Dynamic Scheduling for Delay-Critical Packets in a Networked Control System Using WirelessHART

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

***Abstract*—In future industrial scenarios, Wireless Sensor Networks (WSN) are envisioned to support the traffic of Networked Control Systems (NCS). WirelessHART is a prevalent WSN protocol that uses the Time Slotted Channel Hopping (TSCH) medium access to cope with the delay and reliability requirements of NCS in the harsh industrial environment. In TSCH, time slots and frequencies can be scheduled by a network coordinator to provide Quality of Service (QoS). In contrast to previous works that consider the end-to-end delay requirement of a flow of packets, we focus on a finite sequence of time-critical packets. These packets may belong to a time-critical message whose latency could significantly impact the NCS. Given an end-to-end delay deadline, our objective is to minimize the Delay Violation Probability (DVP) for a finite sequence of packets by dynamically scheduling the time slots in each frame. This is a challenging task as DVP depends on the instantaneous state of the network and requires its transient analysis. In this work, we model the wireless NCS as a two-queues WirelessHART network and propose the first transient analysis of DVP for a finite sequence of time-critical packets. Noting that DVP cannot be directly used for dynamic resource allocation, we propose a heuristic algorithm by relating DVP with the network's throughput. The proposed heuristic maximizes the expected throughput, is computed by solving a finite-horizon Markov Decision Process (MDP) and can be implemented at the network coordinator. Using simulation we demonstrate that the MDP-based heuristic achieves lower DVP compared to the classical MaxWeight and Weighted-Fair Queuing.**

***Index Terms*—NCS, WirelessHART, TSCH, QoS provisioning, transient regime, dynamic scheduling, MDP.**

I. Introduction

Wireless Sensor Networks (WSN) have gained significant momentum over the last decade [1] thanks to their advantages and a large number of applications. Initially deployed for time-critical monitoring of industrial processes, WSN are now studied for steering physical systems such as Networked Control Systems (NCS) [2]. Possible industrial NCS applications are automation control systems, collaborative robots and automatic guided vehicles [2]. In an NCS, represented in Fig. 1, a sensor periodically measures the state of a system and sends it to a controller which computes and sends a command to an actuator to steer it. NCS require low data rates (a few kbps) but are severely affected by network delays and packet losses. Unfortunately, WSN are prone to delays and packet losses due to the harsh industrial environment, where moving machines and interference from coexisting technologies critically degrade the communication quality [3].

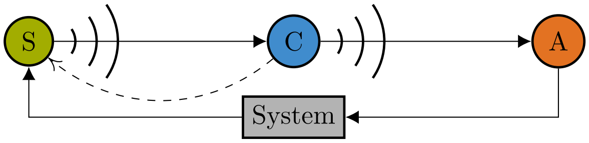

Fig. 1. NCS Architecture. Sensor (S), controller (C), and actuator (A) operate in a closed loop WirelessHART.

WirelessHART [4] is a prevalent wireless sensor networking protocol that caters to the above requirements of NCS. It uses the Time Slotted Channel Hopping (TSCH) medium access [5], which provides a time-division multiple access (TDMA) scheme in combination with frequency hopping – the former enables deterministic allocation of slots in a TDMA frame, and the latter prevents correlated packet drops due to sudden changes in the wireless medium. Under TSCH, the time slots and frequencies can be scheduled by a *network coordinator* to provide Quality of Service (QoS). Therefore, several delay- and reliability-aware scheduling algorithms were proposed in the literature to optimize the QoS for a stationary flow of packets [6]. In contrast to previous works, we model the communication on sensor-controller and controller-actuator links as a two-queue network and consider the end-to-end delay requirement of a finite sequence of time-critical packets arrived in a frame. These packets, for instance, may belong to a time-critical message whose latency could significantly impact the control performance, i.e. the stability/safety of the controlled plant. Therefore, given an end-to-end delay deadline w, we aim to find a dynamic schedule to minimize the Delay Violation Probability (DVP) of the time-critical packets.

Characterizing DVP for a finite sequence of packets is challenging as it depends on the instantaneous state of the network – such as backlogs in the queues at the time of arrival and temporal variability of the service offered by wireless links – and calls for the analysis of the short-term behaviour, aka transient analysis of the network. Nevertheless, analyzing a network in transient regime received little attention in the literature (cf. [7]), and to the best of our knowledge, it hasn't been addressed for WirelessHART. Furthermore, computing a dynamic schedule that minimizes the DVP is highly non-trivial because DVP computation requires all the future scheduling decisions.

