

State-dependent Priority Scheduling for Networked Control Systems

Ben W. Carabelli*, Rainer Blind[†], Frank Dürr*, Kurt Rothermel*

*Institute of Parallel and Distributed Systems
University of Stuttgart, 70569 Stuttgart, Germany
{ben.carabelli|duerr|rothermel}@ipvs.uni-stuttgart.de

[†]Institute for Systems Theory and Automatic Control
University of Stuttgart, 70569 Stuttgart, Germany
rainer.blind@ist.uni-stuttgart.de

Abstract—Networked control systems (NCS) have attracted considerable attention in recent years. While the stabilizability and optimal control of NCS for a given communication system has already been studied extensively, the design of the communication system for NCS has recently seen an increase in more thorough investigation. In this paper, we address an optimal scheduling problem for a set of NCS sharing a dedicated communication channel, providing performance bounds and asymptotic stability. We derive a suboptimal scheduling policy with dynamic state-based priorities calculated at the sensors, which are then used for stateless priority queuing in the network, making it both scalable and efficient to implement on routers or multi-layer switches. These properties are beneficial towards leveraging existing IP networks for control, which will be a crucial factor for the proliferation of wide-area NCS applications. By allowing for an arbitrary number of concurrent transmissions, we are able to investigate the relationship between available bandwidth, transmission rate, and delay. To demonstrate the feasibility of our approach, we provide a proof-of-concept implementation of the priority scheduler using real networking hardware.

Index terms—cyber-physical systems; networked control systems; optimization; scheduling

I. Introduction

Networked control systems (NCS) [1] enable the economical and flexible deployment of control applications by closing feedback loops over a packet-switched network. Applications lending themselves naturally to this kind of architecture include autonomous traffic systems, smart factories, and tele-operation, to name but a few. With the rise of an Internet of Things (IoT) ecosystem that brings forth an increasing number of network-enabled appliances, we also expect to see more and more transient and dynamically changing NCS in the future. All these types of applications may commonly involve widely distributed sensors, actuators and controllers. A crucial factor for the proliferation of these applications will be to what extent existing IP networks can be leveraged by control applications.

Building NCS on top of packet-switched networks is a challenging task, because the network imposes resource constraints that conflict with the communication requirements of the control systems. Therefore, problems relating to NCS have been studied from various angles over the last decade. In

most of these works, the communication system is assumed to be given and modelled as random packet loss or delay. Under these assumptions, a stabilizing or optimal controller as well as bounds on the loss probability or delay are derived. E.g., in [2] the optimal state estimation with intermittent observations as well as bounds for the maximal allowed packet loss probability are derived. The dual problem, i.e., optimal control over links with random packet losses is studied in [3], [4]. These works are extended to more complex scenarios and more detailed models for the loss process. E.g., in [5] the case of packet loss and delay is studied, in [6] also acknowledgment packets can get lost, in [7], packet loss is modelled by a Markov chain, and in [8] trade-offs between various network related quantities are studied. To sum up, stabilizability and optimal control for a given model of the communication system has been studied extensively.

Recently, the (co-)design of the communication system has also received increasing attention. For instance, it has been investigated how the control performance may be improved through modifications at the transport [9], network [10] or data link layer [11]–[17]. In this paper, we concern ourselves with the latter.

As already stated, most of the previous works model the network as a random loss and delay process. Unfortunately, this does not allow strict stability guarantees, but only allows stochastic stability guarantees. Moreover, loss probabilities and delay distributions of a network are usually neither constant nor readily available. While such assumptions make perfect sense in wireless networks with a high degree of random channel failures, stronger guarantees are possible in switched Ethernet-based IP networks, where bandwidth restrictions and queuing delays are the main limiting factors.

In this paper, we start out under the premise that a dedicated ‘network slice’ with fixed resources is available for a set of control systems. This could be realized, e.g., by making a bandwidth reservation along a common route using the integrated services architecture (IntServ) [18], [19] of the Internet protocol stack. Alternatively, several network virtualization technologies have been proposed [20] which allow provisioning of such an isolated network slice with arbitrary topologies. Our question is then how to optimally allocate the available resources among all control systems.

This question boils down to the scheduling discipline which is used to manage the control systems’ access to the network. An early example of a dynamic scheduler for control systems

This work was supported by the German Research Foundation (DFG) under the research grant “Integrated Controller Design Methods and Communication Services for Networked Control Systems (NCS)” (RO 1086/20-1, AL 316/13-1).

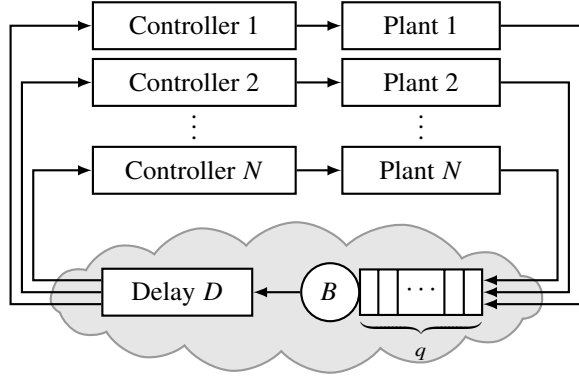


Fig. 1. System architecture: N different NCS communicate over a shared (virtual) link with bandwidth B , delay D , and input queue of capacity q .

is the Maximum-Error-First policy used for the Try-Once-Discard (TOD) protocol [21], [22]. It assumes that delay-free broadcast communication is used, which may be a reasonable approximation for fieldbuses, but is an unrealistic assumption for IP networks. In [13], a static periodic scheduling policy is derived for a set of different NCS, where the schedule is derived from average dwell times. In [23], a stochastic RTOS scheduler for anytime control on embedded systems is studied. In [24], the suitability of weakly hard real-time schedulers [25] has been investigated by deriving sufficient stability conditions for a single NCS, however, without directly taking network utilization and competing traffic into account.

