ AoI simulation environment
$episode \leftarrow 0$
for each $episode$ **do**
  $s \leftarrow$ **reset**$(environment)$
  while not **done**$(environment)$ **do**
    $a \leftarrow \Gamma(\hat{Q}_\phi(s))$
    observe next state s' and cost $c = \Delta$
    $d \leftarrow$ **done**$(environment)$
    $transition \leftarrow (s, a, c, s', d)$
    push$(buffer, transition)$
    $\epsilon \leftarrow \beta * \epsilon$
    sample batch B from $buffer$
    for each (s, a, c, s', d) in B **do**
      $y \leftarrow c$
      if not d **then**
        $y \leftarrow c + \gamma \min_{a'} \hat{Q}_{\phi'}(s')$
      end if
      $\phi \leftarrow \phi - \eta \frac{d\hat{Q}_{\phi'}(s)}{d\phi}(\hat{Q}_{\phi'}(s) - y)$
    end for
  end while
  $\phi \leftarrow \tau\phi + (1 - \tau)\phi'$
end for

III. Emulation Testbed

In this study, CORE, an open-source network emulator, is used on a Linux machine and the emulation is constructed by its GUI. When the emulation is started CORE creates virtual nodes, each sharing the same OS resources except networking. Each node has its own network interface cards. Once the network topology is generated, commands can be executed in each node. The topology is shown in Fig. 4.

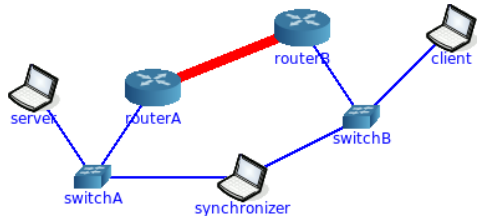

Fig. 4: CORE networking topology for experimentation

In this topology, the path delay of normal distribution with adjustable mean and variance is set for the direct link between routerA and routerB. For the remaining links, there is no delay, and for all links bandwidths are unlimited. The routing configuration is done such that the communication between server and client can occur only over one path, which involves routerA and routerB.The investigated algorithm is performed on the server and client nodes. Time synchronization between the server and client is important for age measurements. The synchronizer node is used for this purpose. Since the links between the synchronizer, and the server and client have no delays, time synchronization is done accurately.The client has three roles: generating samples containing timestamps at a fixed rate, sending samples to the server, and responding to pause and resume commands from the server. The pause command causes the client to stop generating new samples and the resume command makes the client continue sample generation.

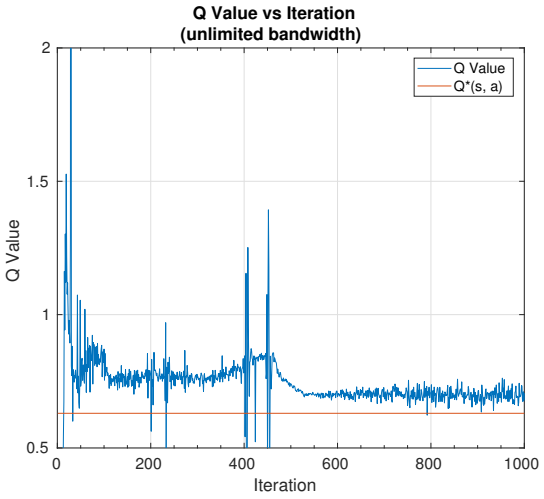

Fig. 5: Q Value per 10 iterations

This study begins by exploring the case where the link has unlimited bandwidth. In this case, no queueing occurs. Therefore the aging is only due to the path delay and it's possible to transmit data at any sampling rate without loading the network buffer. Since the network buffer is not loaded, pause command cannot decrease the age, it can only increase it. Therefore, the ideal command that should be sent by the server is resume. The age is calculated at the server, by using the timestamps in the received samples. Once the age is calculated, the deep RL algorithm uses this value and decides a command to minimize the AoI. In this case each packet transmission is instantaneous as in [4] due to the unlimited bandwidth, therefore, an optimized algorithm should send the ideal command, which is resume.