In this work, we model the wireless NCS as a two-queue WirelessHART network (see Fig. 2) and present the first transient analysis for DVP for a sequence of time-critical packets arrived in a frame. We are interested in investigating dynamic resource allocation to minimize the DVP of time-critical packets. For this reason, we describe how our method can be applied to the WirelessHART medium access, but do not cover low-level implementation details. Noting that DVP cannot be directly used for dynamic resource allocation, we propose a heuristic algorithm by relating DVP with the network's throughput. The proposed heuristic maximizes the expected throughput; it is computed by solving a finite-horizon Markov Decision Process (MDP) and can be implemented at the network coordinator. Using simulation we demonstrate that the MDP-based heuristic achieves lower DVP in comparison to the classical MaxWeight (MW) [8] and Weighted-Fair Queuing (WFQ) [9] algorithms.

The rest of the paper is structured as follows. Sec. I-A provides a discussion of the related work. Sec. II defines the model of the network, the service provided by a WirelessHART device, and the problem statement. In Sec. III we characterize DVP and propose an MDP-based heuristic schedule, while Sec. IV evaluates the performance of the proposed schedule. Sec. V concludes the paper.

A. Related Work

In the literature, a large number of scheduling methods for TSCH industrial networks are available [6]. QoS provisioning in lossy TSCH networks is generally achieved via the allocation of re-transmissions. The idea was initially investigated by Zhang et al. [10] with a simple delay-aware scheme. Patti et al. [11] proposed a deadline-aware scheduling scheme where reliability is achieved via re-transmissions and blacklisting of lossy channels. Jin et al. [12] proposed a low-latency offline schedule where re-transmissions are allocated via virtual cells proportionally to the traffic and hop distance. Authors in [13] formulated a minimum latency TDMA scheduling problem, allocating redundant cells for re-transmissions based on the ETX routing metric. Hosni et al. [14] allocated transmission slots based on the Packet Delivery Ratio (PDR) while sending the packet to multiple receivers. Dobslaw et al. proposed SchedEx [15], a heuristic offline scheduling algorithm that guarantees network reliability using a lower bound on the required number of attempts from a given reliability bound. Similarly, authors in [16] investigated its latency improvement by scheduling the re-transmissions in contention slots and extending the probabilistic analysis. Zoppi et al. [17] improved the reliability of lossy links via re-transmissions of packets by designing the hopping sequence using the knowledge of the channels' PDR. Additionally, in [18] they proposed a reliability-aware dynamic scheduler that allocates re-transmissions based on the time-varying average PDR of the wireless channels.

Furthermore, reliable schedulers allocate slots considering the end-to-end reliability of a path. Yang et al. [19] provided a Markov chain model for the efficient re-transmission of packets in shared slots. Online re-allocation of slots is possible but it is only triggered by failures of nodes. Wang et al. [20] calculated an offline schedule solving a multi-objective optimization problem in terms of network reliability, worst-case delay, and energy consumption. Reliability is calculated for multiple flows and relays in terms of average PDR and relay forwarding probability. Gaillard et al [21] presented an extension of the landmark TSCH centralized scheduler TASA [22] for enhanced reliability. In their work, re-transmissions are based on the PDR of the path while satisfying the reliability constraints.

Our work differentiates from the available state-of-the-art for different reasons. By allocating a finite amount of re-transmissions, all the existing methods allow application packets to be dropped. This can cause severe failures to closed-loop systems like NCS. Furthermore, QoS was only considered for a stationary flow of packets and did not take into consideration the transient behaviour of a finite sequence of time-critical packets. Finally, in contrast to works that studied static and centralized scheduling in WirelessHART, we study centralized dynamic scheduling of time slots based on the instantaneous state of the system.

II. System Model and Problem Statement

We study a communication scheduling problem in an NCS consisting of a sensor, a controller, and an actuator. The data communication from the sensor to the controller, and the controller to the actuator is through WirelessHART. For analytical simplicity, in the model below, we assume that a packet received by the controller is processed within the same frame of reception and results in a new packet carrying the control command, and all packets have a fixed size of B bits. We now describe the details of the model and the network elements.