Dynamic state-based schedulers for NCS have also been investigated. In [14], [15], network schedulers are designed for a single delay-free control loop, investigating under which conditions a separation principle holds for the scheduler and certainty-equivalent controller design. This is extended in [16], [17] to multiple control loops, where each sensor uses a local scheduling policy to reduce the number of transmissions, with probabilistic contention based medium access among all NCS. However, these works do not strictly follow a fixed resource constraint. In [17], for instance, a price for transmissions is added to the LQR cost and adapted dynamically to maintain an upper bound on the total average transmission rate. Therefore, while available bandwidth may be shared with other applications, a significant amount of bandwidth has to be over-provisioned in advance.

By contrast, we propose to use priority scheduling to utilize the reserved bandwidth as well as possible to optimize control performance. A similar approach is proposed in [11], [12], where a state-based dynamic priority scheduler for physically coupled NCS is derived from a quadratically structured event-triggering rule and designed together with a suboptimal controller. We, however, assume that the controllers are already given, and design a scheduler accordingly. In [12], a tuning parameter is used to penalize transmissions, thereby reducing control traffic. Like the state-based scheduling strategies mentioned in the previous paragraph, this approach assumes that only one NCS can transmit at any time. We, on the other hand, provide formulations for an arbitrary fixed

transmission rate (i.e., number of concurrent transmissions). Also, while [12] assumes that the delays are given a priori, we use a channel abstraction conforming with [18] to model the relationship between available bandwidth, overall transmission rate, and delay. We use LMI stability conditions for switched linear systems from [26] to design a scheduler that guarantees asymptotic stability and provides performance bounds for all NCS, and show how these conditions can be generalised to accommodate concurrent transmissions. Our scheduling policy uses dynamic state-based priorities calculated at the sensors which are then used for stateless priority queuing in the network, making it both scalable and efficient to implement. Priority queuing can be found, e.g., as part of the weighted fair queuing (WFQ) algorithm [27] supported by many routers.

Our paper is organized as follows. In Sec. II, we present our system model, which comprises a set of NCS and the shared communication channel. Based on this model, we formally state our scheduling problem in Sec. III. We then derive a scheduler under the assumption that the communication channel admits only one transmission each sampling period in Sec. IV, which we then go on to generalize for an arbitrary number of transmissions per sampling period in Sec. V. In Sec. VI, we illustrate our approach with a proof-of-concept implementation and simulation examples. A short discussion of our results and possible avenues for future work in Sec. VII concludes this paper.

II. System Model

Consider a set of N different NCS, each comprising a plant and controller, where state samples are sent from sensor to controller over a (virtual) network, as shown in Fig. 1. The network is shared by these NCS, but no other applications (i.e., no cross-traffic), and is modelled as a virtual link with bandwidth B , delay D , and an input queue of capacity q . The queue is served by a priority scheduler. In the following, we describe the components of the system model—control systems, virtual link, and scheduler—in more detail.

A. Control System Model

We assume that all NCS are sampled synchronously at times t_k with a common sampling period $t_{k+1} - t_k = T_s$. Each sample of a plant's state is sent in a packet addressed to the corresponding controller, together with an attached priority value. At the same time, each controller applies a control input to the corresponding plant based on the packets that it has received so far.

The plant of NCS $i \in \{1, \dots, N\}$ is modelled as a discrete-time system

$$x_{k+1}^i = A^i x_k^i + B^i u_k^i, \quad (1)$$

where $x_k^i \in \mathbb{R}^{n_i}$ is the state and $u_k^i \in \mathbb{R}^{m_i}$ is the input of plant i at time t_k . Upon sampling, the sensor sends a tuple (x_k^i, v_k^i) over the network, where v_k^i is the packet priority used by the scheduler to dequeue packets for transmission. We will derive a function for calculating these priorities in Sec. IV. As measurements may be dropped by the scheduler due

to bandwidth limitations, each controller uses the following predictive control law proposed in [9].

$$\hat{x}_{k+1}^i = \theta(i, k)A^i \hat{x}_k^i + (1 - \theta(i, k))A^i \hat{x}_k^i + B^i u_k^i, \quad (2)$$

$$u_k^i = -K^i \hat{x}_k^i \quad (3)$$

Here, the binary arrival function $\theta(i, k)$ indicates whether the state measurement x_k^i has been successfully received at the controller by time t_{k+1} ($\theta = 1$) or not ($\theta = 0$). The controller uses the predictive state estimate $\hat{x}_k^i \in \mathbb{R}^{n_i}$ to compensate for a constant transfer delay of T_s and for dropped packets.

We assume that all plants (A^i, B^i) are controllable and that a stabilizing controller K^i is given, i.e., the matrices $A^i - B^i K^i$ are Schur-stable. We use the standard LQR cost

$$J^i = \sum_{k=1}^{\infty} x_k^{i\top} Q^i x_k^i + 2x_k^{i\top} H^i u_k^i + u_k^{i\top} R^i u_k^i \quad (4)$$

as a measure of the control performance of NCS i . Therefore, a natural choice for K^i is the standard infinite-horizon LQR controller, which we use in our evaluations.

However, the controller may be arbitrary, and we assume it to be designed independently from the scheduler. Please note that we do not derive the optimal controller for this setup. Concerning the separation of optimal control and scheduling, we refer, e.g., to [16], [17].

B. Virtual Link Model

We assume that all NCS packets are of uniform size L , which accounts for the memory required for both the state and the attached priority value. A certain bandwidth B is allocated to the shared link, with an input queue of capacity q (in packets) to buffer packets for forwarding. Moreover, the link incurs a fixed delay D , which results from the network delay of the underlying communication service. The end-to-end delay experienced by a batch of q NCS packets is therefore $T = \frac{L}{B}q + D$. This channel model corresponds to the end-to-end service offered, for instance, by the ‘Guaranteed Service’ class of the IntServ architecture [18] for IP networks.

