Scheduling of Wireless Edge Networks for Feedback-Based Interactive Applications

Samuele Zoppi*, Jaya Prakash Champati†, James Gross‡, and Wolfgang Kellerer*
*Chair of Communication Networks, Technical University of Munich, Germany
†IMDEA Networks Institute, Spain
‡School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden
Email: {samuele.zoppi, wolfgang.kellerer}@tum.de, jamesgr@kth.se, jaya.champati@imdea.org

***Abstract*—Interactive applications with automated feedback will largely influence the design of future networked infrastructures. In such applications, status information about an environment of interest is captured and forwarded to a compute node, which analyzes the information and generates a feedback message. Timely processing and forwarding must ensure the feedback information to be still applicable; thus, the quality-of-service parameter for such applications is the end-to-end latency over the entire loop. By modelling the communication of a feedback loop as a two-hop network, we address the problem of allocating network resources in order to minimize the delay violation probability (DVP), i.e. the probability of the end-to-end latency exceeding a target value. We investigate the influence of the network queue states along the network path on the performance of semi-static and dynamic scheduling policies. The former determine the schedule prior to the transmission of the packet, while the latter benefit from feedback on the queue states as time evolves and reallocate time slots depending on the queue's evolution. The performance of the proposed policies is evaluated for variations in several system parameters and comparison baselines. Results show that the proposed semi-static policy achieves close-to-optimal DVP and the dynamic policy outperforms the state-of-the-art algorithms.**

***Index Terms*—Feedback applications, end-to-end delay, delay violation probability, network state information, semi-static scheduling, dynamic scheduling, MDP.**

I. Introduction

Interactive applications with automated feedback are arguably one of the most discussed application class when it comes to large-scale impact in future networked infrastructures today [1]. In the literature, we can broadly distinguish two sub-cases of this new application class, namely cyber-physical systems (CPS) and human-in-the-loop systems [2]–[4]. In both cases, however, the underlying principle is the same: status information about a plant or an environment of interest is captured, and forwarded to a compute node, where the information is analyzed and potentially a feedback is generated in form of an actuation command or an augmentation/perceptual feedback; see Fig. 1. In CPS, we encounter applications where direct actuation is applied to a physical object, for instance, in the context of industrial automation and automated driving. In contrast, augmented reality, cognitive assistance, and also to some extent virtual reality fall under the category of human-in-the-loop systems, where no direct actuation results from the feedback; instead, a human is presented perceptual feedback which potentially triggers some human reaction [4].

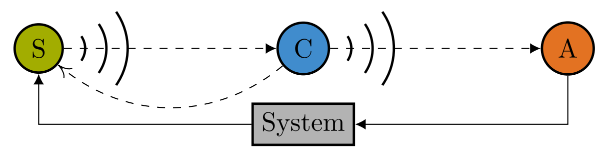

Fig. 1. Architecture of interactive applications with automated feedback. Sensor (S), controller (C), and actuator (A) operate in a closed loop via a two-hop wireless network path.

Key features about these applications is their tight integration with reality as well as the degree of automation that they allow. Furthermore, by offloading these applications to networked infrastructures, an almost ubiquitous availability of corresponding services will be enabled in the future. However, the successful deployment of such applications rests crucially on a timely processing and forwarding along the pipeline to ensure that the feedback information is timely with respect to the original sensing data. Thus, the quality-of-service (QoS) parameter for interactive applications is the latency over the entire loop, i.e. from capturing the status information until the point in time when the corresponding feedback information is exposed [5]. For instance, motion-to-photon latency in virtual reality is a well-known concept that captures this QoS parameter. Loosely speaking, one might refer to this QoS parameter as the *end-to-end latency* with the crucial differentiation that a flow conversion occurs at the point of computation. In fact, depending on the application, for a certain fraction of sensing data no feedback is generated at all, for instance in the case of cognitive assistance. Finally, different applications may also require time-varying end-to-end latencies. For instance, in CPS it is known that the criticality of sensing information can vary, leading to time-varying end-to-end latency constraints as the plant dynamics evolve.

Given the relevance of interactive applications, as well as their novel QoS requirements, a central question relates to the optimal support of such applications by networked infrastructures. Truly ubiquitous service offering mandates wireless connectivity to the point of computation. Furthermore, general latency requirements mandate near-by computational service, typically provided by edge computing [6]. An exemplary wireless network infrastructure serving the QoS of interactive applications is WirelessHART in the industrial scenario. Here, the star topology wireless network allows sensors and actuators of industrial CPS applications, such as production machines and mobile robots, to be interconnected with the control logic, usually co-located with the network coordinator, at the centre of the network. Hence, a networked infrastructure realizing an interactive application needs to provide bounded end-to-end latencies over a concatenated, heterogeneous network path with at least one compute element incorporated [1]. This complicates the provisioning of end-to-end latencies, as the individual elements of the network path are typically subject to multiple random effects, such as fading in case of the wireless links and/or cross-traffic for both the communication and computation elements [7]. As a consequence, the end-to-end latency becomes essentially a random variable, which also depends on parameterizations of the network path, such as resource allocation. In order to steer the parameterization of the network path facing the uncertainty from different effects, a suitable metric is to minimize the delay violation probability (DVP), defined as the probability of the end-to-end latency exceeding a (constant or varying) target value. The key question then is to manage network path resources to minimize the DVP for a given deadline. Addressing this question is at the heart of this work.

We refer to the network behavior under resource allocation over a finite time horizon is as the transient regime [7]. Under this regime, the typical stability assumptions of the queuing system are not required for the steady-state to exist and the latency performance of the network is influenced by the *initial conditions* of the network path. By initial conditions, we refer to the queue states along the network path, which is modelled as a concatenated queuing network. This can be included in the resource allocation policies in at least two different ways. On the one hand, once a new sensor arrival enters the network, a *semi-static* resource allocation can be determined based on the initial state of the queues, which is kept during the subsequent evolution of the system until the corresponding actuation command is delivered to the actuator. On the other hand, a *dynamic* policy re-computes the resource allocation after every frame during the evolution of the system based on the current queue states until the actuation command is delivered. We note that a semi-static policy is a special case of a dynamic policy where the queue state information in all frames, except the first frame, is ignored, and one can expect that dynamic policies provide a better performance in terms of observed delay violations at the price of increased signalling load. But exactly how such algorithms should work, which complexity they bring, and which performance differences they imply for interactive applications is to the best of our knowledge open to date.

In this paper, the above questions are investigated for a two-hop network path incorporating the loop communication of the status information to a compute node and of the feedback message to an actuator, cf. Fig. 1. The two-hop system follows an uplink/downlink model where network resources need to be assigned *in competition* over a packet-erasure channel. This applies to control loops closed over the same node that originates the data and where the actuation data terminates as well and communication systems where time resources are typically shared between uplink and downlink, such as WirelessHART, LTE TDD, and 5G TDD NR. The end-to-end latency of packets is dominated by the random delay caused by the retransmission of lost packets and thus the processing delay introduced by the compute node is assumed to be negligible. Packet transmission follows a time-slotted medium access where network resources are organized in frames. In each frame, the available time slots are entirely allocated to the two links that compete for resources. Therefore, given the initial queue states of the two links, we investigate scheduling policies that allocate time slots of each frame to the links in order to minimize the DVP of packets belonging to interactive applications.

The main contributions of this paper are summarized in the following:

- We show that the closed-form expression of DVP is intractable and derive two upper bounds for the DVP of packets traversing a two-hop network path given the initial network conditions.
- Novel heuristic scheduling policies that compute a semi-static resource allocation are proposed.
- Noting that DVP cannot be directly used for dynamic resource allocation, a dynamic heuristic scheduling policy that maximizes the network's throughput is proposed.
- A comprehensive simulation analysis is performed in order to compare the DVP of proposed semi-static policies with the optimum semi-static policy and the DVP of the proposed dynamic scheduling policy with the classical Backpressure (BP) [8], Max Weight (MW) [9], and Weighted-Fair Queuing (WFQ) [10] policies, respectively.

The rest of the paper is structured as follows. Sec. II provides a discussion of the related work. Sec. III defines the model of the two-hop network path and the problem statement. Sec. IV provides a general derivation of DVP and discusses its application for scheduling. In Sec. V heuristic semi-static scheduling policies are derived, while Sec. VI describes an MDP-based heuristic dynamic scheduler. Sec. VII evaluates the performance of the proposed scheduling methods and provides a discussion on their applicability in different scenarios. Finally, we conclude in Sec. VIII.

II. Related Work

Several existing works tackle the problem of resource allocation in a multi-hop wireless network to support time-critical applications. Methods that make use of the queue state to allocate network resources follow a *theoretical* approach [8], [11], [12]. In their pioneering work [11], Tassiulas et al. derived the Max Weight scheduling policy, which allocates resources based on the transmitters' backlogs and achieves maximum throughput and minimum delay. Their scenario, however, is different from the one in this work as they only considered a single-hop network. For a multi-hop network, maximum throughput was achieved by the backpressure algorithm [8], which allocates resources based on the backpressure of queues in the network. Differently than this work that minimizes the DVP of time-critical packets, their scheduling policy focused on throughput optimality. Singh et al. [12] exploited information about the queue state to schedule transmissions in order to maximize throughput under delay constraints. Their approach, however, considers the steady-state performance of packets and, differently than our approach, does not allocate resources to optimize the network for a single time-critical arrival.

From a different perspective, many practical works investigate resource allocation methods for real-time flows in Industrial Wireless Sensor Networks (IWSN). Some of them compute schedules to allow several time-constrained applications to meet their deadlines assuming *deterministic transmission outcomes* [13]–[18]. Saifullah et al. [13], [14] investigated the problem of real-time scheduling subject to end-to-end deadlines between sensors and actuators. Differently than our work, however, communication between sensors and actuators is deterministic and packet loss is not considered. In their recent works [15], [16], communication failures due to packet loss are considered and retransmissions are used. In contrast to our work, however, the only provide a deterministic delay model for the communication between sensors, controller, and actuators. Similarly, Wang et al. [17] calculated schedules to ensure the worst-case delay of packets in a flow, however, without assuming random packet loss. Modekurthy et al. [18] derived a distributed deadline-based scheduling algorithm based on the Earliest Deadline First policy. Also in this case, differently than our scenario, packet loss is not considered and the random end-to-end delay of packets in the network is not characterized.