A. Arrivals, Backlogs, and Departures

We model the sensor-controller link, and controller-actuator link using a packet-flow, discrete-time, two-queue network with first-come-first-serve discipline; see Fig. 2. The time is discretized into TDMA frames. A sequence of y packets arrive 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 control performance, i.e. the stability/safety of the controlled plant. We are thus interested in analyzing the two-queue network for the time frames $k \in \{0, 1, \dots, w-1\}$. In this transient regime, the delay incurred by the time-critical packets depends on the initial backlogs in the queues at time k_0, and the temporal variations in the service received by the queues.

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

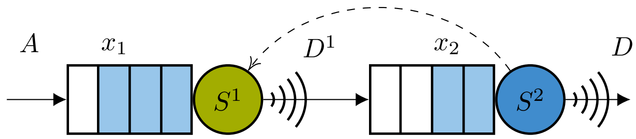

Fig. 2. Two-hops WirelessHART network model of a closed loop NCS.

We use $i \in \{1, 2\}$ to index the queues. Let x_i denote the backlog in queue i in frame 0. 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

$$A^1(k) = y + x_1, \tag{1}$$

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

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

where d_j^i is the number of packets departed queue i in frame j. In Eq. (2) a one-step delay is introduced between the reception of a packet and its service at the second queue. This models a practical aspect that WirelessHART devices cannot relay packets that have not been fully received. In the following, we use $A(k) = A^1(k)$ and $D(k) = D^2(k)$.

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

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

It quantifies the delay faced by the cumulative arrivals till frame $k-1$.

B. WirelessHART Service Model

Each TDMA frame comprises of N time slots to be shared between the transmissions of packets from the two queues. In this time-slotted medium access, the consecutive packet transmissions are characterized by uncorrelated channel fades thanks to the frequency hopping mechanism of TSCH. The network operates in a harsh industrial environment and is subject to packet loss. In WirelessHART, packets are transmitted using the OQPSK modulation with DSSS following the IEEE Std. 802.15.4 [23] and perform a simple CRC recovery mechanism. The corresponding packet error rate (PER) for the transmission of a fixed-size packet of B bits in a time slot is given by

$$p_e = 1 - \left[1 - Q\left(4\sqrt{\gamma}\right)\right]^B , \tag{5}$$

where $Q(\cdot)$ denotes the standard Gaussian error function, and γ is the signal-to-interference-and-noise ratio (SINR) [24]. Thus the random service provided for a single packet transmission is modelled as a Bernoulli r.v. with parameter $1 - p_e$.

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 TDMA allocation, the service offered by the i-th WirelessHART transmitter at frame k is distributed as a Binomial r.v.

$$s_k^i(n_k^i) \sim \mathcal{B}\left(n_k^i, 1 - p_e\right). \tag{6}$$

The cumulative service provided by the same transmitter over k frames is the sum of Binomial random variables with parameters $1 - p_e$, which is also a Binomial r.v.

$$S^i(k) = \sum_{j=0}^{k-1} s_j^i(n_j^i) \sim \mathcal{B}\left(\sum_{j=0}^{k-1} n_j^i, 1 - p_e\right). \tag{7}$$

C. Problem Statement

In order to provide enhanced QoS, we assume that the NCS controller has the role of network coordination and can dynamically allocate the time slots for both queues in every frame. A scheduling policy π determines this allocation and is defined as $\pi \triangleq \{n_0^1, n_1^1, \ldots, n_{w-1}^1\}$. We assume that, at each frame, the controller knows the network state $\mathbf{q}_k = (q_k^1, q_k^2)$, where q_k^1 and q_k^2 denote the lengths of first and second queues in frame k, respectively. This can be implemented in practical systems as 1) the controller knows the state of its own queue, and 2) knowing the initial state of the first queue, its evolution can be inferred by observing arrivals at the controller. The information of network state $\mathbf{q}_k$ is used to determine the allocation of TDMA slots for the next frame by a stationary scheduling policy π, where $n_k^1 = \pi(\mathbf{q}_k)$. The new allocation of TDMA slots is communicated to the first queue using the TSCH synchronization beacon, which can be transmitted at the beginning of every frame by the network coordinator.

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

$$\mathrm{DVP}(w, y, x_1, x_2) \coloneqq \mathbb{P}\{W(1) > w\} . \tag{8}$$

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 NCS; for example, if the sampling period is w frames, DVP represents the probability that the packet (carrying control command) in response to a packet generated by the sensor in a sampling period is delivered to the actuator in the next sampling period.