Because the transfer delay of a packet must not exceed one sampling period T_s in order to be useful to the controller, the queue must be dimensioned accordingly, such that

$$q \leq \frac{B}{L}(T_s - D). \quad (5)$$

Naturally, this is equivalent to a channel with a one step delay and fixed capacity q . For a given (physical) network, we may trade capacity for sampling frequency subject to (5).

As mentioned in the introduction, this shared link need not necessarily be physically restricted, but could also be realized through a virtualized network slice or an IP resource reservation, for instance. Therefore, the bandwidth B may also be regarded as a design parameter to adjust queue capacity and sampling period.

C. Priority Scheduler Model

Following the assumption of synchronous sampling, we also assume that all packets from one sampling period arrive at the queue simultaneously, as this yields the worst-case transfer

delay. The scheduler is then responsible for servicing the q highest-priority packets, the remainder being dropped. To this end, we define the set of q -out-of- N -subsets as

$$S_q = \left\{ S \subset \{1, \dots, N\} \mid |S| = q \right\} = \bigcup_s \{S_s\}, \quad (6)$$

with an arbitrary numbering $s = 1, \dots, \binom{N}{q}$. (For the simplest case $q = 1$, we choose $S_s = \{s\}$.)

Now, we can define the priority scheduling function

$$\sigma(k) = \arg \min_s \sum_{i \in S_s} v_k^i, \quad (7)$$

such that each NCS $i \in S_{\sigma(k)}$ receives a successful transmission in the period $[t_k, t_{k+1})$, whereas all others do not. Thereby, the scheduling function (7) imposes a coupling on the individual arrival indicators $\theta(i, k)$ of all NCS:

$$\theta(i, k) = \delta(i, \sigma(k)) \quad \text{with} \quad \delta(i, s) = \begin{cases} 1 & \text{if } i \in S_s \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

III. Problem Statement

In order to fully specify the scheduler, we need to define the priority values v_k^i for the packets generated by all NCS. The priorities must be designed such that all participating NCS remain stable under limited-capacity priority scheduling. Within these constraints, the scheduler should optimize the overall control performance. In order to formalize the problem statement, we consider the overall system using a lumped switching model that integrates all NCS models with the scheduling behaviour.

First, we rewrite the model of each individual NCS in a simpler form. Plugging (1)–(3) together, we get the following autonomous model for an individual NCS with augmented state $\mathbf{x}_k^i = [x_k^{i\top} \hat{x}_k^{i\top}]^\top$:

$$\mathbf{x}_{k+1}^i = \mathbf{A}_{\theta(i,k)}^i \mathbf{x}_k^i, \quad \text{with} \quad \mathbf{A}_{\theta}^i = \begin{bmatrix} A^i & -B^i K^i \\ \theta A^i & (1-\theta)A^i - B^i K^i \end{bmatrix}. \quad (9)$$

Note that this is a switching system with two modes: \mathbf{A}_0^i for open-loop and \mathbf{A}_1^i for closed-loop behaviour. The cost (4) of the individual NCS can be rewritten as

$$J^i = \sum_{k=1}^{\infty} \mathbf{x}_k^{i\top} \mathbf{Q}^i \mathbf{x}_k^i, \quad \text{with} \quad \mathbf{Q}^i = \begin{bmatrix} Q^i & -H^i K^i \\ -K^{i\top} H^{i\top} & K^{i\top} R^i K^i \end{bmatrix}. \quad (10)$$

Next, we combine all the systems into one model for the lumped state $\eta_k = [\mathbf{x}_k^{1\top}, \mathbf{x}_k^{2\top}, \dots, \mathbf{x}_k^{N\top}]^\top$ of all NCS:

$$\eta_{k+1} = \mathcal{A}_{\sigma(k)} \eta_k, \quad (11)$$

$$\text{with } \mathcal{A}_s = \text{diag}(\mathbf{A}_{\delta(1,s)}^1, \mathbf{A}_{\delta(2,s)}^2, \dots, \mathbf{A}_{\delta(N,s)}^N)$$

This overall system switches between $\binom{N}{q}$ modes, one for every possible outcome of the scheduler $\sigma(k)$. The mode of each subsystem is determined by whether it is a member of

the set $S_{\sigma(k)}$ of scheduled systems as determined by (8). The cost of the overall system is the sum over all NCS

$$\mathcal{J} = \sum_{i=1}^N J^i = \sum_{k=1}^{\infty} \eta_k^T \mathcal{Q} \eta_k, \quad (12)$$

with $\mathcal{Q} = \text{diag}(\mathbf{Q}^1, \mathbf{Q}^2, \dots, \mathbf{Q}^N)$.

Our goal is to choose the priorities v_k^i which determine the scheduling function $\sigma(k)$ in (7) such that the overall system (11) is asymptotically stable, while aiming to minimize the overall cost \mathcal{J} . Moreover, the priority v_k^i must depend solely on state information of the individual NCS i that is available at the sensor.

IV. State-dependent Scheduler for $q = 1$

For ease of presentation, we will begin our analysis for a link capacity of $q = 1$, and later generalize the problem setting for an arbitrary queue length in Sec. V. In this particular case, the scheduling function (7) simplifies to

$$\sigma(k) = \arg \min_i v_k^i.$$

We use the following sufficient condition from Geromel et al. [26] to find a state-dependent, stabilizing scheduling function and performance bound for the system (11)–(12). (Other stabilizing switching designs can be found, e.g., in [28], [29] and references therein.)