Other IWSN works tackle the problem of reliable communication in presence of *random packet loss* [19]–[21]. Dobslaw et al. [20], allocated resources to each transmitter in a path based on the required number of retransmissions to fulfil a given reliability constraint. Differently than our model, however, their work did not consider a deadline for the packets. Following a similar approach, Gaillard et al. [21] extended the pioneer traffic-aware centralized scheduler TASA [22] including retransmissions to guarantee flow reliability requirements. Also in this case, however, traffic is not time-constrained. Yan et al. [19] developed a scheduling method that allocates time slots to the transmitters in order to maximize the network reliability under delay constraints. A major difference of their work, which is common to Dobslaw and Gaillard et al., is that reliability constraints are defined for all the flows in the network. Our approach instead optimizes the network resources to maximize the application reliability of each time-critical packet.

Recent works tackle investigate resource allocation methods providing *per-packet* delay and reliability performance [23]–[26]. Similarly to DVP, Chen et al. [23] computed, for each packet, the number of transmissions required to fulfil the application deadline with a given probability. Their work, however, only considered a single-hop scenario and cannot be applied to the considered two-hop network path. Brummet et al. [24] developed a method to dynamically allocate retransmissions to each packet and at each network hop subject to delay and reliability requirements. They followed a different approach as their schedules are designed to fulfil the requirements and limit the maximum number of retransmissions. Therefore, their scenario did not optimize network resources to maximize the per-packet DVP. A similar approach was used by Gong et al. [25], which allocated time slots to transmitters fulfilling per-packet delay and reliability constraints while minimizing the number of resources. Also in this case, however, they considered a finite number of retransmissions and the network resources are not optimized to minimize DVP. On the contrary, Soldati et al. [26] allocated time slots over multiple hops maximizing the end-to-end reliability of each time-critical packet subject to a deadline. Differently than our approach that characterizes the distribution of end-to-end delay considering the correlation of transmission outcomes over consecutive transmitters, their scenario assumes that resource allocations of consequent transmitters are independent.

This work is based on the the mathematical fundamentals of [7] for the analysis of systems in the transient regime, leveraging stochastic network calculus. In [7], analysis of DVP was presented assuming that the services of different links are independent and identically distributed. In contrast, we consider the minimization of DVP and that the number of slots allocated to link 1 affects the number of slots allocated to link 2, making the service at link 1 correlated to that of link 2, thus requiring a new derivation for DVP. This work extends our previous work [27] by deriving semi-static scheduling policies and investigating the impact of queue state information on the minimum achievable DVP. We achieve this thanks to a transient queuing model of the network and by modelling the end-to-end delay of each packet of interactive applications.

In summary, this article extends the available state-of-the-art with the following *contributions*. We analyse the random end-to-end delay incurred for each time-critical arrival by analysing the network in the transient regime and not for stationary flows. The considered queuing model allows us to derive scheduling policies taking into account the correlation between subsequent transmitters introduced by the slot allocation. This is different from the related work as existing scheduling policies optimize network resources based on the interaction of multiple independent flows and transmitters sharing the network. Furthermore, by allocating a finite amount of retransmissions, all the existing methods allow application packets to be dropped, which may result in critical failures of feedback systems. Finally, we investigate the impact of queue state information on semi-static and dynamic scheduling policies, which, to the best of the knowledge of the authors, was never tackled in the literature for a two-hop network path.

III. System Model and Problem Statement

We study the communication scheduling problem of a feedback system consisting of a sensor, a control logic, and an actuator. The two-hop data communication from the sensor to the controller, and the controller to the actuator is performed via error-prone wireless links. We now describe the details of the model and the network elements, and present the formulation of the scheduling problem.

A. Arrivals, Backlogs, and Departures

We model the sensor-controller link and controller-actuator link using a packet-flow, discrete-time, two-queue lossy wire-

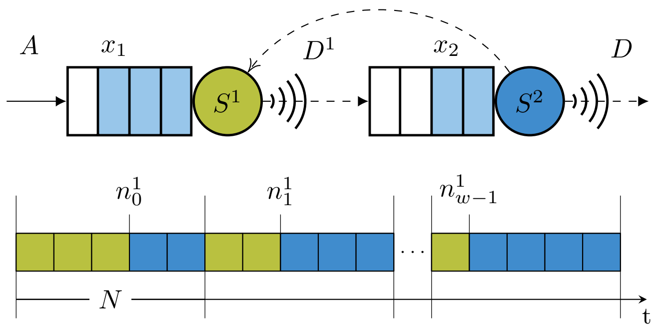

Fig. 2. Network model of the two-hop network path. The available time slots are entirely allocated to the two queues at each frame until the deadline.

less network with first-come-first-serve discipline, cf. Fig. 2. The time is discretized into slots, which are grouped in frames with N slots per frame. In this way, different number of slots can be allocated to different transmitters in each frame. This radio resource allocation is common in industrial wireless protocols such as WirelessHART.

A sequence of y packets arrives at the first queue in frame 0[1]. These packets are time critical with a requirement that they depart the second queue within next w frames, where w is finite. These packets, for instance, may belong to a time-critical message whose latency could significantly impact the application performance, i.e. the stability/safety of a feedback system. We are thus interested in analyzing the two-queue network path for the time frames $k \in \{0, 1, \ldots, w-1\}$. In this transient regime, the delay incurred by the time-critical packets depends on the initial backlogs in the queues at frame 0, and the temporal variations in the service received by the queues.

We use $i \in \{1, 2\}$ to index the queues. Let x_i denote the backlog in queue i in frame 0. The initial backlog models the condition where an instantaneous arrival enters the system and one or more packets are already present at the transmission buffers. As discussed in Sec. IV, the transient performance of a time-critical arrival is highly impacted by the presence of initial buffers. Let $A^i(k)$ and $D^i(k)$ denote the cumulative arrivals and departures at queue i, in frame k. For $k = 0$, all the quantities are set to zero. For $k \geq 1$, we define

$$A^1(k) = \sum_{j=0}^{k-1} a_j^1 = y + x_1, \tag{1}$$

$$A^2(k) = D^1(k-1) + x_2, \tag{2}$$

$$D^i(k) = \sum_{j=0}^{k-1} d_j^i, \tag{3}$$

where d_j^i is the number of packets departed queue i in frame j. Eq. (1) describes the cumulative arrivals at the first queue, which correspond to the time critical arrival y and the initial backlog of the first queue x_1. In Eq. (2) a one-step delay is introduced between the reception of a packet and its service at the second queue indicating that packets must be fully received before being relayed. This does not constrain the packet transmission to a single frame and, if multiple transmissions are successful, multiple packets can be transmitted within one frame. In the following, we use $A(k) = A^1(k)$ and $D(k) = D^2(k)$. For analytical simplicity, we assume that a packet received by the controller is processed within the same frame of reception, i.e. processing latencies are negligible, and results in a new packet carrying the feedback information. Sensor messages undergo a conversion at the processing node that changes the data being transmitted in the actuator messages and can assume arbitrary sizes. For analytical simplicity, however, we assume that their size is fixed to a maximum size of B bits and that each packet transmission can be performed in a single slot. This assumption simplifies the problem formulation as different packet sizes affect the experienced Packet Error Rate (PER) of each transmission. For packet sizes that exceed the maximum packet size, multiple transmissions are needed to transmit the information of an arrival. The analysis of the network under variable packet sizes is out of the scope of this paper.

The end-to-end virtual delay, denoted by $W(k)$, is defined as

$$W(k) = \inf\left\{u \geq 1 : A(k) + x_2 \leq D(k+u-1)\right\}. \tag{4}$$

It quantifies the delay faced by the cumulative arrivals till frame $k-1$. Note that, Eq. (4) quantifies the delay in terms of frames, although transmissions are performed on a slot basis.

Arrivals and departures are related via the cumulative service S at each queue according to the input-output relation for a queue with a dynamic server

$$D(k) \geq \min_{0 \leq u \leq k} \left[A(u) + S(k-u)\right]. \tag{5}$$

As discussed in the next section, the effect of departures from both queues and of arrival at the second queue include the effect of packet loss and re-transmissions, which is captured by the service model.

B. Service Model of a Lossy Wireless Network

At the link layer, we consider an error-prone time-slotted system where multiple frequencies can be used for transmission. Packet loss is caused by fading in the received signal, which can arise, for instance, from shadowing, mobility, or external interference. We assume that a frequency diversity mechanism is used in the network and sequential packet transmissions are characterized by uncorrelated channel fades. Whenever critical messages are transmitted via unreliable wireless links, it is a common approach to deploy frequency diversity techniques, such as frequency hopping or frequency scheduling, to avoid sequential packet drops due to correlated channel fades. Thus, we restrict our analysis to the time domain.

We model the random service provided for a single packet transmission as a Bernoulli r.v. according to an average PER of the communication link. That is, a packet is lost with probability p_e and received with probability $1-p_e$. The PER is

[1]We consider frame 0 for notational simplicity; nevertheless, our analysis is equally valid starting the system with any other frame.

determined by the average Signal-to-Noise-and-Interference-Ratio (SINR) which in turn is determined by the combination of the propagation environment and the modulation and coding scheme (MCS) used for transmission. In our model the MCS is not specified as it is assumed that the transmitters use a single MCS and the values of PER solely vary based on the fluctuations of the wireless channel. This assumption corresponds to the situation where simpler transceivers are used and no MCS selection is possible, for instance, as for WirelessHART transmitters. Although the PER values of our model can belong to arbitrary wireless technologies, the numerical PER values used in this work are computed using the BER equation of the IEEE 802.15.4 Std. [28] (cf. Eq. (6)) and assuming absence of coding, thus by modelling a single packet drop as the probability that at least one of its bits is wrong (cf. Eq. (7))

$$\mathrm{BER} = \frac{1}{30}\sum_{u=2}^{16}(-1)^u\binom{16}{u}e^{-20\,\mathrm{SINR}\left(1-\frac{1}{u}\right)}, \tag{6}$$

$$\mathrm{PER} = 1-(1-\mathrm{BER})^B\,. \tag{7}$$

Each frame comprises of N time slots to be shared between the transmissions of packets from the two queues in the uplink and the downlink, cf. Fig. 2. In frame k, let n_k^1 and $n_k^2 = N - n_k^1$ denote the slots used for transmitting the packets from the first queue and the second queue, respectively. Given this frame allocation, the service offered by the i-th link at frame k is distributed as a Binomial r.v. given by

$$b_k^i(n_k^i) \sim \mathcal{B}\left(n_k^i, 1-p_e\right). \tag{8}$$

The cumulative service provided on the link at queue i in k consecutive frames is equal to a summation of Binomial random variables with parameters $1-p_e$, which is also a Binomial r.v. given by

$$S^i(k) = \sum_{j=0}^{k-1} b_j^i(n_j^i) \sim \mathcal{B}\left(\sum_{j=0}^{k-1} n_j^i, 1-p_e\right). \tag{9}$$

The cumulative service of Eq. (9) directly models the number of packets successfully received in the presence of random packet loss and re-transmission. In this way, these effects are included in the cumulative departure of Eq. (5) and the end-to-end virtual delay of the network of Eq. (4) which, as discussed in the next section and derived in Sec. IV, determine the DVP of time critical packets. Furthermore, from Eq. (7) we observe that the maximum packet size B directly determines the transmission PER, thus the cumulative service performance, which in turn affects the end-to-end virtual delay of the network.