We are interested in a scheduling policy π that minimizes the DVP of y packets arrived in frame 0 and thus formulate the following optimization problem:

$$\underset{\pi \in \Pi}{\text{minimize}} \quad \mathrm{DVP}(w, y, x_1, x_2), \tag{9}$$

where Π denotes the set of all possible scheduling policies and is non-empty as each resulting slot allocation is valid.

III. Delay Violation Probability and Dynamic Throughput Maximisation

In this section we first characterize DVP using Stochastic Network Calculus (SNC) [25]. The input-output relation for a queue with a dynamic server is given by

$$D(k) \geq \min_{0\leq u\leq k}\left[A(u)+S(k-u)\right]. \tag{10}$$

Using Eq. (4), DVP can be obtained in terms of the virtual delay of the network as follows.

$$\begin{aligned}\text{DVP}\left(w,y,x_1,x_2\right) &= \mathbb{P}\left\{W(1)>w\right\}\\ &= \mathbb{P}\left\{D(w)<y+x_1+x_2\right\}. \end{aligned}\tag{11}$$

Using the above definition and Eq. (10), we can derive the exact expression of DVP.

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*

$$\begin{aligned}&\text{DVP}(w,y,x_1,x_2) =\\ &\mathbb{P}\Big\{\left\{S^2(w)<y+x_1+x_2\right\}\cup\left\{S^2(w-1)<y+x_1\right\}\cup\\ &\bigcup_{u=2}^{w}\left\{S^2(w-u)+S^1(u-1)<y+x_1\right\}\Big\}.\end{aligned}\tag{12}$$

Proof.

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

□

From Eq. (12), we observe that 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 highly non-trivial to use DVP to obtain a dynamic scheduling policy which causally allocates the time slots using only the past information. Furthermore, computing exact value of DVP is not tractable as it requires the calculation of the probability of union of w events that are not mutually disjoint.

Given the above challenges, we resort to computing heuristic schedules. To this end, in the following we obtain an upper bound for DVP using Markov's inequality.

$$\begin{aligned}&\text{DVP}(w,y,x_1,x_2) =\\ &\quad=\mathbb{P}\{D(w)\leq y+x_1+x_2-1\}\\ &\quad=\mathbb{P}\left\{y+x_1+x_2-D(w)\geq 1\right\}\\ &\quad\leq y+x_1+x_2-\mathbb{E}[D(w)],\end{aligned}\tag{13}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator. From Eq. (13) we infer that maximising the expected cumulative departures (throughput) of the network minimizes the upper bound of the DVP and thus potentially reduces DVP. Therefore, in the following we compute a heuristic schedule by solving the expected throughput maximization problem stated below.

$$\underset{\pi\in\Pi}{\text{maximize}}\quad \mathbb{E}[D(w)]=\sum_{k=0}^{w-1}\mathbb{E}\left[d_k^i\right]. \tag{14}$$

A. MDP-Based Heuristic

In order to solve the optimization problem in Eq. (14), 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 $n_k^1\in\{0,1,\ldots,N\}$. Given n_k^1, from (6) we have

$$\mathbb{P}\{s_k^1=r\}=\binom{n_k^1}{r}(1-p_e)^r p_e^{n_k^1-r}$$

$$\mathbb{P}\{s_k^2=r\}=\binom{N-n_k^1}{r}(1-p_e)^r p_e^{N-n_k^1-r}$$

The queues evolve as below:

$$q_{k+1}^1=\max(q_k^1-s_k^1,0), \tag{15}$$

$$q_{k+1}^2=\max(q_k^2-s_k^2,0)+\min(q_k^1,s_k^1). \tag{16}$$

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:

$$q_k^1\leq q_{k-1}^1, \tag{17}$$

$$q_k^1+q_k^2\leq q_{k-1}^1+q_{k-1}^2. \tag{18}$$

Conditions Eq. (17) and Eq. (18) 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. (17) and Eq. (18), 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: $0 < l_1' \leq l_1$, $0 < l_2'$, and $l_1' + l_2' \leq l_1 + l_2$. In this case $s_k^1 < q_k^1 = l_1$ and $s_k^2 < q_k^2 = l_2$. From Eq. (15) we have

$$q_{k+1}^1 = q_k^1 - s_k^1 \quad \Rightarrow \quad s_k^1 = l_1 - l_1'.$$

The number packets served from the second queue are computed from Eq. (16).