Theorem 1 ([26]): Let $\mathcal{Q} \geq 0$ be given. If there exist a set of positive definite matrices $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_N$ and a matrix Π with entries $\pi_{ij} \geq 0$ and $\sum_{i=1}^N \pi_{ij} = 1$, $j = 1, \dots, N$ satisfying the so-called Lyapunov–Metzler inequalities

$$\mathcal{A}_j^T \left(\sum_{i=1}^N \pi_{ij} \mathcal{P}_i \right) \mathcal{A}_j - \mathcal{P}_j + \mathcal{Q} < 0 \quad (13)$$

for all $j \in \{1, \dots, N\}$, then the switching policy

$$\sigma(k) = \arg \min_i \eta_k^T \mathcal{P}_i \eta_k \quad (14)$$

makes the origin $\eta = 0$ of the system (11) globally asymptotically stable and

$$\mathcal{J} = \sum_{k=1}^{\infty} \eta_k^T \mathcal{Q} \eta_k < \eta_0^T \mathcal{P}_{\sigma(0)} \eta_0, \quad (15)$$

i.e., the overall performance is bounded. \square

Note that the condition (13) can also be found in the context of Markov Jump Linear Systems (MJLS). More precisely, if Π were the transition probability matrix of a Markov chain, (13) would give mean square stability, see, e.g., [30]. However, in the considered scenario, the jumps are not driven by a Markov chain but selected deterministically by the scheduler (14). Thus, (13) together with the scheduler (14) guarantees stability in the classical, non-stochastic sense.

Note further that finding a feasible solution to (13) is a non-convex problem due to the products of π_{ij} and \mathcal{P}_i . However, if the matrix Π is fixed, it becomes an LMI problem which can be solved efficiently using available numeric methods. Therefore, we will later propose a heuristic for determining Π using necessary stability conditions.

However, the scheduling function (14) depends on the full state of the lumped system, i.e. without further knowledge the states of all NCS have to be aggregated before a scheduling decision can be taken. However, our system architecture requires that packet priorities only depend on the local state.

We can rectify this by imposing some constraints on the structure of the matrices \mathcal{P}_i as follows. For each NCS, we introduce two positive definite $2n_i \times 2n_i$ matrices \mathbf{P}_0^i and \mathbf{P}_1^i and add the restriction

$$\mathcal{P}_i = \text{diag}(\mathbf{P}_{\delta(1,i)}^1, \mathbf{P}_{\delta(2,i)}^2, \dots, \mathbf{P}_{\delta(N,i)}^N) \quad (16)$$

to the conditions in Thm. 1. This allows us to rewrite the scheduling function (14) as follows:

$$\begin{aligned} \sigma(k) &= \arg \min_i \eta_k^T \mathcal{P}_i \eta_k = \arg \min_i \sum_n \mathbf{x}_k^{nT} \mathbf{P}_{\delta(n,i)}^n \mathbf{x}_k^n \\ &= \arg \min_i \mathbf{x}_k^{iT} \mathbf{P}_1^i \mathbf{x}_k^i + \sum_{n \neq i} \mathbf{x}_k^{nT} \mathbf{P}_0^n \mathbf{x}_k^n \\ &= \arg \min_i \mathbf{x}_k^{iT} (\mathbf{P}_1^i - \mathbf{P}_0^i) \mathbf{x}_k^i + \sum_{n \neq i} \mathbf{x}_k^{nT} \mathbf{P}_0^n \mathbf{x}_k^n. \end{aligned}$$

As the minimum in the expression above is independent of the last term, this is equivalent to the priority scheduler (7) together with the priority functions

$$v_k^i = v^i(\mathbf{x}_k^i) = \mathbf{x}_k^{iT} (\mathbf{P}_1^i - \mathbf{P}_0^i) \mathbf{x}_k^i, \quad (17)$$

which depend only on the state of the corresponding NCS.

In order to use this scheduler, we must first determine whether a particular set of NCS can be stabilized under this discipline, and calculate the corresponding coefficient matrices of the priority functions. We will formulate this admission phase as an optimization problem based on the conditions in Thm. 1. In the following, we reformulate the Lyapunov–Metzler inequalities (13) and propose a heuristic for choosing the matrix Π in order to make the problem convex. The benefits of this formulation will become apparent in Sec. V, where we show how to generalize our approach to arbitrary queue lengths.

All \mathcal{P}_i as defined in (16) have the same block diagonal structure as \mathcal{A}_i and \mathcal{Q} . Furthermore, as $q = 1$, the i^{th} diagonal block of $\mathcal{A}_i/\mathcal{P}_i$ is always given by $\mathbf{A}_1^i/\mathbf{P}_1^i$ (closed-loop), whereas the remaining blocks are $\mathbf{A}_0^j/\mathbf{P}_0^j$ (open-loop). This allows us to replace the inequalities (13) by the following set of lower-dimensional inequalities:

$$\mathbf{A}_1^{iT} \left(\pi_{ii} \mathbf{P}_1^i + (1 - \pi_{ii}) \mathbf{P}_0^i \right) \mathbf{A}_1^i - \mathbf{P}_1^i + \mathbf{Q}_i < 0, \quad \forall_i \quad (18)$$

$$\mathbf{A}_0^{iT} \left(\pi_{ij} \mathbf{P}_1^j + (1 - \pi_{ij}) \mathbf{P}_0^j \right) \mathbf{A}_0^i - \mathbf{P}_0^i + \mathbf{Q}_i < 0, \quad \forall_{i \neq j} \quad (19)$$