C. Problem Statement

We are interested in optimizing the dynamic service offered by the wireless links of sensor and controller to minimize the end-to-end delay of a time-critical arrival while it traverses the network. In particular, in order to investigate scheduling policies that exploit initial network conditions, we study the impact of queue state information on the achievable performance of semi-static and dynamic resource allocations.

We define a scheduling policy π as the allocation of time slots to both queues in every frame until the deadline, i.e. $\pi \triangleq \mathbf{n}^1 = \{n_0^1, n_1^1, \ldots, n_{w-1}^1\}$, equivalently $\pi \triangleq \mathbf{n}^2 = N - \mathbf{n}^1$. Different scheduling algorithms are computed based on the queue state information $\mathbf{q}_k = (q_k^1, q_k^2)$, where q_k^1 and q_k^2 denote the lengths of first and second queues in frame k, respectively.

In the following, we consider scheduling policies that compute semi-static and dynamic resource allocations. On the one hand, a semi-static scheduling policy, denoted by π_S, computes a schedule based on the initial state $\mathbf{q}_0$. Semi-static policies can be applied, for instance, to resource-constrained wireless networks such as WSN, where updating the network allocation over time is difficult due to the availability of a single radio interface and unreliable feedback channels. On the other hand, a dynamic scheduling policy, denoted by $\pi_D(\mathbf{q}_k)$, relies on the availability of the queue state $\mathbf{q}_k$ at a centralized network logic, which is used to determine the allocation of slots for the next frame, i.e. at k-th frame $n_k^1 = \pi_D(\mathbf{q}_k)$. An exemplary application of dynamic policies is in cellular networks, where reliable feedback channels can timely deliver new queue states to the network coordinator and resource allocations to the devices.

Given the end-to-end deadline w, we define the Delay Violation Probability (DVP) of a sequence of time-critical packets that arrived in frame 0 as the probability that one or more packets of the sequence do not depart the second queue by the end of frame w. For initial backlogs x_1, x_2 this is denoted by $\mathrm{DVP}\,(w, y, x_1, x_2)$ and is given by

$$\mathrm{DVP}\,(w, y, x_1, x_2) := \mathbb{P}\left\{W(1) > w\right\}. \tag{10}$$

The above equivalence is obtained using Eq. (4), where the event $\{W(1) > w\}$ implies that the cumulative departures by the end of frame w are smaller than the total number of packets in frame 0. Note that DVP could potentially be used as QoS in networked feedback systems; for example, given a deadline of w frames, DVP represents the probability that a packet (carrying control command) in response to a packet generated by the sensor is delivered to the actuator within the deadline.

We are interested in finding semi-static and dynamic scheduling policies that minimize the DVP of packets belonging to interactive applications. The policies are obtained by formulating and solving the following optimization problems. Let Π_S and Π_D denote the sets of all possible semi-static and dynamic scheduling policies[2]. Given y application packets arrived in frame 0, an optimal semi-static policy π_S and a dynamic scheduling policy π_D are obtained solving

$$\underset{\pi_S \in \Pi_S}{\text{minimize}} \quad \mathrm{DVP}\,(w, y, x_1, x_2)\,, \tag{11}$$

$$\underset{\pi_D \in \Pi_D}{\text{minimize}} \quad \mathrm{DVP}\,(w, y, x_1, x_2)\,. \tag{12}$$

[2]These sets are non-empty; an example policy allocates slots equally to both links in all frames.

In Eq. (11) and Eq. (12), Π_S, Π_D denote the sets of all possible semi-static and dynamic scheduling policies, and are non-empty as each resulting slot allocation is valid.

In the sequel, we will be using the following definitions. Given a set of events $E_1, E_2, \ldots$ the *union bound* is given by

$$\mathbb{P}\left\{\bigcup_i E_i\right\} \leq \sum_i \mathbb{P}\left\{E_i\right\}.$$

Furthermore, given a random variable X, the *Chernoff bound* is given by

$$\mathbb{P}\left\{X \geq x\right\} \leq \min_{t>0} \frac{\mathbb{E}[e^{tX}]}{e^{tx}},$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator.

IV. Derivation of Delay Violation Probability

We characterize DVP using Stochastic Network Calculus (SNC) [29]. From Eq. (4), DVP can be obtained in terms of the virtual delay of the network

$$\begin{aligned}\mathrm{DVP}\left(w, y, x_1, x_2\right) &= \mathbb{P}\left\{W(1) > w\right\} \\ &= \mathbb{P}\left\{D(w) < y + x_1 + x_2\right\}.\end{aligned} \tag{13}$$

Using the relationship between arrivals, service and departures as defined in Eq. (5) an analytical expression for DVP is obtained.

Proposition 1. *The delay violation probability (DVP) of a time critical arrival of y packets at $k = 0$, given initial queue backlogs x_1, x_2 is*

$$\begin{aligned}&\mathrm{DVP}(w, y, x_1, x_2) = \\ &\mathbb{P}\Big\{\left\{S^2(w) < y + x_1 + x_2\right\} \cup \left\{S^2(w-1) < y + x_1\right\} \cup \\ &\bigcup_{u=2}^{w} \left\{S^2(w-u) + S^1(u-1) < y + x_1\right\}\Big\}.\end{aligned} \tag{14}$$

Proof. The proof can be found in Appendix A ☐

Eq. (14) shows that the initial backlogs at the two queues highly impacts the values of DVP. This is intuitive, because x_2 packets must be successfully delivered by the second transmitter and x_1 by both the transmitters before a time-critical arrival can be successfully received. This impact is higher when the time-critical arrival is smaller or comparable to the amount of initial buffer.

From Eq. (14), we observe that the calculation of DVP is highly non-trivial and we argue that it is likely intractable. Note that the RHS in (14) involves the computation of the distribution of sum of *dependent* binomial random variables. Note that the results for the distribution of the sum of dependent random variables are scarce due to the complexity introduced by the dependence, and to the best of our knowledge, there are no known results on computing the sum of dependent binomial random variables. Even for the case of *independent* binomial random variables, computing approximations for the distribution of the sum is still an active research field [30], [31].

Furthermore, the DVP computation requires the knowledge of future, i.e. both the allocations n_k^1 and the resulting queue states, in order to calculate the cumulative services. Thus, it is impossible to use DVP to obtain a scheduling policy which causally allocates the time slots at each frame within the deadline using only the past information.

In this work, we address this issue by deriving upper bounds for DVP which are then used to design semi-static and dynamic schedulers. In Sec. V, we investigate *semi-static* scheduling policies based on two upper bounds of Eq. (14) to determine the allocation of slots until the deadline solely relying on the initial queue states. Then, in Sec. VI, a *dynamic* scheduling policy is derived based on another upper bound for DVP that reallocates the network resources according to the changes in the queue states.

V. Semi-static Scheduling Policies

In this section, we derive an upper bound for DVP, referred to as DVPUB, and formulate an upper bound minimization problem, which is then used to compute the proposed semi-static policies.

Using the union bound for DVP in (14), we obtain DVPUB, given by

$$\begin{aligned}&\mathrm{DVPUB}(w, y, x_1, x_2) = \mathbb{P}\left\{S^2(1+w) < y + x_1 + x_2\right\} + \\ &\sum_{u=1}^{1+w} \mathbb{P}\left\{S^2(1+w-u) + S^1(u-1) < y + x_1\right\} \\ &= \mathbb{P}\left\{S^2(1+w) \leq y + x_1 + x_2 - 1\right\} + \\ &\sum_{u=1}^{1+w} \mathbb{P}\left\{S^2(1+w-u) + S^1(u-1) \leq y + x_1 - 1\right\}.\end{aligned} \tag{15}$$

Applying Eq. (9) to Eq. (15), the DVPUB resulting from the allocation of $\mathbf{n}^1 = \{n_0^1, \ldots, n_w^1 - 1\}$ slots at the second transmitter and $\mathbf{n}^2 = N - \mathbf{n}^1$ slots at the first one is given by

$$\begin{aligned}&\mathrm{DVPUB}(w, y, x_1, x_2, \mathbf{n}^2) = \sum_{x=0}^{y+x_1+x_2-1} \mathbb{P}\left\{\sum_{i=0}^{w} b_i^2\left(n_i^2\right) = x\right\} \\ &+ \sum_{u=1}^{1+w} \sum_{z=0}^{y+x_1-1} \mathbb{P}\left\{\sum_{j=0}^{w-u} b_j^2\left(n_j^2\right) + \sum_{k=0}^{u-2} b_k^1\left(n_k^1\right) = z\right\} \\ &\overset{\text{(a)}}{=} \sum_{x=0}^{y+x_1+x_2-1} \left(\frac{1}{p_e} - 1\right)^x \binom{\sum_{i=0}^{w} n_i^2}{x} p_e^{\sum_{i=0}^{w} n_i^2} \\ &+ \sum_{u=1}^{1+w} \sum_{z=0}^{y+x_1-1} \left(\frac{1}{p_e} - 1\right)^z p_e^{\sum_{j=0}^{w-u} n_j^2 - \sum_{k=0}^{u-2} n_k^2 + (u-1)N} \cdot \\ &\binom{\sum_{j=0}^{w-u} n_i^2 - \sum_{k=0}^{u-2} n_j^2 + (u-1)N}{z}.\end{aligned} \tag{16}$$

In step (a), we used the fact that $S^i(k)$ is distributed as a Binomial r.v. as shown in Eq. (9).

