An Experimental Framework for Age of Information and Networked Control via Software-Defined Radios

Onur Ayan
Chair of Communication Networks
Technical University of Munich
Munich, Germany
onur.ayan@tum.de

H. Yağız Özkan
Chair of Communication Networks
Technical University of Munich
Munich, Germany
yagiz.oezkan@tum.de

Wolfgang Kellerer
Chair of Communication Networks
Technical University of Munich
Munich, Germany
wolfgang.kellerer@tum.de

***Abstract*—Cyber-physical systems (CPS) classify the set of applications where a physical, real-time process is monitored and controlled over a communication network. In CPS, providing fresh information is essential to satisfy the requirements imposed by time-critical applications. In order to quantify information freshness, age of information (AoI) has been proposed and employed as a metric for cross-layer design. In contrast to the vast majority of AoI-research, there have been a few attempts to measure AoI in real deployment scenarios. However, those contain either unalterable communication stack or are not publicly available for possible extensions. In this work, we present an open-source, experimental framework that is using software-defined radios for wireless communication. Our implementation contains centralized resource allocation using beacon packets and various conventional packet management policies such as first come first serve and last come first serve. In a case study with multiple inverted pendulums sharing a wireless channel, we show how the communication stack can be tailored to keep the information fresh in the network. We present the performance of feedback control loops in relation to AoI and show the benefit of keeping the information fresh on realistic CPS applications.**

***Index Terms*—Age of Information, Cyber-Physical Systems, Networked Control Systems, Software Defined Radios**

I. INTRODUCTION

With the ongoing advances in computing, communications and process control, cyber-physical systems (CPS) are envisioned to constitute a serious portion of the data traffic in communication networks. In typical CPS scenarios, computers monitor and control physical processes that rely on fast and regular exchange of information over networks. Collision avoidance for autonomous driving, environmental control, distributed robotics, manufacturing and smart structures are some of the most prominent examples of CPS [1] where a feedback loop is closed over a communication network.

Due to their time-sensitive nature, CPS applications gave rise to metrics as Age of Information (AoI) to be adopted in the cross-layer network design. AoI is defined as the elapsed time since the generation of the most recent information. It captures the information freshness at the receiver that is monitoring and controlling a remote process [2]. It serves as a cross-layer metric since it is defined in the application layer but can be employed for decision making within the lower layers [3]–[5]. However, such cross-layer considerations are feasible in real-world scenarios only if the lower layer policies such as queuing and medium access control (MAC) are modifiable during the process of system design. Proposing such an open-source, experimental framework where one can implement custom algorithms in any layer of the communication stack is the goal of this paper.

In this work, we present an experimental framework implemented with GNU Radio that employs software-defined radios as hardware. Our main contributions can be summarized as follows:

- First open-source implementation of a real-world control application where the feedback loop is closed over a completely modifiable communication stack.
- Performance evaluation of different medium access and packet management policies using real-world equipment in the context of information freshness in control over networks.
- A baseline protocol for centralized allocation of wireless resources via broadcast packets that is easily extendable.

In addition to AoI performance, we provide results on the achieved performance of feedback control loops, i.e., *Quality-of-Control (QoC)*, when different queuing and medium access strategies are applied.

A. Related Work

The vast majority of the AoI research focuses on theoretical work [4], [6] as summarized in [7]. In particular, there have been only a few practical works mentioning AoI [8]–[12] that conduct a study with real-world equipment. [8], [10], [11] measure AoI performance using various technologies, namely Ethernet, Wifi, 2G, 3G and LTE.

To the best of our knowledge, the only implementation-based works mentioning AoI and proposing a customized solution are [9], [12]. They measure AoI in a network of multiple software-defined radios (SDRs) where the considered application is timestamped packet exchange but not a real-world example requiring time-critical information. In [9],

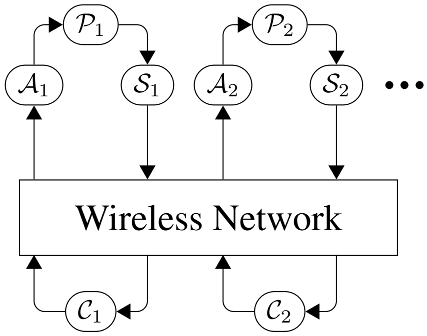

Fig. 1: Considered scenario with multiple sensor-controller pairs exchanging information via the shared wireless communication network.

authors propose a MAC layer protocol using last come first serve (LCFS) queues and polling mechanism managed by a central node. They show that AoI is decreased significantly through the use of LCFS strategy. Authors in [12] use SDRs programmed with GNURadio, which is a software framework with a rich selection of signal processing blocks. They compare two different random access protocols, namely *Age-Independent Random Access* and *Age-Dependent Random Access* . They show that taking AoI into account while deciding on channel access at each user improves the network's AoI performance when compared to the age-independent strategy.