$$q_{k+1}^2 = q_k^2 - s_k^2 + s_k^1 \quad \Rightarrow s_k^2 = l_2 - l_2' + l_1 - l_1'.$$

Therefore,

$$\begin{aligned}&\mathbb{P}\{q_{k+1}^1 = l_1', q_{k+1}^2 = l_2' | q_k^1 = l_1, q_k^2 = l_2\}\\ &\quad = \mathbb{P}\{s_k^1 = l_1 - l_1', s_k^2 = l_2 - l_2' + l_1 - l_1'\}.\end{aligned}$$

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

$$\begin{aligned}&\mathbb{P}\{q_{k+1}^1 = 0, q_{k+1}^2 = l_2' | q_k^1 = l_1, q_k^2 = l_2\}\\ &\quad = \mathbb{P}\{s_k^1 \geq l_1, s_k^2 = l_2 - l_2' + l_1\}.\end{aligned}$$

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. (16), we have

$$\begin{aligned}&\mathbb{P}\{q_{k+1}^1 = l_1, q_{k+1}^2 = 0 | q_k^1 = l_1, q_k^2 = l_2\}\\ &\quad = \mathbb{P}\{s_k^1 = 0, s_k^2 \geq l_2\}.\end{aligned}$$

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 $\pi^\star$ that solves the maximization problem of Eq. (14). For this, we define the reward r_k of a policy π 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

$$r_k\left(\mathbf{q}_k, \pi(\mathbf{q}_k)\right) = \mathbb{E}\left[d_k^i | \pi(\mathbf{q}_k)\right] = \mathbb{E}\left[\min\left(q_k^2, s_k^2\right) | \pi(\mathbf{q}_k)\right]. \tag{19}$$

The total reward, obtained evaluating Eq. (19) over a horizon of w frames, is equal to Eq. (14). Therefore, the objective of the MDP is equal to the objective of Eq. (14).

Value iteration algorithms solve the MDP optimization recursively computing a value function J based on the Bellman's equation [26]. The optimal value function $J(\mathbf{q}_k)$ given a state $\mathbf{q}_k$ is

$$\begin{aligned}J_k(\mathbf{q}_k) = &\underset{\pi \in \Pi}{\text{maximize}} \;\; r_k(\mathbf{q}_k, \pi(\mathbf{q}_k)) + \\ &\sum_{\mathbf{q}_{k+1} \in \mathcal{Q}_{k+1}} \mathbb{P}\{\mathbf{q}_{k+1} | \mathbf{q}_k, \pi(\mathbf{q}_k)\} J_{k+1}(\mathbf{q}_{k+1}),\end{aligned} \tag{20}$$

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

For a finite number of states and actions, the optimal policy $\pi^\star$ for the MDP can be found by computing the optimal value function in Eq. (20) by backward recursion [26]. By the construct of the MDP, it is easy to see that $\pi^\star$ is optimal for Eq. (14) which is stated in the following theorem.

Theorem 1. *$\pi^\star$ is throughput optimal, i.e.*

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

IV. Evaluation Results

In this section, we evaluate the performance of the MDP-based scheduling policy via extensive simulations. We implement the network simulator in C, the MDP optimization in python, and use Matlab to visualize the results. We simulate different delay deadlines w, which arise from different NCS, and network configurations: TDMA frame sizes N, queue backlogs x_i, and service PERs p_e. For evaluation purposes, we select $y = 1$, $p_e = 0.5$, and low initial backlogs to reduce the runtime of simulations while exploring the network model parameters and application deadlines. This does not impact the validity of our results as arbitrary values of PERs and backlogs can be chosen.

For performance comparison, as existing TSCH scheduling algorithms do not consider the dynamic scheduling of two queues, we select classical packet scheduling algorithms as benchmark: MaxWeight (MW) [8] and Weighted-Fair Queuing (WFQ) [9]. At each frame, MW allocates all slots to the server with maximum backlog, while WFQ allocates slots proportionally to the ratio between the queue sizes.

In Fig. 3, we observe the effect of different TDMA frame sizes for different application deadlines and fix backlogs $x_1 = 2$ and $x_2 = 2$. As expected, lower application deadlines result in higher DVP. Also, we observe that MDP outperforms both MW and WFQ in all the scenarios with a performance gap up to one order of magnitude for MW and half order of magnitude for WFQ. As the frame size increases, the performance gap increases. This could be explained by the fact that a bigger frame size enlarges the action space of the MDP, improving the evaluation of the value function of Eq. (20) for each system's state. Thanks to this effect, given the same time budget from the NCS, MDP-based heuristic would be able to improve the system performance for higher TDMA frame sizes and shorter deadlines.