We are interested in minimizing the DVPUB to find semi-static scheduling policies. However, this is highly non trivial for different reasons. Minimizing DVPUB is a combinatorial problem and finding a heuristic solution by relaxing the domain of $\mathbf{n}^2$ is challenging as DVPUB consists of the sum of several binomial coefficients. Thus, it is highly non trivial to study its convexity. Therefore, following a similar approach as in [7] a looser convex bound, referred to as Wireless Transient

Bound (WTB), is obtained by applying the *Chernoff bound* to Eq. (15).

$$\mathrm{WTB}(w,y,x_1,x_2) = \min_{s>0}\Big\{\mathbb{E}\left[e^{-s\,S^2(1+w)}\right]e^{s\,(y+x_1+x_2-1)}+ \sum_{u=1}^{1+w}\mathbb{E}\left[e^{-s\left[S^1(u-1)+S^2(1+w-u)\right]}\right]e^{s\,(y+x_1-1)}\Big\}. \quad (17)$$

The calculation of WTB for a wireless transmitter is obtained by computing the Mellin transform of the cumulative service of Eq. (9), given by

$$\begin{aligned}\mathbb{E}\left[e^{-s\,S(m,\mathbf{n})}\right] &= \mathbb{E}\left[e^{-s\,\sum_{i=0}^{-1+\sum_{j=0}^{m-1}n_j} b_i}\right]\\ &= \mathbb{E}\left[\left(e^{-s\,b_i}\right)^{\sum_{i=0}^{m-1}n_i}\right]\\ &= \left[(1-p_e)e^{-s}+p_e\right]^{\sum_{i=0}^{m-1}n_i}, \end{aligned} \quad (18)$$

where b_i are i.i.d. Bernoulli random variables for transmission outcomes. Finally, combining Eq. (17) and (18), we obtain

$$\mathrm{WTB}(w,y,x_1,x_2,\mathbf{n}^2) = \min_{s>0}\Big\{\left[(1-p_e)e^{-s}+p_e\right]^{\sum_{i=0}^{w}n_i^2}e^{s\,(y+x_1+x_2-1)}+ \sum_{u=1}^{1+w}\left[(1-p_e)e^{-s}+p_e\right]^{(u-1)N-\sum_{j=0}^{u-2}n_j^2+\sum_{i=0}^{w-u}n_i^2}e^{s\,(y+x_1-1)}\Big\}. \quad (19)$$

Analytical solutions for Eq. (11) require closed-form expressions for DVP, which, as discussed in Sec. IV and earlier in this section, is not likely tractable. This implies computing the optimal solutions for Eq. (11) is also not likely tractable. Therefore, we rely on the upper bounds for DVP for solving Eq. (11).

Given the initial queue backlogs $\mathbf{q}_0 = \{x_1, x_2\}$, we aim to solve the upper bound minimization problem below

$$\underset{\mathbf{n}^2\in\{1,\ldots,N-1\}^w}{\arg\min}\ \mathrm{WTB}(w,y,x_1,x_2,\mathbf{n}^2). \quad (20)$$

To solve the integer-programming problem (20), one may employ exact algorithms (such as *branch-and-bound*), which however has run-time that scales exponentially with N and w. Instead, we relax the integer constraints, show that the relaxed problem is convex, and use different methods to round the continuous values of the solution and obtain multiple heuristics for the minimization of DVPUB.

Let $\widetilde{\pi}_S^\star$ denote the optimal solution for the relaxed problem of (20) which is given by

$$\widetilde{\pi}_S^\star = \underset{\mathbf{n}^2\in[1,N-1]^w}{\arg\min}\ \mathrm{WTB}(w,y,x_1,x_2,\mathbf{n}^2). \quad (21)$$

Lemma 1. *The optimization problem in* (21) *is convex.*

Proof. The proof is given in Appendix B. □

Thanks to Lemma 1, well-known convex optimization algorithms, such as the subgradient or interior-point can be used, which provide scalability for an increasing number of slots N and frames within the deadline w. In this work, (21) is solved using the nonlinear programming solver *fmincon* available in Matlab™ employing the sequential quadratic programming (SQP) algorithm. Once $\widetilde{\pi}_S^\star$ is found, a conversion to the integer domain is needed in order to determine a feasible solution, which we refer by $\hat{\pi}_S^\star$. Although the optimal selection of an integer solution would require the exploration of the entire problem's domain, heuristic methods can be applied to find a solution in the neighbourhood of $\widetilde{\pi}_S^\star$. To this end, we investigate different neighbour search methods in order to achieve near-optimal performances.

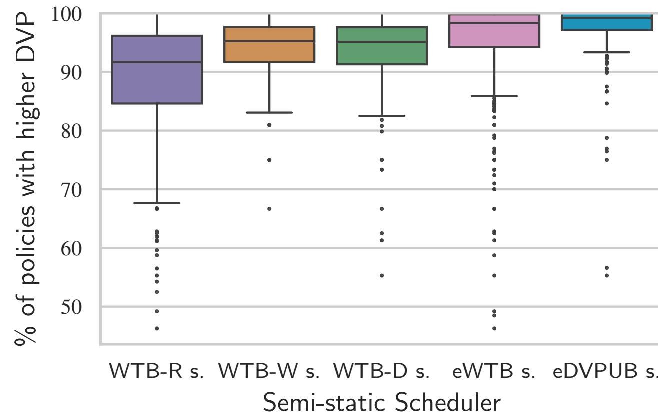

Fig. 3. Performance of the proposed semi-static schedulers for different system parameters. For each scheduler, the percentage of existing policies that achieve higher DVP is evaluated.

The simplest way to derive $\hat{\pi}_S^\star$ is to round each frame allocation of $\widetilde{\pi}_S^\star$ to its closest integer value. We refer to this method as WTB-R. Alternatively, a heuristic policy can be found as follows. For each frame allocation $n_i^1 \in \widetilde{\pi}_S^\star$, two integer values are derived applying the floor and ceiling functions. A search space is constructed by computing all the combinations of the integer values for each frame until the deadline which leads to a total of 2^w combinations. As final step, Eq. (16) and (19) are used to evaluate each combination and identify the best one. The semi-static policies corresponding to the evaluation of Eq. (16) and (19) over this search space are referred to as WTB-D and WTB-W, respectively.

We evaluate the performance of the different heuristics algorithms based on WTB-R, WTB-D and WTB-W of the relaxed problem in Eq. (21) by performing extensive simulations over a broad range of values for each system parameter, i.e. for $x_1, x_2 \in \{0,1,2\}$, $w \in \{2,3,4,5,6\}$, $N \in \{2,3,4,5\}$, and $p_e \in \{0.2, 0.33, 0.4, 0.5\}$. The performance gap of the heuristics with respect to the optimization of the DVPUB and WTB upper bounds is evaluated by means of two additional schedulers based on eDVPUB and eWTB, which exhaustively explore the problem's domain to find the policies that respectively achieve the minimum values of DVPUB, cf. Eq. (16), and WTB, cf. Eq. (19). Furthermore, to evaluate the performance gap with respect to the solutions of the original problem in Eq. (11), the heuristics are compared with the performance of the policy that achieves the minimum DVP. This optimal policy is computed by enumeration. That is, we computed all the finite policies in a given set and evaluated the resulting DVP via simulations and stored the element with lowest DVP.

Fig. 3 shows the performance of the proposed semi-static schedulers by computing, for each scheduler and system configuration, the percentage of feasible policies that achieve higher DVP. As expected, the exhaustive search methods eWTB and eDVPUB achieve the best results, with eDVPUB finding, in the large majority of cases, the top 10% semi-static policies. The performance gap between eWTB and eDVPUB is introduced by the Chernoff bound in Eq. (18). These methods, however, do not represent a feasible way of computing policies as they require the exhaustive exploration of the entire problem domain. Differently, the heuristic methods WTB-R/W/D can be used to efficiently find semi-static policies for arbitrary parameters. The low complex WTB-R achieves the worst performance. WTB-W and WTB-D achieve similar performances and show a small performance penalty with respect to the exhaustive methods, with WTB-W achieving better performances. This can be explained by the fact that, in WTB-D, Eq. (19) is used to solve the relaxed problem, while Eq. (16) to find the integer solution. For this reason, only WTB-W is presented in the numerical section.

VI. Dynamic Scheduling Policy

As shown in Eq. (14) and discussed in Sec. IV, the computation of DVP requires the allocation of slots and corresponding realization of the service at both the links in future time slots. Therefore, it cannot be used directly to compute dynamic policies for Eq. (12). Note that, we also cannot use WTB as it requires the allocation for all w slots. In such cases, one may resort to exhaustive search for finding an optimal policy. However, this approach is not scalable as one has to enumerate all possible channel realizations for all w frames. Therefore, to address this problem, we resort to a heuristic solution that maximizes the number of departures in w. In the following, we first obtain an upper bound for DVP using Markov's inequality:

$$\begin{aligned}\mathrm{DVP}(w,y,x_1,x_2) &= \mathbb{P}\{D(w) < y+x_1+x_2\}\\ &= \mathbb{P}\{D(w) \le y+x_1+x_2-1\}\\ &= \mathbb{P}\{1/D(w) \ge 1/(y+x_1+x_2-1)\}\\ &\le (y+x_1+x_2)\mathbb{E}[1/D(w)]. \qquad (22)\end{aligned}$$

From Eq. (22) we infer that minimizing the expectation of the inverse of the cumulative departures (throughput) of the network minimizes the upper bound of the DVP and thus potentially reduces DVP. Using this insight, in the following, we compute a heuristic schedule by solving the expected throughput maximization problem stated below:

$$\underset{\pi_D\in\Pi_D}{\text{maximize}} \quad \mathbb{E}[D(w)] = \sum_{k=0}^{w-1} \mathbb{E}\left[d_k^i\right]. \qquad (23)$$

In order to solve the optimization problem in Eq. (23), we formulate a discrete-time, finite-horizon MDP. We use $\mathbf{q}_k$ to denote the state of the system and n_k^1 to denote the action in frame k. The maximum number of slots in a frame is N and therefore $n_k^1 \in \{0,1,\dots,N\}$. Given n_k^1, from (6) we have

$$\begin{aligned}\mathbb{P}\{s_k^1 = r\} &= \binom{n_k^1}{r}(1-p_e)^r p_e^{n_k^1-r},\\ \mathbb{P}\{s_k^2 = r\} &= \binom{N-n_k^1}{r}(1-p_e)^r p_e^{N-n_k^1-r}.\end{aligned}$$

The queues evolve as below:

$$q_{k+1}^1 = \max(q_k^1 - s_k^1, 0), \qquad (24)$$
$$q_{k+1}^2 = \max(q_k^2 - s_k^2, 0) + \min(q_k^1, s_k^1). \qquad (25)$$

Note that the number of departures from the first queue in frame k equals $\min(q_k^1, s_k^1)$, which are added to the second queue to be served in frame $k+1$.