None of the existing works consider a realistic, time-critical application at the source which lies at the root of the motivation for fast and regular status updates. Moreover, most of the practical AoI research consists of measurements conducted on real-world equipment without any modifications inside the communication stack. Even though [9] and [12] propose a customized MAC layer implementation with SDRs, they are neither able to provide any open-source code to be used by the AoI community, nor any comprehensive study including different queuing and medium access strategies jointly.

II. SCENARIO & CONTROL MODEL

We consider multiple feedback control loops communicating over a shared wireless medium. Each loop consists of a plant $\mathcal{P}_i$, a sensor $\mathcal{S}_i$, a controller $\mathcal{C}_i$ and an actuator $\mathcal{A}_i$. The task of $\mathcal{S}_i$ is to sample the plant state periodically at each time step k and transmit it over the wireless link to $\mathcal{C}_i$. Upon reception of a new state measurement, controller responds with a control input to $\mathcal{A}_i$ where it is applied to the plant in order to drive the plant state to a desired value. We assume that each $\mathcal{S}_i$, $\mathcal{P}_i$ and $\mathcal{A}_i$ is co-located while $\mathcal{C}_i$ operates remotely. A practical example corresponding to this architecture is formation control of drones from a ground station with drone being the plant, ground station being the controller and set of motors installed on the drones being the actuators. Fig. 1 illustrates the considered scenario throughout the paper.

We denote the k-th plant state by $\boldsymbol{x}_i[k] \in \mathbb{R}^{n_i}$ and the control input applied at the k-th time step by $\boldsymbol{u}_i[k] \in \mathbb{R}^{m_i}$. We assume a request-response system in which the controller transmits only upon a reception of a sensor measurement, but not triggering any transmissions itself. We characterize the discrete time dynamics of our systems, which are sampled with a constant frequency f_s, by a linear time-invariant (LTI) model as:

$$\boldsymbol{x}_i[k+1] = \boldsymbol{A}_i\boldsymbol{x}_i[k] + \boldsymbol{B}_i\boldsymbol{u}_i[k] + \boldsymbol{w}_i[k], \tag{1}$$

where $\boldsymbol{A}_i \in \mathbb{R}^{n_i \times n_i}$ and $\boldsymbol{B}_i \in \mathbb{R}^{n_i \times m_i}$ are the system and input matrices, respectively[1]. Disturbances on the plant state are modeled by a zero-mean, normally distributed random noise vector with covariance matrix $\boldsymbol{\Sigma_i}$, i.e., $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma_i})$.

The control input at $\mathcal{C}_i$ to any state measurement $x_i[k']$ is obtained by the following equation:

$$\boldsymbol{u}'_i[k'] = -\boldsymbol{K}_i\boldsymbol{x}_i[k'], \tag{2}$$

with the feedback gain matrix $\boldsymbol{K}_i \in \mathbb{R}^{m_i \times n_i}$. We assume that controller design is independent of the network, thus $\boldsymbol{K}_i$ is a time-invariant matrix. Network dependent control design, which aims to compensate the negative of wireless networks, exists in the CPS literature. However this is beyond the scope of this paper.

In our considered scenario, each sensor-to-controller link as well as the controller-to-actuator link consists of a single transmission, thus the feedback loop is closed over a 2-hop wireless communication network. In such a setting, the freshest control input received both by controllers and actuators can be outdated. This leads to degradation of the control performance due to sub-optimal actuation at $\mathcal{A}_i$. Nevertheless, we do not assume any intelligence at actuators, as in many practical scenarios, and hold the latest received control input constant until a new packet is received from $\mathcal{C}_i$. In other words, $\boldsymbol{u}_i[k] = \boldsymbol{u}_i[k-1]$ holds in case there hasn't been any successful reception on the controller-to-actuator link during the last sampling period.