IV. Experiments

Since, to our knowledge, this study is the first to apply deep RL to minimize the average AoI, the algorithm is tested on a set up where the optimal policy is known. To test the algorithm, a network with 1 ± 0.5s delay is emulated where the optimal policy is to resume at each time step.

$$Q(s, a) = c(s, a) + \gamma \min_{a'} Q(s', a') \tag{5}$$

In the experiment, the Q-Value for the *resume* action a should approach to $Q(s, a) \approx 1 - \exp\{-\Delta_{min\_age}\}$ by applying boundary conditions on the Bellman equation, (5), that is at the terminal state, $Q^*(s, a) = c(s, a) = 1 - \exp\{-\Delta_{min\_age}\}$. Due to the expected value, the delay is $\mathbb{E}[D] = 1$s. It is expected on the Q-Value for the resume action a to fall to $Q(s, a) = 1 - \exp\{-1\} \approx 0.63$.

To initiate the experiment, Double DQN with experience replay is constructed with two hidden layers of 24 units. The main network is updated per iteration and the target network is updated per 100 iterations. The double DQN with Experience Replay is trained for 10000 iterations and the Q-Value converged to the desired value after 5000 iterations where the optimal action of resuming at each time step is taken. The Q-Value vs. Iteration is illustrated in Fig. 5. The blue line represents the instantaneous Q-Value of taking

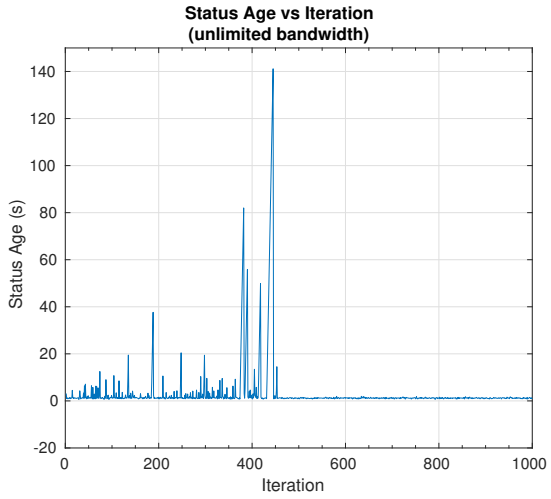

Fig. 6: Status age per 10 iterations

resume action on the given state and the red line shows the expected minimum value of the ideal Q-Value $Q^*(s, a)$. Since the task is to minimize the AoI, the agent's performance is monitored via plotting the status age versus each iteration, which is illustrated in Fig. 6.

The experiments corresponding to the Fig. 1 and 2 are conducted with a bandwidth limited link, and the same sampling rates are repeated in each iteration without any delay between the iterations. In this scenario, queueing occurs. Therefore, increasing the sampling rate doesn't always decrease the AoI. This behaviour can be observed in Fig. 1, where the AoI first decreases and then increases. In Fig. 2, since the sampling rate is very high at the end of each iteration, the loaded queue increases the age in the next iteration. In these experiments, observing the effect of excessive sampling rates on AoI is aimed. It is expected to observe the initial AoI when the same sampling rate is repeated, however, this is not the case. To conserve space, these observations are to be examined in the upcoming studies.

V. Conclusions

In this paper, we formulated the AoI minimization as a reinforcement learning problem and trained a DQN, popular deep reinforcement learning algorithm, to control the network's actions on an unknown network topology and delay distribution. It is shown that, with no prior knowledge, our approach converged to the optimal solution. Furthermore, by utilizing the Universal Approximation Theorem [21], it is shown that the approach can be scaled up to any network with any delay distribution. Moreover, using off-policy RL algorithms (e.g. Q-Learning), a network that controls the network can be trained without simulation, given the transition data exists. With the trust of the results which are obtained from the experiments and the theory of convergence of the algorithm, we propose that an all-purpose network age optimizer can be generated using off-policy neural network algorithms. It is important to note that this is a proof of concept to illustrate that learning-based methods can be applied to keep the data fresh on networks.

VI. Acknowledgements

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) Project No. 117E215.

References

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

[2] E. Najm, R. Nasser, "Age of information: The gamma awakening", IEEE ISIT, pp. 2574-2578, July 2016.