In the following, we formulate the transition probabilities for the states. Note that the initial backlogs in the queues are $(y+x_1, x_2)$, where y is due to the message of interest. We have $q_0^1 = y+x_1$ and $q_0^2 = x_2$. We now analyse the set of possible states in our system. In any frame k, a feasible state (q_k^1, q_k^2) should satisfy the following conditions:

$$q_k^1 \le q_{k-1}^1, \qquad (26)$$
$$q_k^1 + q_k^2 \le q_{k-1}^1 + q_{k-1}^2. \qquad (27)$$

Conditions Eq. (26) and Eq. (27) follow from the fact that we ignore arrivals after the message of interest and in every frame each queue will receive certain service. Note that while the length of the first queue can only decrease as the packets are served, the length of the second queue may increase up to $y+x_1+x_2$ as departures from first queue are added to the second queue. Therefore, for every state $\mathbf{q}_k$ in the state space, say $\mathcal{Q}$, $q_k^1 \in \{0,1,\dots,y+x_1\}$ and $q_k^2 \in \{0,1,\dots,y+x_1+x_2\}$. This implies that $\mathcal{Q}$ can contain at most $(y+x_1+1)(y+x_1+x_2+1)$ possible states.

Consider that in frame k, $q_k^1 = l^1$ and $q_k^2 = l^2$. We would like to present the transition probabilities to the states $q_{k+1}^1 = l_+^1$ and $q_{k+1}^2 = l_+^2$. We have the following cases.

Case 1: $l_+^1 > l^1$ or $l_+^1 + l_+^2 > l^1 + l^2$. From Eq. (26) and Eq. (27), we infer that

$$\mathbb{P}\{q_{k+1}^1 = l_+^1, q_{k+1}^2 = l_+^2 | q_k^1 = l^1, q_k^2 = l^2\} = 0.$$

Case 2: $0 < l_+^1 \le l^1$, $0 < l_+^2$, and $l_+^1 + l_+^2 \le l^1 + l^2$. In this case $s_k^1 < q_k^1 = l^1$ and $s_k^2 < q_k^2 = l^2$. From Eq. (24) we have

$$q_{k+1}^1 = q_k^1 - s_k^1 \quad \Rightarrow \quad s_k^1 = l^1 - l_+^1.$$

The number of packets served from the second queue are computed from Eq. (25).

$$q_{k+1}^2 = q_k^2 - s_k^2 + s_k^1 \quad \Rightarrow s_k^2 = l^2 - l_+^2 + l^1 - l_+^1.$$

Therefore,

$$\begin{aligned}&\mathbb{P}\{q_{k+1}^1 = l_+^1, q_{k+1}^2 = l_+^2 | q_k^1 = l^1, q_k^2 = l^2\}\\ &\qquad = \mathbb{P}\{s_k^1 = l^1 - l_+^1, s_k^2 = l^2 - l_+^2 + l^1 - l_+^1\}.\end{aligned}$$

Case 3: $l_+^1 = 0$, $0 < l_+^2$, and $l_+^2 \leq l^1 + l^2$. In this case all l^1 packets from the first queue are served. This implies $s_k^1 \geq q^1 = l^1$. Using similar analysis as above, we obtain

$$\mathbb{P}\{q_{k+1}^1 = 0, q_{k+1}^2 = l_+^2 | q_k^1 = l^1, q_k^2 = l^2\} \\ = \mathbb{P}\{s_k^1 \geq l^1, s_k^2 = l^2 - l_+^2 + l^1\}.$$

Case 4: $l_+^1 = l^1$, $l_+^2 = 0$. In this case we have $s_k^1 = 0$, and all l^2 packets from the second queue are served, i.e. $s_k^2 \geq q^2 = l^2$. From Eq. (25), we have

$$\mathbb{P}\{q_{k+1}^1 = l^1, q_{k+1}^2 = 0 | q_k^1 = l^1, q_k^2 = l^2\} \\ = \mathbb{P}\{s_k^1 = 0, s_k^2 \geq l^2\}.$$

Note that the case $0 \leq l_+^1 < l^1$ and $l_+^2 = 0$ cannot happen as $l^1 - l_+^1$ packets will be added to the second queue in the current slot. All the above cases are written assuming that $l^1 > 0$ and $l^2 > 0$. If either of them is zero, then the transition probability only involves the probability for service received by the non-empty queue.

Given the initial state $\mathbf{q}_0 = (y + x_1, x_2)$, we are interested in finding a scheduling policy $\pi_D^\star$ that solves the maximization problem of Eq. (23). For this, we define the reward r_k of a policy π_D for a given state $\mathbf{q}_k$ as the expected number of departures from the system, i.e. the expected number of packets that are served at the second queue under this policy, and is given by

$$r_k(\mathbf{q}_k, \pi_D(\mathbf{q}_k)) = \mathbb{E}\left[d_k^i | \, \pi_D(\mathbf{q}_k)\right] \\ = \mathbb{E}\left[\min\left(q_k^2, s_k^2\right) | \, \pi_D(\mathbf{q}_k)\right]. \quad (28)$$

The total reward, obtained evaluating Eq. (28) over a horizon of w frames, is equal to Eq. (23). Therefore, the objective of the MDP is equal to the objective of Eq. (23).

Value iteration algorithms solve the MDP optimization recursively computing a value function J based on the Bellman's equation [32]. The optimal value function $J(\mathbf{q}_k)$ given a state $\mathbf{q}_k$ is

$$J_k(\mathbf{q}_k) = \underset{\pi_D \in \Pi_D}{\text{maximize}} \;\; r_k(\mathbf{q}_k, \pi_D(\mathbf{q}_k)) + \\ \sum_{\mathbf{q}_{k+1} \in \mathcal{Q}_{k+1}} \mathbb{P}\{\mathbf{q}_{k+1} | \mathbf{q}_k, \pi_D(\mathbf{q}_k)\} J_{k+1}(\mathbf{q}_{k+1}), \quad (29)$$

where $\mathbb{P}\{\mathbf{q}_{k+1} | \mathbf{q}_k \pi_D(\mathbf{q}_k)\}$ is the transition probability from state $\mathbf{q}_k$ to state $\mathbf{q}_{k+1}$ in one time step using $\pi_D(\mathbf{q}_k)$, $\mathcal{Q}_{k+1}$ denotes the set of all states reachable from $\mathbf{q}_k$ with one time step transition.

By the construct of the MDP, it is easy to see that $\pi_D^\star$ is optimal for Eq. (23) which is stated in the following corollary.

Corollary 1. *$\pi_D^\star$ is throughput optimal, i.e.*

$$\pi_D^\star = \underset{\pi_D \in \Pi_D}{\arg\max} \quad \mathbb{E}\left[D(w)\right].$$

For a finite number of states and actions, the optimal policy $\pi_D^\star$ for the MDP can be found by computing the optimal value function in Eq. (29) by backward recursion (cf. [32]). The calculation of the optimal dynamic scheduling policy satisfying Corollary 1 can be performed using the finite-horizon value iteration algorithm. For each epoch k until the deadline w and state $\mathbf{q}_k$, the value function in Eq. (29) is computed for all possible actions. Therefore, computing the optimal dynamic policy requires $\mathcal{O}\left((N+1)\, |\mathcal{Q}| w\right)$ operations. We note that optimal actions for all the states in $\mathcal{Q}$ can be computed a priori and stored in a table, and for a queue state observed in a frame the corresponding optimal action can be retrieved from the table.

VII. Performance Evaluation of Semi-static and Dynamic Scheduling Policies

In this section, the performance of semi-static and dynamic scheduling policies are evaluated numerically. The evaluations leverage C, Matlab™ and python to compute the performance of, respectively, semi-static and dynamic policies and comparison baselines. We evaluate the schemes for variations in several main system parameters: (1) Different initial queue backlogs x_1, x_2; (2) Application deadlines w; (3) Number of slots per frame N; and (4) Average service PERs p_e. The simulated service PERs are computed following Eqs. (6) and (7) for a transmission power of -20dBm and an indoor multipath propagation model typical of industrial environments with propagation coefficient equal to 2.2 and multipath variance 6 [33]. The PER values of 0.2 and 0.4 are achieved at the transmission distances of 34.5m and 36.1m.

In Sec. VII-A, we present first a performance comparison of purely semi-static scheduling policies. In Sec. VII-B, we then move to purely dynamic policies. Finally, in Sec. VII-C, we present results on the performance gap between the proposed semi-static and dynamic scheduling policies.

A. Semi-static Scheduling Policies

As discussed, semi-static scheduling policies in this work take the initial queue state at the moment of the arrival of a time-critical packet into account. The semi-static schedule is then determined prior to the transmission of the corresponding packet and defines the allocation of slots until the deadline of the packet. We compare the DVP achieved by the proposed WTB-W scheduler, cf. Sec. V, with the exhaustive search methods eDVPUB and Optimum, which compute semi-static policies, respectively, evaluating Eq. (11) and via simulations. Due to its high complexity, the performances of the optimal policy is shown for smaller parameter sets. Exhaustive schemes are shown to evaluate the performance of WTB-B and do not represent a feasible way of computing semi-static scheduling policies. The last comparison scheme is a agnostic allocation of half of the slots for uplink and half of the slots for downlink transmission. In the case of an odd number of slots, one extra slot is allocated to the first link. We refer to this scheme as 50/50. Throughout the evaluation of the semi-static schemes, we keep the channel error rate p_e at 0.2.

Fig. 4 shows the DVP for different application deadlines, backlogs, while the frame configuration is fixed with $N = 4$. WTB-W and eDVPUB achieve close-to-optimal DVP for all backlog sizes and deadlines. Thus, taking the initial queue states into account is beneficial and provides up to one order of magnitude improvement compared to the agnostic 50/50 scheme. As the deadline w increases, the performance gap of the proposed polices in comparison to the 50/50 scheme

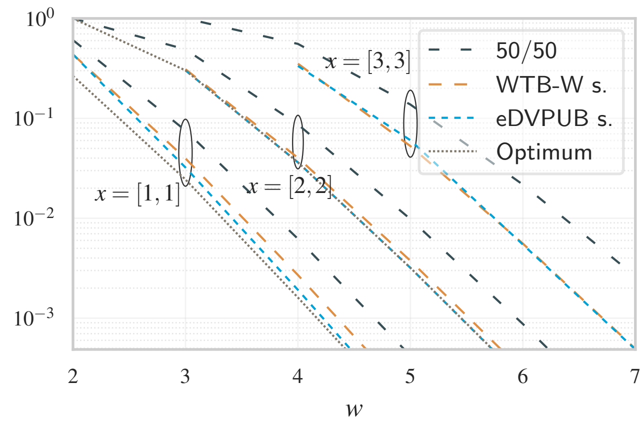

Fig. 4. DVP achieved by semi-static schedulers for different deadlines w, increasing backlogs x_1, x_2, $N = 4$, $p_e = 0.2$.