In case of a successful update at $\mathcal{A}_i$, $\boldsymbol{u}_i[k] = -\boldsymbol{K_i}\boldsymbol{x}_i[k - \Delta_i[k]]$ describes the behavior of $\boldsymbol{u}_i$ according to the control law defined in Eq. (2). $\boldsymbol{x}_i[k - \Delta_i[k]]$ denotes the freshest plant state that $\mathcal{C}_i$ has successfully received, processed and forwarded to $\mathcal{A}_i$ by the beginning of the k-th sampling period[2]. Here, we define $\Delta_i[k]$ as the age of information in our i-th feedback loop measured in units of time steps. It is important to mention that the information content changes from state to control input along the $\mathcal{S}_i$-to-$\mathcal{C}_i$-to-$\mathcal{A}_i$ path. Nevertheless, only the final information, namely the applied control input, has an effect on the control performance. One could introduce an additional variable defining only the AoI at the controller, however this would not play any role within the application layer. Therefore, we omit this intermediate definition due to presentation purposes and proceed only with the AoI at actuators.

---

[1] It is a conventional method in control theory to represent continuous time systems in discrete time dynamics [13]. This coincides with the practical CPS scenarios which include digital components communicating over networks.

[2] Throughout the paper, we assume that a control input can be used by $\mathcal{A}_i$ in time step k only if it has been received before the beginning of the k-th sampling period.

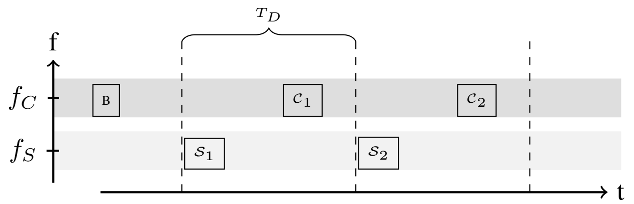

Fig. 2: Example diagram of an information exchange by two feedback control loops. $\mathcal{S}_1$ and $\mathcal{S}_2$ represent the status update packets sent by the two sensor nodes. Similarly, $\mathcal{C}_i$ are the control inputs that are sent within the same slot as the corresponding status updates. B stands for the beacon packet. Note that status updates are sent on a different frequency, f_S than the beacon and control packets, i.e., f_C.

III. Design and Implementation

The implementation in [14], which constitutes the starting point of this work, does not contain any MAC layer considerations. In other words, having a new information to send, each user sends the incoming packet to the physical layer processing blocks of GNU Radio from where it is forwarded to the transceiver. However, leveraging the fundamentals of communication theory, we know that this may cause high packet loss due to simultaneous access of the wireless medium by multiple nodes.

A. Scheduling with Beacon Packets

To reduce packet loss and its negative effects on QoC, we implement a wireless MAC protocol based on time-slotted, centralized resource allocation via broadcast packets, i.e., ***beacon packets*** that are sent periodically by a central node. Among other fields that are required for data-link layer processing, such as packet header, cyclic redundancy check, beacon packets carry:

- duration of each time slot, i.e., T_D,
- transmission schedule for the next T_S time slots, which form a *superframe* together[3].

Upon a beacon reception, each sensor node in the network aligns its timing in order to keep track of time slots. The node, to which the current time slot is allocated according to the latest beacon packet, initiates the transmission of its next packet. All sensor nodes except the currently scheduled one, do not access the channel.

Status update packets, transmitted from each $\mathcal{S}_i$ to $\mathcal{C}_i$ contain:

- a unique identifier of the feedback loop, i.e., i,
- sensor measurement data, i.e., $\boldsymbol{x}_i[k']$,
- generation time step of data, i.e., k'.

Moreover, having received a status update packet containing $\boldsymbol{x}_i[k']$, controller replies directly with a ***control packet*** which carries the following information:

- Unique identifier of the feedback loop,
- control input $\boldsymbol{u}_i[k']$ as a response to the corresponding status packet,
- generation time step of data, i.e., k'.

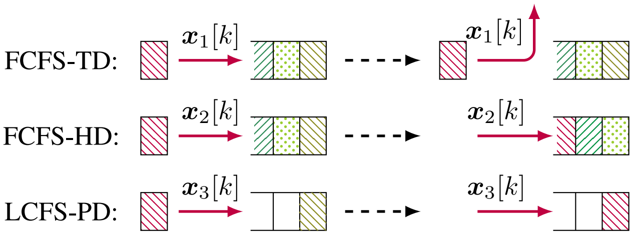

Fig. 3: Illustration of the selected FCFS-TD, FCFS-HD and LCFS-PD packet management policies when the transmission queue has a capacity of 3. FCFS-TD and FCFS-HD discard the most recent and oldest packet available, respectively. In LCFS-PD policy, the new packet replaces the one in the queue.