For each $i = 1, \dots, N$, we can replace all inequalities in (19) by a single inequality by choosing $\pi_{ij} = p_i$, $\forall_{j=1, \dots, N}$. For notational convenience we also define $m_i = \pi_{ii}$, which gives us

$$\Pi = \begin{bmatrix} m_1 & p_1 & \cdots & p_1 \\ p_2 & m_2 & \cdots & p_2 \\ \vdots & \vdots & \ddots & \vdots \\ p_N & p_N & \cdots & m_N \end{bmatrix}.$$

$$\mathbf{A}_1^{i\top} \left(\sum_{j \in S_j} \pi_{jk} \mathbf{P}_1^i + \sum_{j \notin S_j} \pi_{jk} \mathbf{P}_0^i \right) \mathbf{A}_1^i - \mathbf{P}_1^i + \mathbf{Q}^i < 0, \quad \forall(i, k), i \in S_k \quad (27)$$

$$\mathbf{A}_0^{i\top} \left(\sum_{j \in S_j} \pi_{jk} \mathbf{P}_1^i + \sum_{j \notin S_j} \pi_{jk} \mathbf{P}_0^i \right) \mathbf{A}_0^i - \mathbf{P}_0^i + \mathbf{Q}^i < 0, \quad \forall(i, k), i \notin S_k \quad (28)$$

If we fix all m_i and p_i , then (18)–(19) become LMIs in $\mathbf{P}_{0/1}^i$. In [26], the authors use an approach which is equivalent to choosing $m_i = \alpha \in [0, 1]$ and performing a line search over α to accomplish this simplification. However, since (13) implies that all $\sqrt{m_i} \mathcal{A}_i$, $i = 1, \dots, N$ are Schur-stable [31], we propose the heuristic

$$m_i = \rho(\mathcal{A}_i)^{-2} \cdot \alpha, \quad (20)$$

where $\rho(\cdot)$ is the spectral radius, in order to exclude infeasible values a priori. Because in Thm. 1 the matrix Π is required to be left-stochastic, i.e. with non-negative entries and columns summing up to 1, we can determine the coefficients p_i entirely from m_i :

$$p = (\mathbf{1}\mathbf{1}^\top - I)^{-1}(\mathbf{1} - m), \quad (21)$$

where p and m are the corresponding column vectors of p_i and m_i . This allows us to formulate our first main result.

Theorem 2: Let $q = 1$ and a set of N control systems of the form (1)–(4) be given with $\theta(i, k)$ as in (8). Let \mathbf{A}_0^i , \mathbf{A}_1^i , \mathbf{Q}^i , $i \in \{1, \dots, N\}$, be defined as in (9)–(10). If there exist a scalar $\alpha \in [0, 1]$ and matrices $\mathbf{P}_0^i, \mathbf{P}_1^i > 0$ solving the semidefinite program

$$\min \rho \quad (22)$$

$$\text{s. t. } \mathbf{P}_1^i - \rho I < 0, \quad \forall_i \quad (23)$$

$$\mathbf{P}_0^i - \rho I < 0, \quad \forall_i \quad (24)$$

$$\mathbf{A}_1^{i\top} \left(m_i \mathbf{P}_1^i + (1 - m_i) \mathbf{P}_0^i \right) \mathbf{A}_1^i - \mathbf{P}_1^i + \mathbf{Q}^i < 0, \quad \forall_i \quad (25)$$

$$\mathbf{A}_0^{i\top} \left(p_i \mathbf{P}_1^i + (1 - p_i) \mathbf{P}_0^i \right) \mathbf{A}_0^i - \mathbf{P}_0^i + \mathbf{Q}^i < 0, \quad \forall_i \quad (26)$$

where m and p are defined as in (21) and (20), then the scheduler (7) with the priority functions (17) makes the origin of each control system globally asymptotically stable. Moreover, the joint LQR cost is bounded by

$$\mathcal{J} < \rho \sum_{i=1}^N \|\mathbf{x}_0^i\|^2.$$

□

This follows from Thm. 1 together with the definition of \mathcal{P}_i in (16), as shown above. To confirm the upper bound on the control cost, we can verify that

$$\mathcal{J} < \eta_0^\top \mathcal{P}_{\sigma(0)} \eta_0 = \sum_{i=1}^N \mathbf{x}_0^{i\top} \mathbf{P}_{\sigma(0)}^i \mathbf{x}_0^i < \sum_{i=1}^N \rho \cdot \mathbf{x}_0^{i\top} \mathbf{x}_0^i,$$

due to (23) and (24).

Clearly, decomposing the scheduler to enable the use of priority scheduling in the network comes at the cost of suboptimal performance compared to a centralized scheduler

TABLE I
Size of optimization problem for our approach compared to [12].

Approach	#LMIs	size(LMI)	#vars
ours	$4N$	$2n \times 2n$	$2N(2n^2 + n) + 1$
[12]	$> (N+1)^2$	$4N(n+m) \times 4N(n+m)$	$> N^3(n+m)^2$

(14), as can be seen by comparing the performance bounds in Theorems 1 and 2.

In Table I we compare the size of the optimization problem in Thm. 2 with that proposed in [12] for a set of N systems, all with state dimensions n and input dimensions m , in terms of the number of LMI constraints, dimensions of LMI constraints, and number of scalar decision variables. While [12] co-designs a suboptimal controller in the process, our approach clearly remains more practical from a computational point of view as the number of participating NCS grows.

V. State-dependent Scheduler for $0 < q < N$

In the previous section, we used the special block diagonal structure of \mathcal{A}_i , \mathcal{P}_i , and \mathcal{Q} and a heuristic for Π to simplify the matrix inequalities in Thm. 1, and rewrite them as an optimization problem in Thm. 2. There, we made the specific assumption that $q = 1$. If we drop this assumption in favor of the generalization $0 < q < N$, then rewriting inequalities (13) in a similar fashion yields the inequalities (27)–(28) instead (see the top of this page). It is easy to verify that (18)–(19) are in a special case thereof.