[3] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, pp. 3008-3012, 2015.

[4] B. T. Bacinoglu, E. T. Ceran and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," 2015 Information Theory and Applications Workshop (ITA), San Diego, CA, pp. 25-31, 2015.

[5] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Optimizing data freshness, throughput, and delay in multi-server information-update systems," in Proc. IEEE ISIT, pp. 2569–2573, July 2016.

[6] A. M. Bedewy, Y. Sun and N. B. Shroff, "Age-optimal information updates in multihop networks," 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, pp. 576-580, 2017.

[7] R. D. Yates and S. Kaul, "Real-time status updating: Multiple sources," 2012 IEEE International Symposium on Information Theory Proceedings, Cambridge, MA, pp. 2666-2670, 2012.

[8] A. M. Bedewy, Y. Sun and N. B. Shroff, "Optimizing data freshness, throughput, and delay in multi-server information-update systems," 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, pp. 2569-2573, 2016.

[9] Y. Sun, E. Uysal-Biyikoglu, R. Yates, C.E. Koksal, N.B. Shroff, "Update or wait: How to keep your data fresh," IEEE INFOCOM 2016, pp. 1-9, April 2016.

[10] Y. Sun, Y. Polyanskiy and E. Uysal-Biyikoglu, "Remote estimation of the Wiener process over a channel with random delay," IEEE International Symposium on Information Theory (ISIT), Aachen, pp. 321-325, 2017.

[11] B. T. Bacinoglu and E. Uysal-Biyikoglu, "Scheduling status updates to minimize age of information with an energy harvesting sensor," IEEE International Symposium on Information Theory (ISIT), Aachen, 2017, pp. 1122-1126, 2017.

[12] Baran Tan Bacinoglu, Yin Sun, Elif Uysal-Biyikoglu, and Volkan Mutlu, "Achieving the Age-Energy Tradeoff with a Finite-Battery Energy Harvesting Source," submitted to IEEE International Symposium on Information Theory (ISIT), 2018.

[13] Ceran, E. T., Gunduz, D., Gyorgy, A. (2017). "Average Age of Information with Hybrid ARQ under a Resource Constraint". arXiv preprint arXiv:1710.04971.

[14] J. Ahrenholz, C. Danilov, T. R. Henderson and J. H. Kim, "CORE: A real-time network emulator," MILCOM 2008 - IEEE Military Communications Conference, San Diego, CA, 2008, pp. 1-7, 2008.

[15] C. Kam, S. Kompella and A. Ephremides, "Experimental evaluation of the age of information via emulation," MILCOM 2015 - IEEE Military Communications Conference, Tampa, FL, pp. 1070-1075, 2015.

[16] V. Mnih, et al., "Human-level control through deep reinforcement learning," Nature, 518(7540), 529, 2015.

[17] Sutton, R. S., & Barto, A. G. (1998). "Introduction to reinforcement learning" (Vol. 135). Cambridge: MIT press.

[18] Watkins, C. J. C. H., "Learning from delayed rewards," Doctoral dissertation, King's College, Cambridge, 1989.

[19] Lin, L. J., "Reinforcement learning for robots using neural networks," (No. CMU-CS-93-103). Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1993.

[20] Van Hasselt, H., Guez, A.,Silver, D. "Deep Reinforcement Learning with Double Q-Learning," in AAAI, Vol. 16, pp. 2094-2100, Feb. 2016.

[21] Hornik, K., Stinchcombe, M., & White, H., "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks". Neural networks, 3(5), 551-560, 1990.

[22] S. Baghaee, S. Zubeyde Gurbuz, and E. Uysal-Biyikoglu, "Implementation of an enhanced target localization and identification algorithm on a magnetic WSN," IEICE Transactions on Communications , vol. E98-B, no. 10, pp. 2022-2032, October 2015.

[23] S. Baghaee, S. Z. Gurbuz and E. Uysal-Biyikoglu, "Application and Modeling of a Magnetic WSN for Target Localization," 2013 UKSim 15th International Conference on Computer Modelling and Simulation, Cambridge, pp. 687-692, 2013.