For design and presentation purposes, our implementation employs frequency-division duplexing (FDD), i.e., beacon and control packets are transmitted on a different channel than status update packets. Fig. 2 depicts an example diagram where following a beacon packet, status and control packets of two feedback loops are exchanged in two consecutive time slots.

B. Packet Management Policies

In order to investigate the effect of packet management on AoI and control performance, we implemented various policies that are commonly used in the literature[4]:

- First come first serve with tail drop (FCFS-TD)
- First come first serve with head drop (FCFS-HD)
- Last come first serve with packet discard (LCFS-PD)

All implementations include a limited capacity of the transmission queue at each $\mathcal{S}_i$. If the queue is full and a new packet is received from the application layer, FCFS-TD discards the newest packet without any enqueuing. On contrary, FCFS-HD discards the oldest packet in the queue, which is located at the head of the queue. Both FCFS-TD and FCFS-HD policies add any incoming packet to the queue from its back, i.e., queue's tail, while LCFS-PD discards the packet at the queue's head prior to enqueuing. Thus, the LCFS-PD serves only the most recent packet and discards all the others without processing further. Fig. 3 depicts the operation of the packet management policies for a better understanding.

C. Case Study: Inverted Pendulum

For our considered scenario, we select control of an inverted pendulum (IP). An IP is a pendulum mounted on a motorized cart, which is illustrated in Fig. 4. The controller's objective is to hold the pendulum in an upright position by moving the cart back and forward. Due to its unstable nature, i.e., the pendulum falls if the cart is not controlled correctly, it is an example CPS application that requires fresh information to

---

[3] If a sensor node fails to decode a beacon packet successfully, the transmission schedule from the previous superframe is assumed to be valid.

[4] Our framework contains more policies than this work presents. We limit ourselves to 3 policies due to presentation and space considerations.

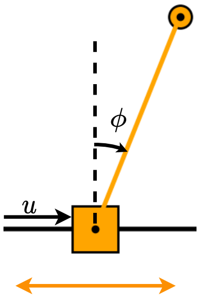

Fig. 4: Illustration of an inverted pendulum that is balanced through the movements of a cart. $\boldsymbol{u}$ is applied to move the cart back and forward in order to keep the pendulum vertical. The dynamics of the system can be linearized around the equilibrium point, which is located at $\phi = 0$.

operate correctly. It is commonly used in control theory textbooks and in the literature [15], [16]. Discrete time dynamics of an IP that is sampled with a frequency of 100 Hz can be characterized with the following $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices for all i:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 9.99 \times 10^{-3} & 9.56 \times 10^{-5} & 3.19 \times 10^{-7} \\ 0 & 0.998 & 1.91 \times 10^{-2} & 9.56 \times 10^{-5} \\ 0 & -2.44 \times 10^{-5} & 1 & 0.01 \\ 0 & -4.88 \times 10^{-3} & 0.335 & 1 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} 8.53 \times 10^{-5} \\ 1.71 \times 10^{-2} \\ 2.44 \times 10^{-4} \\ 4.88 \times 10^{-2} \end{bmatrix}.$$

We apply Linear Quadratic Regulator (LQR) method to determine the stabilizing feedback gain matrix:

$$\boldsymbol{K} = \begin{bmatrix} -61.9 & -32.1 & 87.9 & 16.3 \end{bmatrix}.$$

The covariance matrix of the multivariate random noise is chosen as:

$$\boldsymbol{\Sigma} = \begin{bmatrix} 6.4 \times 10^{-9} & 0 & 0 & 0 \\ 0 & 4.9 \times 10^{-9} & 0 & 0 \\ 0 & 0 & 1.22 \times 10^{-7} & 0 \\ 0 & 0 & 0 & 1.22 \times 10^{-7} \end{bmatrix}.$$

IV. RESULTS AND EVALUATION

In this section, we compare different combinations of scheduling and packet management policies from Sec. III. They are implemented using GNU Radio which is a software framework that provides a wide range of signal processing blocks. The resulting code is run on Ettus Research's USRP B200mini-i which are lower-priced SDRs than the ones used in [9] and [12]. Our implementation is publicly available at: https://github.com/oayan/gr-ieee802-15-4/tree/maint-3.7.