As before, we can reduce this to a pair of inequalities for each NCS $i = 1, \dots, N$. For this purpose, the coefficients π_{jk} of the matrix Π must satisfy the following conditions:

$$\sum_{\{j | i \in S_j\}} \pi_{jk} = m_i, \quad \forall(i, k), i \in S_k \quad (29)$$

$$\sum_{\{j | i \in S_j\}} \pi_{jk} = p_i, \quad \forall(i, k), i \notin S_k \quad (30)$$

We assume again that the vector m is given, e.g., using (20). As Π now has additional degrees of freedom and its entries are not directly determined by m and p as they were in the previous section, we need to use a different heuristic to determine a feasible p . For instance, we can use the following optimization problem:

$$\min_{\Pi, p} \text{tr}(\Pi) \quad (31)$$

$$\text{s. t. } \pi_{ij} \geq 0, \quad \forall_{i,j} \quad (32)$$

$$\sum_i \pi_{ij} = 1, \quad \forall_j \quad (33)$$

$$\pi_{ii} \leq \rho(\mathcal{A}_i)^{-2}, \quad \forall_i \quad (34)$$

$$(29), (30), \quad (35)$$

where the first two constraints are required by Thm. 1. The third constraint is equivalent to the necessary condition that $\sqrt{\pi_{ii}}\mathcal{A}_i$ must be Schur-stable, which is also our reasoning behind minimizing the trace of Π .

With (29)–(30) the inequalities (27)–(28) become equivalent to the linear inequalities (25)–(26) in Thm. 2. This allows us to generalise the same result to an arbitrary queue length.

Theorem 3: Let $0 < q < N$ and a set of N control systems of the form (1)–(4) be given with $\theta(i, k)$ as in (8). Let \mathbf{A}_0^i , \mathbf{A}_1^i , \mathbf{Q}^i , $i \in \{1, \dots, N\}$, be defined as in (9)–(10). If there exist a scalar $\alpha \in [0, 1]$ and matrices $\mathbf{P}_0^i, \mathbf{P}_1^i > 0$ solving the semidefinite program (22)–(26), where m and p are defined as in (20) and (31)–(35), then the scheduler (7) with the priority functions (17) makes the origin of each control system globally asymptotically stable. Moreover, the joint LQR cost is bounded by $\mathcal{J} < \rho \sum_{i=1}^N \|\mathbf{x}_0^i\|^2$. \square

Analogous to the previous section, it is straightforward to verify that $\sigma(k) = \arg \min_i \eta_k^T \mathcal{P}_i \eta_k = \arg \min_s \sum_{i \in S_s} \mathbf{x}_k^{iT} (\mathbf{P}_1^i - \mathbf{P}_0^i) \mathbf{x}_k^i$. From there, the same line of argument as for Thm. 2 holds.

Interestingly, this shows that the scheduling problem for any queue length can be reduced to the same semidefinite programming problem, provided the parameters p are chosen appropriately according to the queue length.

Note that our previously defined scheduler requires both the plant state \mathbf{x}_k^i and the controller state $\hat{\mathbf{x}}_k^i$ to be known to calculate the priorities $v^i(\mathbf{x}_k^i)$. This can be accomplished by maintaining a copy of the controller state recurrence (2) at the sensor, for which $\theta(i, k)$ must be known, e.g., by sending acknowledgments from the controllers to the sensors, cf. [16]. However, acknowledgments would have to be delivered reliably, e.g., by reserving an additional virtual link. Also, sending an acknowledgment of size L_{ACK} introduces an additional delay, which must be modelled by modifying the propagation delay D , e.g., to $D' = 2D + \frac{L_{\text{ACK}}}{B}$ under the assumption of a symmetric duplex channel.

Alternatively, we can impose an appropriate structure on the optimization problem (22)–(26). If we add the following constraints

$$\mathbf{P}_1^i = \begin{bmatrix} X_1^i & Y^i \\ Y^{iT} & Z^i \end{bmatrix}, \quad \mathbf{P}_0^i = \begin{bmatrix} X_0^i & Y^i \\ Y^{iT} & Z^i \end{bmatrix}, \quad (36)$$

then the priority functions become $v^i(\mathbf{x}_k^i) = \mathbf{x}_k^{iT} (X_1^i - X_0^i) \mathbf{x}_k^i$, which depend only on the plant state. A similar approach is proposed in [26] in a different setup to design a switching output feedback controller. Of course, the corresponding optimization problem is more conservative. However, it allows us to achieve a higher bandwidth utilization than using acknowledgments.

VI. Evaluation

To demonstrate the feasibility of our approach, we ran experiments using real networking hardware with a set of simulated NCS and a proof-of-concept implementation of the priority scheduler. We also show a set of pure simulation examples to illustrate the relationship between queue size, bandwidth utilization and control cost.

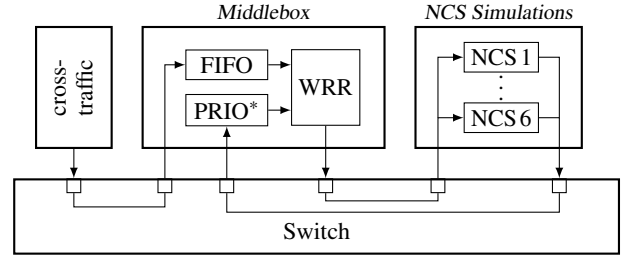


Fig. 2. Networking testbed setup used for evaluation. *) Priority scheduling and deficit round robin scheduling were implemented for comparison.

A. Proof-of-concept Implementation

We ran a proof-of-concept evaluation in a networking testbed consisting of commodity machines (Intel Xeon E5-1650) with 4×10 Gbps Ethernet network interface cards, connected to a 10Gbps Ethernet switch (Edge-Core AS5712-54X). The setup of our testbed is shown in Fig. 2.

A set of $N = 6$ NCS were simulated on one of the machines, with outgoing measurement packets from the simulated sensors of all NCS instances being sent over one network interface, and incoming packets being received on a different interface. In order to improve throughput and reduce unpredictable delays, we used Intel’s Data Plane Development Kit (DPDK) [32] instead of Sockets for sending and receiving packets from the simulation instances. We used identical models of an inverted pendulum on a cart for all 6 NCS simulation instances (for details see [33]). The sampling period was chosen at $T_s = 50$ ms. The pendulum for NCS 1 was started at an initial angle of $\phi = 35^\circ$, while all others were started at the origin $\phi = 0^\circ$.