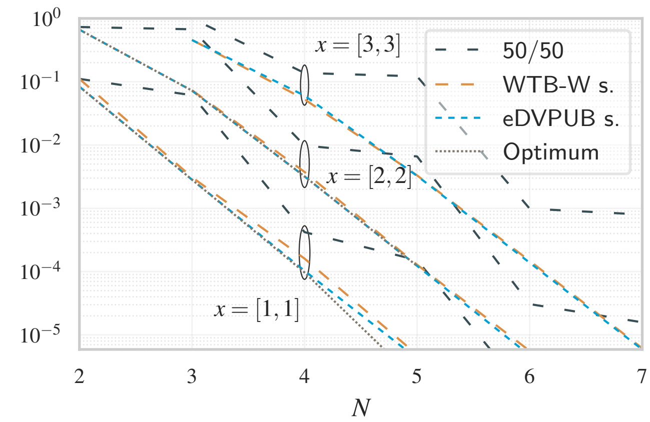

Fig. 5. DVP achieved by semi-static schedulers for different frame sizes N, increasing backlogs x_1, x_2, $w = 5$, $p_e = 0.2$.

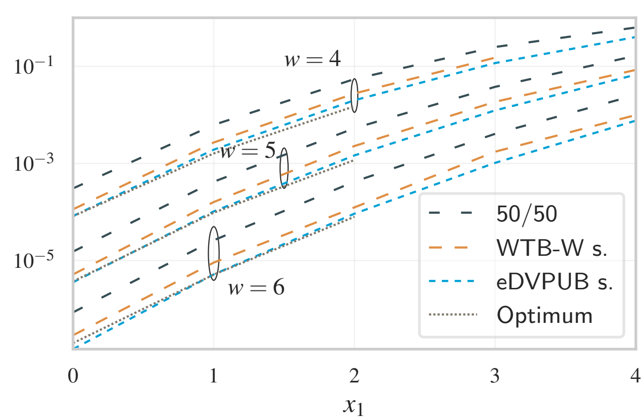

Fig. 6. DVP achieved by semi-static schedulers for different backlogs x_1 and deadlines w, $x_2 = 1$, $N = 4$, $p_e = 0.2$.

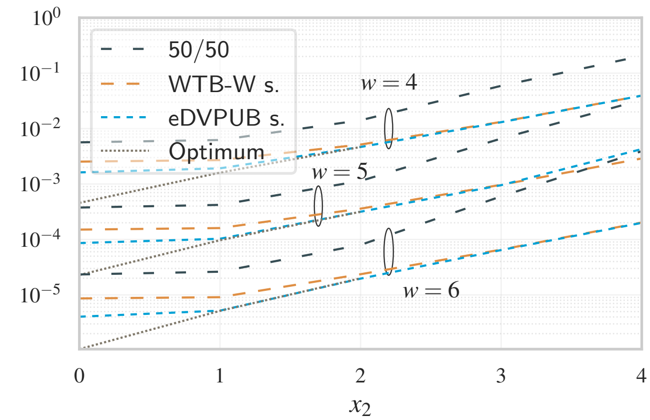

Fig. 7. DVP achieved by semi-static schedulers for different backlogs x_2 and deadlines w, $x_1 = 1$, $N = 4$, $p_e = 0.2$.

increases. For backlogs x_1 and x_2 equaling 1, eDVPUB provides a minor improvement in DVP with respect to WTB-W at the expense of higher computational complexity, while for higher backlogs, the performance difference between the two is negligible.

In Fig. 5 we study the impact of different frame lengths, backlogs, while keeping the deadline fixed with $w = 5$. Again, the improvement in DVP of the proposed semi-static schedulers with respect to the agnostic 50/50 scheme is confirmed. The "stepped" behaviour of the 50/50 scheme is caused by the different ratios of allocated slots to the two links for even and odd frame lengths. Otherwise, the results show a minor difference between WTB-W and eDVPUB, while the optimality gap increases slightly for increasing N when x_1 and x_2 are equal to 1.

Fig. 6 and Fig. 7 show the impact of initial backlog on the DVP for different deadlines and a fixed frame configuration with $N = 4$. Increasing x_1 has a stronger impact than x_2 on the achievable DVP. This is intuitive as packets backlogged in x_1 must be sent by both links. We note that the gap between the proposed semi-static schedulers and the agnostic 50/50 scheme increases with an increasing backlog x_2. In both figures, we observe again that exploiting the initial queue states leads to a gain of approximately half order of magnitude with respect to the agnostic 50/50 scheme with increasing initial backlogs. As shown in Fig. 6, eDVPUB achieves near-optimal DVP and the performance gap with respect to WTB-W is constant with increasing x_1. In contrast, in Fig. 7, the gap between the proposed schedulers and the optimal policy decreases with increasing x_2, achieving close-to-optimal DVP.

B. *Dynamic Scheduling Policies*

In contrast to the semi-static schemes, dynamic schedulers benefit from feedback on the queue states as time evolves, giving them the opportunity to reallocate time slots depending on the queue's backlog evolution. This makes dynamic schedulers causing more overhead and complexity, with the advantage of potentially achieving a higher performance. Initially, we are only turning to different dynamic schedulers in this section. Concretely, we consider the following schemes:

- MDP: Our proposed dynamic scheduler from Section VI.
- Max Weight (MW): Under MW, all slots are allocated to the link with the maximum backlog [9].
- Weighted-Fair Queuing (WFQ): Under WFQ, slots are allocated to the links in proportion to the ratio between their queue sizes [10].
- Backpressure (BP): Under BP, slots are allocated to the link with maximum backpressure, where the backpressure

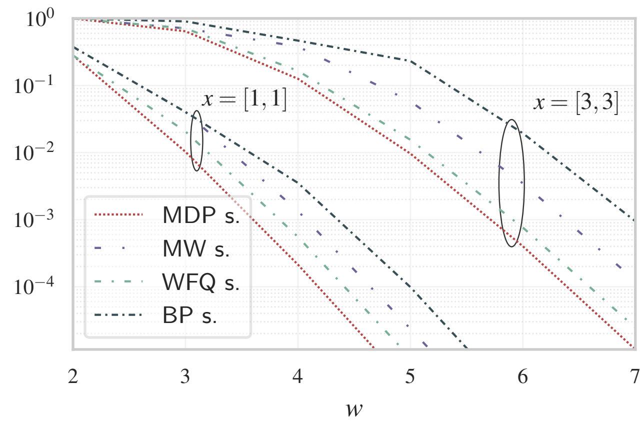

Fig. 8. DVP achieved by dynamic schedulers for different deadlines w, increasing backlogs x_1, x_2, $N = 6$, $p_e = 0.4$.

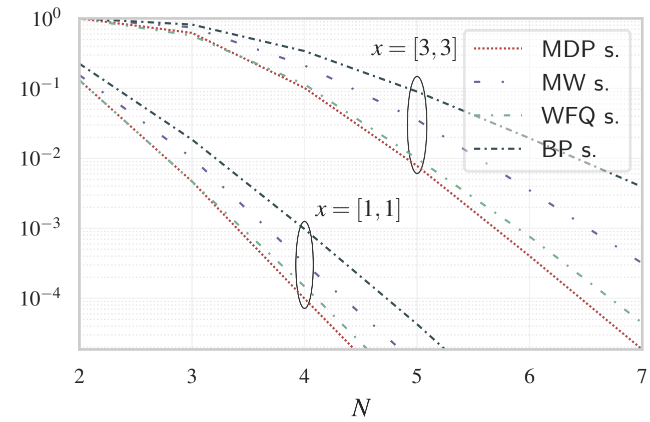

Fig. 9. DVP achieved by dynamic schedulers for different frame sizes N, increasing backlogs x_1, x_2, $w = 6$, $p_e = 0.4$.

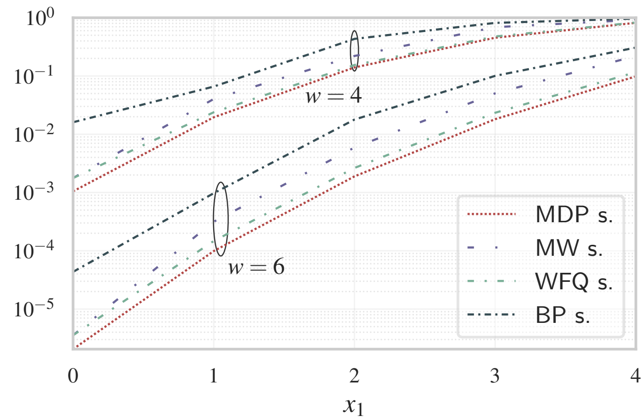

Fig. 10. DVP achieved by dynamic schedulers for different backlogs x_1 and deadlines w, $x_2 = 1$, $N = 4$, $p_e = 0.4$.

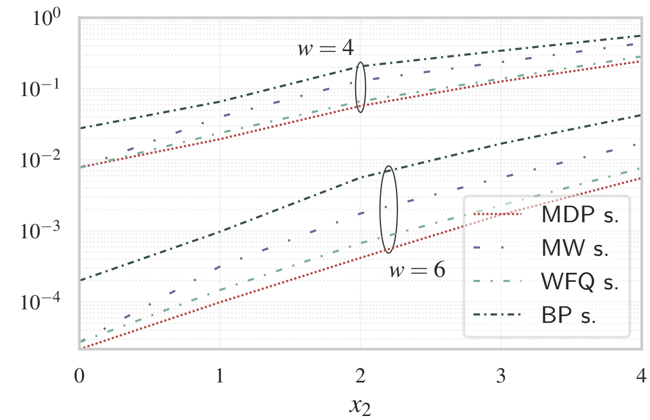

Fig. 11. DVP achieved by dynamic schedulers for different backlogs x_2 and deadlines w, $x_1 = 1$, $N = 4$, $p_e = 0.4$.

at the first link is equal to $x_1 - x_2$ while at the second link it is equal to x_2 [8].

In all the figures below, the value of p_e is set to 0.4.