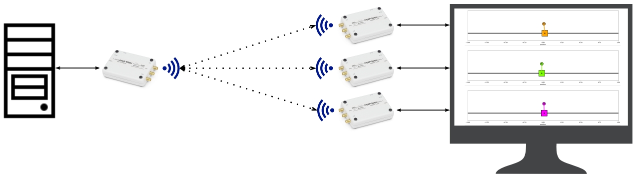

Fig. 5: Drawing of our measurements setup, which consist of 4 USRP B200mini-is. 3 SDRs are used for sensor-to-controller transmission and one common SDR is used for controller-to-actuator transmission. On the monitor, three inverted pendulums are emulated as control applications.

Our setup consists of three inverted pendulum processes that run in parallel on a computer. Traffic generated by each process is forwarded to a different SDR from where it is sent over a shared channel to the controller SDR. Fig. 5 shows our experimental platform with 4 SDRs and the graphical user interface for the three inverted pendulums on a monitor. The controller SDR runs on a different computer and thus operates remotely. Each plant $\mathcal{P}_i$ is sampled with a frequency of 100 Hz and T_D is selected as 8 ms. Round Robin (RR) strategy is used to determine the transmission schedule for the next $T_S = 30$ time slots, i.e., each loop is granted channel access once in every 3 time slots. Each scenario, namely scheduling-packet management policy combination, is repeated 40 times where each run takes 20 seconds. In addition to FCFS-TD, FCFS-HD and LCFS-PD combined with scheduling through beacons, the performance of the implementation from [14] is measured. We refer to it as random access (RA) in the rest of this section as it does not contain any centralized scheduling or any packet management prior to physical layer. Note that, the packet length is the same in all policies for a fair comparison.

We present round-trip time (RTT) for every received control packet. We define RTT as the time between the generation of a status update packet and the reception of the corresponding control packet with the same sequence number. We are able to measure the RTT without any synchronization as each $\mathcal{P}_i$, $\mathcal{S}_i$ and $\mathcal{A}_i$ runs on the same device. Fig. 6 shows the RTT performance of the selected policies where the measurements are done in the application layer. Note that RTT is measured only through successfully received control packets. As all policies except RA include queuing delay and/or waiting time for the next allocated time slot, they do not perform as good as the RA policy. To avoid visual clutter, Fig. 6 does not display outliers. Maximum and minimum values are shown instead.

Fig. 7 shows the average packet loss rate of 3 inverted pendulums sharing a wireless channel. We observe that the RA policy has encountered an average loss rate up to 80% during a single run while the overall average loss rate is approximately 40%. Moreover, the beacon-based scheduling that is employed by FCFS-TD, FCFS-HD and LCFS-PD policies, is able to reduce the average packet loss significantly, i.e., down to approximately 15%. It is important to emphasize that the packet loss is measured only among total transmission attempts and discarding packets due to packet management policy does not play a role.

Since our original aim is to increase information freshness and QoC in the network through packet management and scheduling, RTT and packet loss rate are not sufficient as stand alone metrics to quantify application's performance. As we

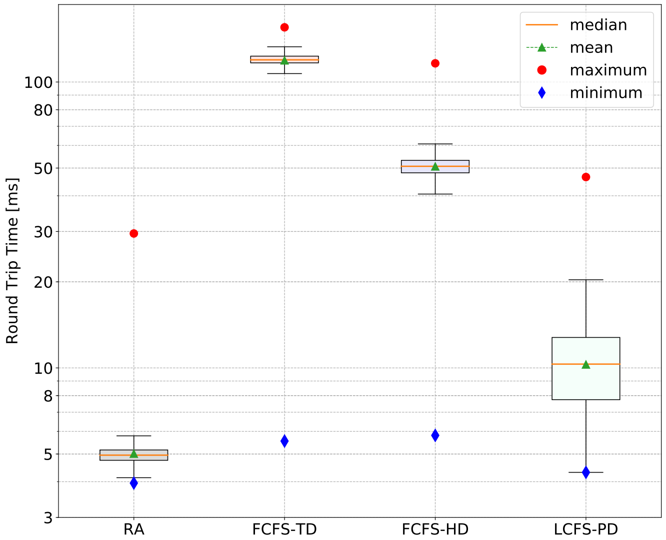

Fig. 6: Round-trip time performance of 3 feedback control loops measured in application layer. RA does not include any MAC considerations. FCFS-TD, FCFS-HD and LCFS-PD policies are based on centralized resource scheduling through beacon packets. Round robin is applied as the scheduling policy. Outliers are not displayed to avoid visual clutter.