A second machine was dedicated to producing cross-traffic approximately at line rate on a dedicated network interface. We use this cross-traffic to demonstrate that our proof-of-concept implementation realizes a dedicated virtual link to isolate NCS traffic from the effects of traffic from other applications, which is one of our initial assumptions.

A third machine was used to host a software middlebox implementation realizing both the virtual link provisioning and priority scheduling, also using DPDK for networking. The middlebox maintains two input queues: a priority queue with capacity $q = 2$ for NCS traffic, and a FIFO queue for all other traffic. The two queues are served in a weighted round robin fashion, where the priority queue is fully served every $T_s = 50$ ms and the FIFO queue is served during the remaining time. All outgoing traffic is transmitted on the same interface, but cross-traffic is discarded at the switch while NCS traffic is forwarded to the NCS simulation node. In order to compare our state-based dynamic scheduler to a fair static scheduler, we also implemented an alternative middlebox, with the difference that the queue for NCS traffic is not served using packet priorities, but using a deficit round robin (DRR) protocol.

We ran evaluations for both configurations: once using naive deficit round robin and once using our state-based priority scheduling for NCS traffic. Fig. 3 shows time series

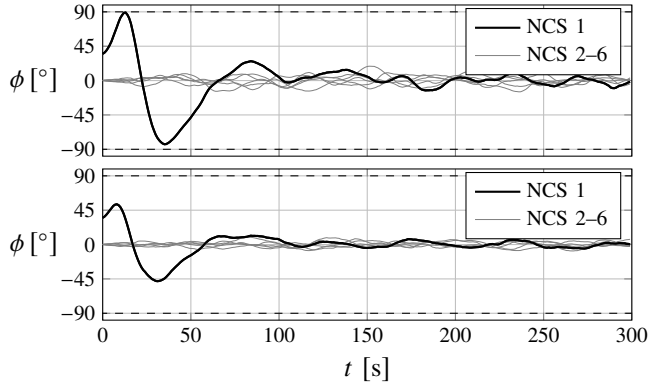


Fig. 3. Time series of pendulum angles from evaluation of proof-of-concept implementation with $N = 6$, $q = 2$, and $T_s = 50$ ms; top: round robin scheduling; bottom: state-based priority scheduling.

TABLE II
LQR cost comparison for proof-of-concept implementation

Scheduler	\mathcal{J}	J^1	J^2	J^3	J^4	J^5	J^6
Round-robin	73.0	48.0	5.3	4.3	3.8	7.3	4.3
Priority	23.0	13.7	2.1	1.8	1.8	1.5	2.1

for the angle ϕ of all six simulated pendulums, while Table II shows joint and individual LQR cost for comparison. Using state-based priority scheduling reduces the joint cost \mathcal{J} by 68% compared to round robin. Moreover, the cost is more evenly distributed between the NCS. While the stretch between the worst and best performing NCS is $\frac{\max_i J^i}{\min_i J^i} = 12.8$ with round robin, it is only 8.8 with state-based priority scheduling.

B. Simulation Example

To illustrate the effects of queue dimensioning, we show some simulation results for a fixed set of NCS and network model with varying q . We simulate a link with bandwidth $B = 10\,000 \frac{\text{bit}}{\text{s}}$ and delay $D = 20$ ms, that is shared by $N = 10$ NCS with identical (continuous) plant dynamics

$$\frac{d}{dt}x^i(t) = \begin{bmatrix} 0 & 1 \\ -2 & 2 \end{bmatrix} x^i(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u^i(t) + w^i(t),$$

initial conditions $x^i(t_0) = [1, 1]^T$, LQR controller for $Q = I$ and $R = 0.1$, and additive white noise $w^i(t) \sim \mathcal{N}(0, 10^{-3}I)$. We assume the packet size to be $L = 192$ bit, which corresponds to two IEEE double-precision floating point values for the state and one for the priority. The overall system with the state-based scheduler is simulated for $q = 1, \dots, N$ over a time period of 100 seconds. In each simulation, the systems are discretized to the minimum possible sampling time according to the link model, i.e. $T_s = \frac{L}{B}q + D$.

The results, averaged over 50 realizations of the simulation, can be seen in Fig. 4. The plot shows the joint LQR cost \mathcal{J} , together with the bandwidth utilization, which is given by the ratio of the transmission rate $\frac{qL}{T_s}$ to the available bandwidth B . Note that the case $q = 10$ corresponds to a static equal-bandwidth schedule. We can see that the cost decreases with

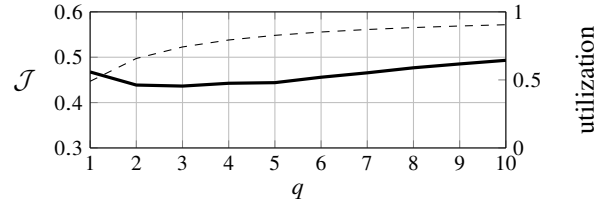


Fig. 4. Simulations for varying queue capacity q : joint cost \mathcal{J} (thick) and bandwidth utilization (dashed).

decreasing q , corresponding to more dynamic scheduling, but increases again towards $q = 1$. This can be attributed to a lower bandwidth utilization, as the propagation delay D dominates the sampling period for small q .

VII. Summary and Discussion

In this paper, we addressed the optimal scheduling problem for a set of NCS sharing a dedicated network slice. We introduced a switched model of the overall system with a limited-capacity queue model for the communication channel. Based on LMI stability conditions for switched linear systems from [26], we first designed a state-based priority scheduler for a channel capacity of one transmission per sampling period. We then generalised our scheduler design to allow an arbitrary number of NCS to transmit concurrently within one sampling period. The resulting scheduling policy guarantees performance and asymptotic stability of all NCS, and only requires stateless priority queuing in the network, making it both scalable and efficient to implement.