Fig. 8 shows the DVP achieved by the dynamic schedulers for different application deadlines, backlogs, and a fixed frame configuration with $N = 6$. We observe that the proposed MDP-based scheduler outperforms all the other methods. BP and MW achieve higher DVP as they allocate all the slots to a single link in each frame. Similar to MDP, WFQ allows a granular allocation of slots to the links and achieves the smallest performance gap.

Fig. 9 presents DVP achieved by different policies for different frame lengths, backlogs, and a fixed deadline $w = 6$. Again, MDP achieves lower DVP while the performances of the other schemes are in line with the previous scenario. Furthermore, the performance advantage of MDP, with respect to the comparison schemes, increases with increasing frame length. This advantage arises from the fact that the action space of MDP is larger for large N, which results in more accurate slot allocations.

Fig. 10 and Fig. 11 show the impact of the initial backlogs on the DVP achieved by different policies for different deadlines and with a fixed frame configuration $N = 4$. Again, increasing x_1 has a higher impact on DVP compared to x_2. Due to the allocation of all slots to a link in a frame, the performance gap of BP and MW increases as x_1 and x_2 increase. WFQ, however, maintains a constant gap, being able to adapt the allocations to different initial backlog scenarios.

C. Impact of Network State Information

A direct comparison of the proposed semi-static and dynamic scheduling policies allows to quantify the performance improvement achieved by exploiting up-to-date queue states. In the following we limit this comparison to the proposed schemes of this paper, WTB-W and MDP, and additionally show eDVPUB to represent the close-to-optimal performances of semi-static policies. Therefore, the semi-static schemes WTB-W and eDVPUB are benchmarked with the dynamic MDP scheme. In all following figures, the value of p_e is set to 0.4.

In Fig. 12 the DVP achieved by the proposed scheduling policies is presented for different deadlines, backlogs, and fixed frame configuration $N = 4$. Most importantly, we witness a significant performance advantage of MDP in comparison to WTB-W and eDVPUP. This advantage increases with increasing deadlines, reaching multiple orders of magnitudes. This effect is intuitive as, at each frame, dynamic scheduling benefits from up-to-date queue states. A similar effect can be

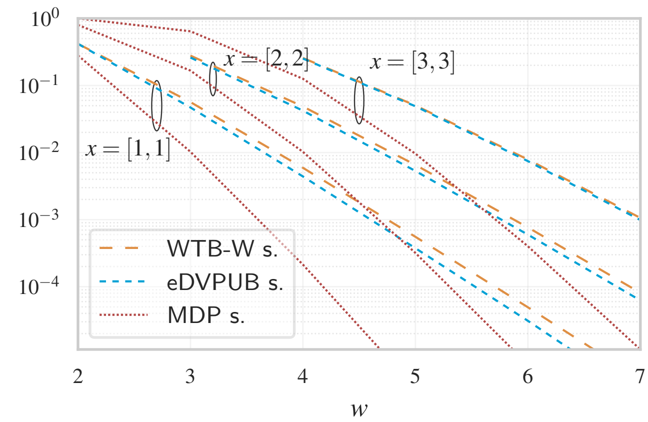

Fig. 12. DVP achieved by semi-static and dynamic schedulers for different deadlines w, increasing backlogs x_1, x_2, $N=6$, $p_e=0.4$.

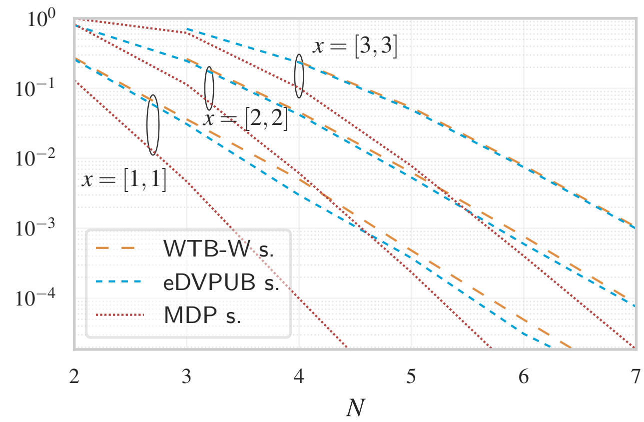

Fig. 13. DVP achieved by semi-static and dynamic schedulers for different frame sizes N, increasing backlogs x_1, x_2, $w=6$, $p_e=0.4$.

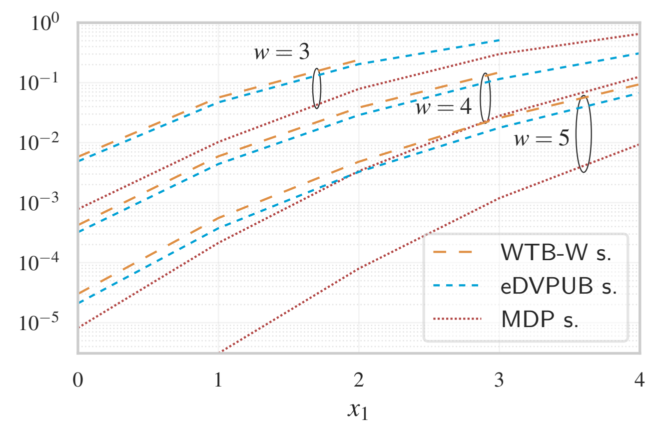

Fig. 14. DVP achieved by semi-static and dynamic schedulers for different backlogs x_1 and deadlines w, $x_2=1$, $N=6$, $p_e=0.4$.

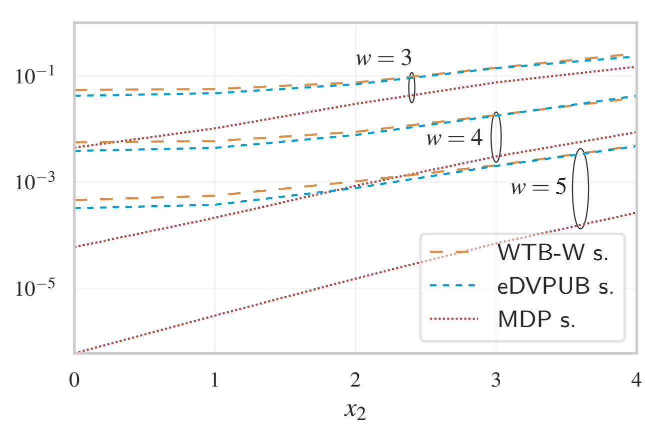

Fig. 15. DVP achieved by semi-static and dynamic schedulers for different backlogs x_2 and deadlines w, $x_1=1$, $N=6$, $p_e=0.4$.

observed in Fig. 13 where the DVP performance is shown for different frame lengths, backlogs, and a fixed deadline $w=6$. Finally, Fig. 14 and 15 show the impact of initial backlog on the DVP for different deadlines and a fixed frame configuration $N=6$. For fixed system parameters, increasing the initial backlogs results in an increasing DVP, which translates into smaller performance gaps between semi-static and dynamic scheduling schemes.

VIII. Conclusions

We proposed scheduling policies aimed at minimizing the end-to-end latency experienced by individual messages traversing the feedback loop, i.e. the delay from capturing the status information until the point in time when the corresponding feedback information is exposed. The scheduling policies allocate time slots to the transmitters to minimize the delay violation probability (DVP) taking the network queue states into account. Via simulations of the main system parameters and comparison baselines, we demonstrate the effectiveness of the proposed methods in reducing DVP. The results show that the proposed WTB-W semi-static scheduling policy achieves up to one order of magnitude improvement compared to a queue-agnostic scheme and close-to-optimal DVP. Further, simulations prove the superiority of the proposed MDP dynamic scheduler in reducing DVP compared to the existing Backpressure, Max Weight, and Weighted-Fair Queuing algorithms. By exploiting up-to-date queue states, MDP achieves a significant performance advantage in comparison to WTB-W reaching multiple orders of magnitude.

The contributions of this paper leave space for interesting future work. The proposed scheduling policies can be investigated for more complex system setups, for instance taking into account flow transformation within the feedback loop, asymmetric transmitter PERs, and non-stationary link qualities. Furthermore, the problem can be extended to consider multiple feedback loops sharing the same network, as well as multiple sensors, controllers and actuators. In this scenario new scheduling policies can be investigated, for instance, to take into account the correlation introduced by the transmission of multiple sensors and controllers sharing the same wireless channel. In addition, it would be interesting to study the impact of heterogeneous PER values for the different transmitters on the scheduling decision and the resulting DVP. Finally, an interesting next step is to investigate the application of the proposed scheduling policies in the existing wireless systems, such as NCS operating in IWSN or in 5G cellular networks.

APPENDIX

A. Proof of Proposition 1

Combining (13) and (5), the DVP can be calculated as

$$\begin{aligned}
&\mathrm{DVP}(w, y, x_1, x_2) = \\
&\quad = \mathbb{P}\{D(w) < y + x_1 + x_2\} \\
&\quad \overset{(1)}{=} \mathbb{P}\left\{ \min_{0\le u\le w} \left[S^2(w-u) + A^2(u)\right] < y + x_1 + x_2 \right\} \\
&\quad \overset{(2)}{=} \mathbb{P}\Big\{ \min_{0\le u\le w} \left[S^2(w-u) + D^1(u-1) + x_2\right] \\
&\qquad\qquad < y + x_1 + x_2 \Big\} \\
&\quad \overset{(3)}{=} \mathbb{P}\Big\{ \left\{S^2(w) < y + x_1 + x_2\right\} \cup \left\{S^2(w-1) < y + x_1\right\} \cup \\
&\qquad \bigcup_{u=2}^{w} \Big\{ \min_{0\le v\le u-1} [S^2(w-u) + S^1(u-1-v) + \\
&\qquad\qquad A(v) + A^1(v)] < y + x_1 \Big\} \Big\} \\
&\qquad \bigcup_{v=1}^{u-1} \{S_2(w-u) + S_1(u-1-v) < 0\} \} \Big\} \\
&\quad \overset{(4)}{=} \mathbb{P}\Big\{ \left\{S^2(w) < y + x_1 + x_2\right\} \cup \left\{S^2(w-1) < y + x_1\right\} \cup \\
&\qquad \bigcup_{u=2}^{w} \left\{ S^2(w-u) + S^1(u-1) < y + x_1 \right\} \Big\}.
\end{aligned}$$

Step (1) applies the dynamic server equation to the departures from second queue. Step (2) includes the fact that the cumulative arrivals at the second queue $A^2(u)$ correspond to the departures from the first queue in the previous frame $D^1(u-1)$. Step (3) applies the dynamic server equation to the departures from first queue and the fact that, for $u = 0$, $D^1(-1) = 0$. Finally, step (4) is obtained given that, for $0 < v \le u-1$, the cumulative arrival at the first queue and the initial backlog implies $\mathbb{P}\{S^2(w-u) + S^1(u-1-v) < 0\}$, which is equal to zero.