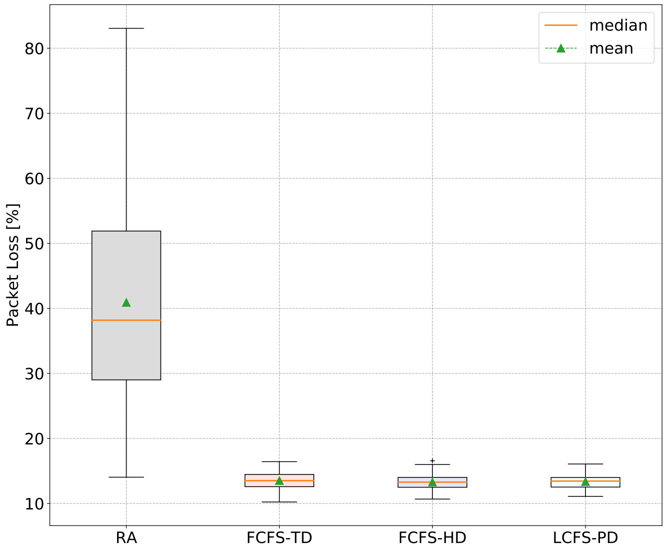

Fig. 7: Packet loss rate of 3 feedback control loops when various packet management and medium access policies are applied. RA does not include any MAC considerations. FCFS-TD, FCFS-HD and LCFS-PD policies are based on centralized resource scheduling through beacon packets. Round robin is applied as the scheduling policy.

Policies	**Time To Failure**		
	Mean	***Median***	***Maximum***
RA	4.96 s	2.73 s	∞
FCFS-TD	0.55 s	0.53 s	0.93 s
FCFS-HD	0.81 s	0.81 s	1.24 s
LCFS-PD	∞	∞	∞

Tab. I: For the considered policies, mean, median and maximum time to failure is given. Successful runs are not considered in mean and median calculations for the RA policy. LCFS-PD policy was able to satisfy the requirements during every run.

have seen in Fig. 6 packets may experience lower RTT but a high loss rate due to simultaneous medium access. Therefore, Fig. 8 gives better insights into information freshness resulting from packet loss rate and RTT combined. Results show that transmitting only the most recent information while avoiding collisions, LCFS-PD, outperforms other policies that serve outdated packets first, i.e., FCFS-TD, FCFS-HD. This result coincides with the theoretical findings from the literature [17]. Furthermore, providing regular updates is essential in order to reduce information staleness at the receiver as we can see from the comparison between RA and LCFS-PD policies.

In addition to AoI, we investigate the ability of feedback loops to stay stable following an initialization at upright position. To that end, we select ± 20 degrees as the maximum allowed deviation as performance criterion. Over 40 runs of length 20 seconds, we measured when each pendulum exceeds ± 20 degrees for the first time, i.e., time to failure. Tab. I summarizes the results with respect to control performance. Infinite sign represents a successful control within the given bounds throughout the measurements.

From Tab. I, we see that LCFS-PD policy with centralized scheduling succeeded to keep the pendulum's angle within ± 20 degrees during every measurement. Moreover, RA policy's mean time to failure is 4.96 seconds, while it was able to satisfy the control requirements at least once. On the other hand, FCFS-TD and FCFS-HD policies were able to keep the pendulum upright at most 1.93 and 1.24 seconds long, respectively. From the difference between RA and FCFS-based policies, we observe that given the system model and control law from Sec. III is applied, QoC of IPs is more sensitive to delayed information than infrequent exchange of data. We emphasize that this is strictly scenario and application dependent.

V. CONCLUSION

Age of information has been used in communication networks as a metric to quantify information freshness. Vast majority of the state-of-the-art in AoI consists of theoretical works. There have been a few attempts recently to measure AoI with real hardware. However, those works employ either unalterable communication devices or do not present any performance gain in the application layer that is caused by improving data freshness.