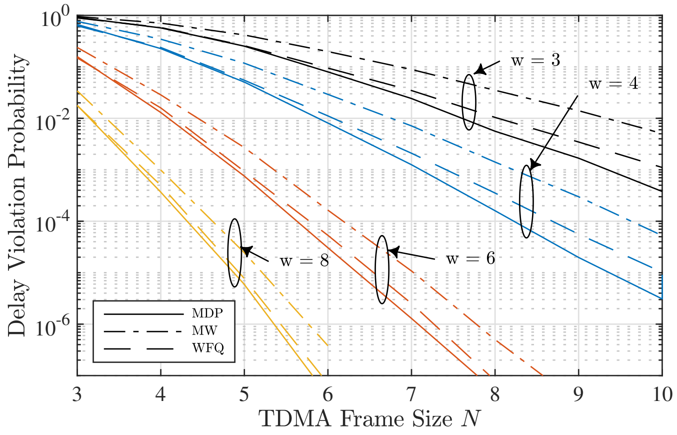

Fig. 3. DVP of the MDP-based heuristic against MW and WFQ for different frame sizes, deadlines, $x_1 = 2$, $x_2 = 2$, $p_e = 0.5$.

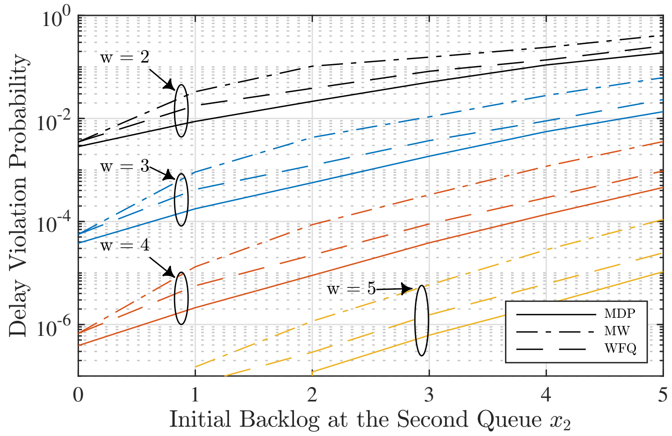

Fig. 5. DVP of the MDP-based heuristic against MW and WFQ for different deadlines, increasing backlogs at the second queue, $x_1 = 1$, $N = 8$, $p_e = 0.5$.

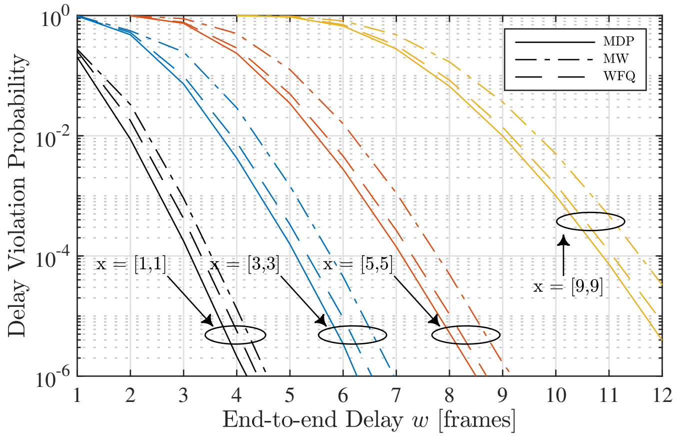

Fig. 4. DVP of the MDP-based heuristic against MW and WFQ for different deadlines, equal initial backlogs, $N = 8$, $p_e = 0.5$.