B. Proof of Lemma 1

To simplify the analysis of Eq. (19), we use $\alpha = (1 - p_e)e^{-s} + p_e$ and $\beta = k_a + x_1 - 1$

$$\begin{aligned}
\mathrm{WTB}(\mathbf{n}^2) &= \min_{s>0} \alpha^{\sum_{i=0}^{w} n^i} e^{s\,(\beta + x_2)} + \\
&\quad \sum_{u=1}^{1+w} \alpha^{(u-1)N - \sum_{j=0}^{u-2} n^j + \sum_{k=0}^{w-u} n^k} e^{s\,\beta} \\
&= \min_{s>0} e^{s\,\beta} \Bigg[\alpha^{\sum_{i=0}^{w} n^i} e^{s x_2} + \\
&\quad \sum_{u=1}^{1+w} \alpha^{(u-1)N - \sum_{j=0}^{u-2} n^j + \sum_{k=0}^{w-u} n^k} \Bigg]. \qquad (30)
\end{aligned}$$

Applying the definition of convexity to Eq. (30), $\forall \mathbf{n}, \mathbf{m} \in \mathbb{N}^{1\times w}, \lambda \in [0, 1]$ we obtain Eq. (31). From Eq. (31), (1) we have applied the definition of convexity to the exponentials $\alpha^{f(\lambda \mathbf{n^1} + (1-\lambda)\mathbf{n^2})}$ knowing that they are convex and (2) we have used the fact that the minimum of the sum is smaller or equal of the sum of the minima.

ACKNOWLEDGMENT

This work was supported by the DFG Priority Programme 1914 grant number KE1863/5-1, the Swedish Research Council (VR) under grant 2016-04404, and the VINNOVA Competence Center for Trustworthy Edge Computing Systems and Applications (TECoSA) at KTH Royal Institute of Technology.

REFERENCES

[1] W. Saad, M. Bennis, and M. Chen, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[2] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Design Automation Conference*, 2010, pp. 731–736.

[3] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir, "The Future of Human-in-the-Loop Cyber-Physical Systems," *Computer*, vol. 46, no. 1, pp. 36–45, 2013.

[4] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards Wearable Cognitive Assistance," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, 2014, p. 68–81.

[5] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systems-A survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 403–416, 2013.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[7] J. P. Champati, H. Al-Zubaidy, and J. Gross, "Transient Analysis for Multihop Wireless Networks under Static Routing," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 722–735, 4 2020.

[8] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 4. Publ by IEEE, 1990, pp. 2130–2132.

[9] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, 2010.

[10] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, June 1993.

[11] L. Tassiulas and A. Ephremides, "Dynamic Server Allocation to Parallel Queues with Randomly Varying Connectivity," *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466–478, 1993.

[12] R. Singh and P. R. Kumar, "Throughput optimal decentralized scheduling of multihop networks with end-to-end deadline constraints: Unreliable links," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 127–142, 1 2019.

[13] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-time scheduling for WirelessHART networks," in *Proceedings - Real-Time Systems Symposium*, 2010, pp. 150–159.

[14] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "End-to-End Delay Analysis for Fixed Priority Scheduling in WirelessHART Networks," in *2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium*, 2011, pp. 13–22.

[15] A. Saifullah, P. Babu Tiwari, B. Li, C. Lu, and P. Babu, "Accounting for Failures in Delay Analysis for WirelessHART Accounting for Failures in Delay Analysis for WirelessHART Networks Networks Recommended Citation Recommended Citation "Accounting for Failures in Delay Analysis for WirelessHART Networks " Report," Tech. Rep., 1 2012.

[16] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "End-to-end communication delay analysis in industrial wireless networks," *IEEE Transactions on Computers*, vol. 64, no. 5, pp. 1361–1374, 5 2015.

[17] Q. Wang, K. Jaffres-Runser, Y. Xu, and J. L. Scharbarg, "A certifiable resource allocation for real-time multi-hop 6TiSCH wireless networks," *IEEE International Workshop on Factory Communication Systems, WFCS*, pp. 1–9, 2017.

[18] V. P. Modekurthy, A. Saifullah, and S. Madria, "DistributedHART: A distributed real-time scheduling system for WirelessHART Networks," in *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS*, vol. 2019-April, 4 2019, pp. 216–227.

$$
\begin{aligned}
&\mathrm{WTB}\left(\lambda \mathbf{n} + (1-\lambda)\mathbf{m}\right) \\
&= \min_{s>0} e^{s\,\beta} \left[\alpha^{\sum_{i=0}^{w} \lambda n_i^1 + (1-\lambda) n_i^2} e^{s x_2} + \sum_{u=1}^{1+w} \alpha^{(u-1)N - \sum_{j=0}^{u-2} \lambda n_j^1 + (1-\lambda) n_j^2 + \sum_{k=0}^{w-u} \lambda n_k^1 + (1-\lambda) n_k^2} \right] \\
&\overset{(1)}{\leq} \min_{s>0} e^{s\,\beta} \Bigg[e^{s x_2} \left(\lambda \alpha^{\sum_{i=0}^{w} n_i^1} + (1-\lambda) \alpha^{\sum_{i=0}^{w} n_i^2} \right) + \\
&\qquad \sum_{u=1}^{1+w} \alpha^{(u-1)N} \left(\lambda \alpha^{-\sum_{j=0}^{u-2} n_j^1} + (1-\lambda) \alpha^{-\sum_{j=0}^{u-2} n_j^2} + \lambda \alpha^{\sum_{k=0}^{w-u} n_k^1} + (1-\lambda) \alpha^{\sum_{k=0}^{w-u} n_k^2} \right) \Bigg] \\
&= \min_{s>0} e^{s\,\beta} \Bigg[\lambda \left(\alpha^{\sum_{i=0}^{w} n_i^1} e^{s x_2} + \sum_{u=1}^{1+w} \alpha^{(u-1)N} \alpha^{-\sum_{j=0}^{u-2} n_j^1} \alpha^{-\sum_{j=0}^{u-2} n_j^1} \right) + \\
&\qquad (1-\lambda) \left(\alpha^{\sum_{i=0}^{w} n_i^2} e^{s x_2} + \sum_{u=1}^{1+w} \alpha^{(u-1)N} \alpha^{-\sum_{j=0}^{u-2} n_j^2} \alpha^{-\sum_{j=0}^{u-2} n_j^2} \right) \Bigg] \\
&\overset{(2)}{\leq} \lambda \min_{s>0} e^{s\,\beta} \left[\alpha^{\sum_{i=0}^{w} n_i^1} e^{s x_2} + \sum_{u=1}^{1+w} \alpha^{(u-1)N} \alpha^{-\sum_{j=0}^{u-2} n_j^1} \alpha^{-\sum_{j=0}^{u-2} n_j^1} \right] + \\
&\quad (1-\lambda) \min_{s>0} e^{s\,\beta} \left[\alpha^{\sum_{i=0}^{w} n_i^2} e^{s x_2} + \sum_{u=1}^{1+w} \alpha^{(u-1)N} \alpha^{-\sum_{j=0}^{u-2} n_j^2} \alpha^{-\sum_{j=0}^{u-2} n_j^2} \right] \\
&= \lambda \mathrm{WTB}\left(\mathbf{n}\right) + (1-\lambda) \mathrm{WTB}\left(\mathbf{m}\right).
\end{aligned}
\tag{31}
$$

[19] M. Yan, K. Y. Lam, S. Han, E. Chan, Q. Chen, P. Fan, D. Chen, and M. Nixon, "Hypergraph-based data link layer scheduling for reliable packet delivery in wireless sensing and control networks with end-to-end delay constraints," *Information Sciences*, vol. 278, 2014.

[20] F. Dobslaw, T. Zhang, and M. Gidlund, "End-to-End Reliability-Aware Scheduling for Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 758–767, 2016.

[21] G. Gaillard, D. Barthel, F. Theoleyre, and F. Valois, "High-reliability scheduling in deterministic wireless multi-hop networks," in *IEEE PIMRC*, Sep 2016.

[22] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, "Traffic aware scheduling algorithm for reliable low-power multi-hop ieee 802.15.4e networks," in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, 2012, pp. 327–332.

[23] Y. Chen, H. Zhang, N. Fisher, L. Y. Wang, and G. Yin, "Probabilistic per-packet real-time guarantees for wireless networked sensing and control," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2133–2145, 5 2018.

[24] R. Brummet, D. Gunatilaka, D. Vyas, O. Chipara, and C. Lu, "A Flexible Retransmission Policy for Industrial Wireless Sensor Actuator Networks," in *2018 IEEE International Conference on Industrial Internet, ICII 2018*, 11 2018, pp. 79–88.

[25] T. Gong, T. Zhang, X. S. Hu, Q. Deng, M. Lemmon, and S. Han, "Reliable dynamic packet scheduling over lossy real-time wireless networks," in *Leibniz International Proceedings in Informatics, LIPIcs*,

[30] K. Butler and M. A. Stephens, "The distribution of a sum of independent

vol. 133, 7 2019.

[26] P. Soldati, H. Zhang, Z. Zou, and M. Johansson, "Optimal routing and scheduling of deadline-constrained traffic over lossy networks," in *GLOBECOM - IEEE Global Telecommunications Conference*, 2010.

[27] S. Zoppi, J. P. Champati, J. Gross, and W. Kellerer, "Dynamic Scheduling for Delay-Critical Packets in a Networked Control System Using WirelessHART," in *2020 IEEE International Conference on Communications*, Jun 2020, pp. 1–7.

[28] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–320, 2006.

[29] Y. Jiang, "A Basic Stochastic Network Calculus," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Aug 2006.
binomial random variables," *Methodology and Computing in Applied Probability*, vol. 19, no. 2, p. 557, 2017.

[31] B. Liu and T. Quertermous, "Approximating the sum of independent non-identical binomial random variables," *arXiv preprint arXiv:1712.01410*, 2017.

[32] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[33] M. Cheffena, "Propagation Channel Characteristics of Industrial Wireless Sensor Networks [Wireless Corner]," *IEEE Antennas and Propagation Magazine*, vol. 58, no. 1, pp. 66–73, 2016.