In this work, we studied the AoI in the context of networked control systems, where multiple inverted pendulum applications exchange data via software-defined radios. Our framework contains various medium access and packet management policies implemented with GNU Radio platform. Our scenario considers inverted pendulum as application and

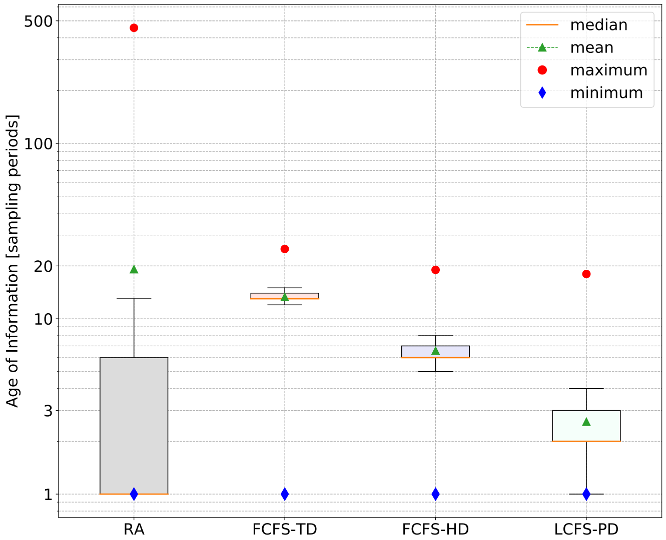

Fig. 8: AoI performance of feedback control loops measured in units of sampling period. Outliers are not displayed to avoid visual clutter. Maximum values reached during measurements are shown with circle markers instead.

studies the resulting performance as an additional metric to quantify performance beyond AoI. We hope that through our framework presented in this work, the practical AoI research becomes easier to implement, extend and deploy.

ACKNOWLEDGMENT

This work has been carried out with the support of DFG priority programme Cyber-Physical Networking (CPN) with the grant number KE 1863/5-2.

REFERENCES

[1] E. A. Lee, "Cyber physical systems: Design challenges," in *11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 2008.

[2] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 Proceedings IEEE INFOCOM*, 2012.

[3] O. Ayan, M. Vilgelm, M. Klügel, S. Hirche, and W. Kellerer, "Age-of-information vs. value-of-information scheduling for cellular networked control systems," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, 2019.

[4] I. Kadota, A. Sinha, E. Uysal-Bıyıkoğlu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," in *IEEE/ACM Transactions on Networking*, 2018.

[5] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "Age of information performance of multiaccess strategies with packet management," vol. 21, no. 3, 2019.

[6] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The Age of Incorrect Information: A New Performance Metric for Status Updates," in *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, 2020.

[7] R. D. Yates, Y. Sun, I. Brown, D. Richard, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of Information: An Introduction and Survey," *arXiv e-prints*, p. arXiv:2007.08564, Jul. 2020.

[8] C. Sönmez, S. Baghaee, A. Ergişi, and E. Uysal-Bıyıkoğlu, "Age-of-Information in Practice: Status Age Measured Over TCP/IP Connections Through WiFi, Ethernet and LTE," in *IEEE International Black Sea Conference on Communications and Networking*, 2018.

[9] I. Kadota, M. S. Rahman, and E. Modiano, "WiFresh: Age-of-Information from Theory to Implementation," *arXiv e-prints*, p. arXiv:2012.14337, Dec. 2020.

[10] H. B. Beytur, S. Baghaee, and E. Uysal, "Towards AoI-aware Smart IoT systems," in *International Conference on Computing, Networking and Communications*, 2020.

[11] B. Barakat, H. Yassine, S. Keates, I. Wassell, and K. Arshad, "How to measure the average and peak age of information in real networks?" in *25th European Wireless Conference*, 2019.

[12] Z. Han, J. Liang, Y. Gu, and H. Chen, "Software-Defined Radio Implementation of Age-of-Information-Oriented Random Access," 2020, p. arXiv:2003.14329.

[13] K. J. Astrom and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

[14] B. Bloessl, C. Leitner, F. Dressler, and C. Sommer, "A GNU Radio-based IEEE 802.15.4 Testbed," in *2. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze*, 2013.

[15] M. Vilgelm, O. Ayan, S. Zoppi, and W. Kellerer, "Control-aware uplink resource allocation for cyber-physical systems in wireless networks," in *23th European Wireless Conference*, 2017.

[16] X. Liu and A. Goldsmith, "Wireless medium access control in networked control systems," in *American Control Conference*, 2004.

[17] S. K. Kaul, R. D. Yates, and M. Gruteser, "Status updates through queues," in *46th Annual Conference on Information Sciences and Systems*, 2012.