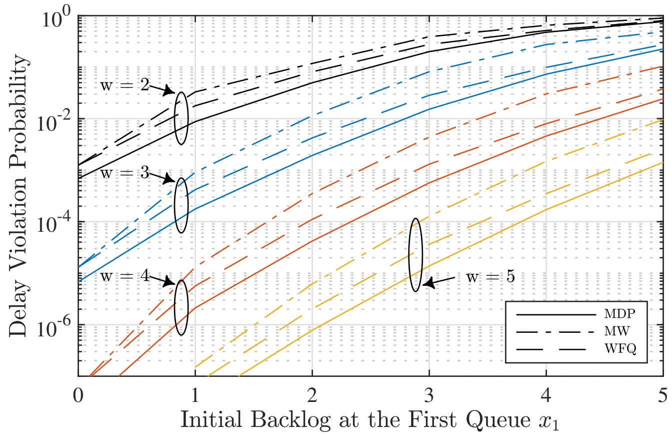

Fig. 6. DVP of the MDP-based heuristic against MW and WFQ for different deadlines, increasing backlogs at the first queue, $x_2 = 1$, $N = 8$, $p_e = 0.5$.

In Fig. 4, we observe the effect of equal initial backlogs on the achievable deadlines for $N = 8$. We notice that the total initial backlog has a major impact on the DVP of the time-critical packet. In fact, a small difference in initial backlog can result in multiple orders of magnitude of DVP. This effect should be taken into consideration by system designers to ensure the correct operation of the NCS. Furthermore, in Fig. 5 and Fig. 6, we investigate the impact different initial backlogs at the first and second queues, for different deadlines. The frame size is set to $N = 8$, while $x_1 = 1$ and $x_2 = 1$ for Fig. 5 and Fig. 6, respectively. The curves show that x_1 results in higher DVP in comparison to x_2. This is intuitive as backlog in the first queue has to traverse both queues and directly affects the delay of the time-critical packets. In particular, for $x_2 = 1$ in Fig. 5 and $x_1 = 1$ in Fig. 6, it can be observed that an unitary increase in x_1 has an impact on the DVP that is double compared to the same increment in x_2.

Finally, in Fig. 7, we evaluate the effect of PERs for different deadlines, frame size $N = 8$, backlogs $x_1 = 1$ and $x_2 = 1$. As the PER increases, the performance gap between different schedulers decreases, while DVP increases. Higher PERs cause high congestion in the wireless links, limiting the achievable performance of the schedulers. From the above results, we conclude that the MDP-based scheduling policy outperforms both MW and WFQ, in general, and results in a DVP that is an order of magnitude lower when compared to MW for certain settings.

V. Conclusions

Wireless NCS are highly affected by delays and packet losses introduced by the harsh industrial environment. WirelessHART is a prevalent protocol to cope with the delay and reliability requirements of NCS as it enables the centralized scheduling of network resources. Delay- and reliability-aware scheduling algorithms proposed in the literature optimize the QoS for a stationary flow of packets. Differently, we model the communication on sensor-controller and controller-actuator links as a two-queue network and consider the end-to-end delay requirement of a finite sequence of time-critical packets whose latency could significantly impact the NCS. Given an end-to-end delay deadline, we minimize the Delay Violation Probability (DVP) for a sequence of packets by

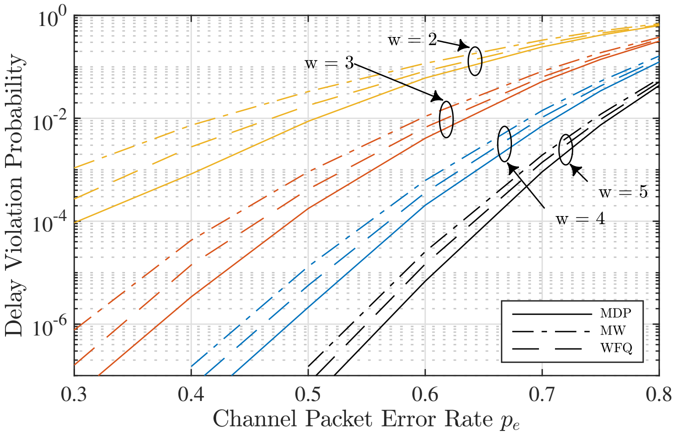

Fig. 7. DVP of the MDP-based heuristic against MW and WFQ for different channel PERs and deadlines, $x_1 = 1$, $x_2 = 1$, $N = 8$.

dynamically scheduling the time slots in each frame. This is a challenging task as DVP depends on the backlogs in the queues at the time of arrival and short-term variability of the service offered by wireless links. Furthermore, computing a dynamic schedule that minimizes the DVP is highly non-trivial because DVP computation requires all the future scheduling decisions. Therefore, we propose a heuristic algorithm by relating DVP with the network's throughput. The proposed heuristic maximizes the expected throughput; it is computed by solving a finite-horizon Markov Decision Process (MDP) and can be easily implemented at the network coordinator. Using simulation we demonstrate that the MDP-based heuristic achieves lower DVP in comparison to the classical MaxWeight (MW) and Weighted-Fair Queuing (WFQ) algorithms. The heuristic achieves a performance gap up to one order of magnitude for MW and half order of magnitude for WFQ.

For future work we plan to investigate the proposed MDP-based heuristic when two transmitters experience different PERs and with non-stationary link qualities. Furthermore, the derivation of exact DVP and its application for the optimization of a static network is an interesting open problem. Finally, the application of the proposed scheduling policy to other wireless systems, such as NCS operating in 5G cellular networks, could be investigated to evaluate its impact with respect to existing dynamic scheduling mechanisms.

Acknowledgment

This work was supported by the DFG Priority Programme 1914 Cyber-Physical Networking grant number KE1863/5-1.

References

[1] V. C. Gungor and G. P. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, Oct 2009.

[2] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A Review of Industrial Wireless Networks in the Context of Industry 4.0," *Wireless networks*, vol. 23, no. 1, 2017.

[3] M. Gürsu, M. Vilgelm, S. Zoppi, and W. Kellerer, "Reliable co-existence of 802.15.4e TSCH-based WSN and Wi-Fi in an aircraft cabin," in *IEEE ICC Workshops*, May 2016.

[4] HART Communications Foundation, *WirelessHART*.

[5] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, Apr 2012.

[6] R. T. Hermeto, A. Gallais, and F. Theoleyre, "Scheduling for IEEE802. 15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey," *Computer Communications*, vol. 114, 2017.

[7] J. P. Champati, H. Al-Zubaidy, and J. Gross, "Transient Delay Bounds for Multi-Hop Wireless Networks," *to be published in IEEE/ACM Transactions on Networking (TON)*, 2020.

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

[9] 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.

[10] S. Zhang, G. Zhang, A. Yan, Z. Xiang, and T. Ma, "A highly reliable link scheduling strategy for WirelessHART networks," in *International Conference on Advanced Technologies for Communications*, Oct 2013.

[11] G. Patti and L. Lo Bello, "A Priority-Aware Multichannel Adaptive Framework for the IEEE 802.15.4e-LLDN," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, Oct 2016.

[12] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara, "A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks," in *IEEE WCNC*, Apr 2016.

[13] G. Daneels, S. Latré, and J. Famaey, "Efficient Recurrent Low-Latency Scheduling in IEEE 802.15.4e TSCH Networks," in *IEEE BlackSeaCom*, June 2019.

[14] I. Hosni and F. Théoleyre, "Adaptive k-cast scheduling for high-reliability and low-latency in IEEE802. 15.4-TSCH," in *International Conference on Ad-Hoc Networks and Wireless*. Springer, 2018.

[15] 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, Apr 2016.

[16] M. Hashimoto, N. Wakamiya, M. Murata, Y. Kawamoto, and K. Fukui, "End-to-end reliability- and delay-aware scheduling with slot sharing for wireless sensor networks," in *International Conference on Communication Systems and Networks*, Jan 2016.

[17] S. Zoppi, H. M. Grsu, M. Vilgelm, and W. Kellerer, "Reliable hopping sequence design for highly interfered wireless sensor networks," in *IEEE LANMAN*, June 2017.

[18] S. Zoppi, A. Van Bemten, H. M. Grsu, M. Vilgelm, J. Guck, and W. Kellerer, "Achieving Hybrid Wired/Wireless Industrial Networks With WDetServ: Reliability-Based Scheduling for Delay Guarantees," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, May 2018.

[19] D. Yang, Y. Xu, H. Wang, T. Zheng, H. Zhang, H. Zhang, and M. Gidlund, "Assignment of Segmented Slots Enabling Reliable Real-Time Transmission in Industrial Wireless Sensor Networks," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, June 2015.

[20] Q. Wang, K. Jaffrs-Runser, Y. Xu, and J. Scharbarg, "A certifiable resource allocation for real-time multi-hop 6TiSCH wireless networks," in *IEEE WFCS*, May 2017.

[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, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH," *IEEE Sensors Journal*, vol. 13, no. 10, Oct 2013.

[23] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, Sep. 2011.

[24] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*. John Wiley & Sons, 2010, vol. 4.

[25] Y. Jiang, "A Basic Stochastic Network Calculus," in *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. ACM, 2006.

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