To conclude this work, we would like to discuss some aspects of our results. We designed a scheduler under the assumption that all NCS come with a given controller. Of course, joint optimal design of scheduling and controller, e.g. as in [12], is an interesting topic for future work. Furthermore, it could be studied how to choose the queue length for a given available bandwidth and network delay, such as to optimize the overall control performance. Also, while we considered priority scheduling for full state feedback here, the output feedback case can be studied by adding restrictions to the LMI constraints, as proposed in [26].

Our approach does not provide isolation between the traffic of individual NCS. While this allows us to utilize the available bandwidth to improve overall performance, it opens up the opportunity for one or more NCS to use more of their “fair” share to the detriment of all others, possibly to the point of instability. Apart from malicious priority inflation, modelling errors could also lead to this kind of behaviour. One possible solution is to use traffic shaping to limit the bandwidth available to each NCS. The consideration of additional shaping constraints is a topic for future work.

In our system model, we make some assumptions that should also be discussed briefly. First, sampling times and therefore packet arrival times of all NCS are synchronized. If this assumption is violated, i.e. sampling times of different NCS are phase shifted or packets experience different delays between sensor and scheduler, then any low-priority packet arriving early could impose an additional queuing delay on the

remaining NCS traffic and might ultimately cause a higher-priority packet to be dropped. Moreover, we did not account for priorities from different (overlapping) sampling periods to be compared at the scheduler.

Second, the communication channels of all NCS are modelled by one shared link. However, in a realistic application scenario, it should be assumed that the traffic of different NCS is routed over overlapping multi-hop paths. This means that a scheduling decision may be required at every hop. In principle, the same scheduler could be implemented at all network elements, since the priorities are preserved under scheduling of different packet subsets in an arbitrary order. However, even if the sampling times are synchronized, this is not necessarily true for the arrival times at schedulers in the network, which has been discussed above. Also, it may lead to suboptimal resource utilization, since unnecessary constraints may be imposed on traffic flows with mutually disjoint network paths. How our approach can be extended to address these issues is to be investigated in future work.

Acknowledgments

The authors would like to thank the German Research Foundation (DFG) for financial support, and the anonymous reviewers for their thoughtfulness and helpful suggestions.

References

- [1] J. P. Hespanha, P. Naghshabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, Jan. 2007.
- [2] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Automat. Contr.*, vol. 49, no. 9, pp. 1453–1464, Sept. 2004.
- [3] O. Imer, S. Yüksel, and T. Basar, "Optimal control of LTI systems over unreliable communication links," *Automatica*, vol. 42, no. 9, pp. 1429–1439, Sept. 2006.
- [4] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, Jan. 2007.
- [5] L. Schenato, "Optimal estimation in networked control systems subject to random delay and packet drop," *IEEE Trans. Automat. Contr.*, vol. 53, no. 5, pp. 1311–1317, Aug. 2008.
- [6] E. Garone, B. Sinopoli, and A. Casavola, "LQG control over lossy TCP-like networks with probabilistic packet acknowledgements," in *Proc. 47th IEEE Conf. Decision and Control (CDC)*, Cancun, Mexico, Dec. 2008, pp. 2686–2691.
- [7] Y. Mo, E. Garone, and B. Sinopoli, "LQG control with Markovian packet loss," in *Proc. European Control Conf. (ECC)*, Zurich, Switzerland, July 2013, pp. 2380–2385.
- [8] W. P. M. H. Heemels, A. R. Teel, N. van de Wouw, and D. Nesic, "Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance," *IEEE Trans. Automat. Contr.*, vol. 55, no. 8, pp. 1781–1796, Feb. 2010.
- [9] R. Blind and F. Allgöwer, "Is it worth to retransmit lost packets in networked control systems?" in *Proc. 51st IEEE Conf. Decision and Control (CDC)*, Maui, HI, USA, Dec. 2012, pp. 1368–1373.
- [10] B. W. Carabelli, F. Dürr, B. Koldehofe, and K. Rothermel, "A network abstraction for control systems," Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, Technical Report 2014/01, Apr. 2014.
- [11] S. Al-Areqi, D. Görges, S. Reimann, and S. Liu, "Event-based control and scheduling codesign of networked embedded control systems," in *Proc. American Control Conf. (ACC)*, Washington, DC, USA, June 2013, pp. 5299–5304.
- [12] S. Al-Areqi, D. Görges, and S. Liu, "Event-based networked control and scheduling codesign with guaranteed performance," *Automatica*, vol. 57, pp. 128–134, July 2015.
- [13] S. L. Dai, H. Lin, and S. S. Ge, "Scheduling-and-control codesign for a collection of networked control systems with uncertain delays," *IEEE Trans. Contr. Syst. Technol.*, vol. 18, no. 1, pp. 66–78, Jan. 2010.
- [14] A. Molin and S. Hirche, "On LQG joint optimal scheduling and control under communication constraints," in *Proc. 48th IEEE Conf. Decision and Control (CDC) and 28th Chinese Control Conf. (CCC)*, Shanghai, P.R. China, Dec. 2009, pp. 5832–5838.
- [15] C. Ramesh, H. Sandberg, L. Bao, and K. H. Johansson, "On the dual effect in state-based scheduling of networked control systems," in *Proc. American Control Conf. (ACC)*, San Francisco, CA, USA, June 2011, pp. 2216–2221.
- [16] C. Ramesh, H. Sandberg, and K. H. Johansson, "Design of state-based schedulers for a network of control loops," *IEEE Trans. Automat. Contr.*, vol. 58, no. 8, pp. 1962–1975, Aug. 2013.
- [17] A. Molin and S. Hirche, "Price-based adaptive scheduling in multi-loop control systems with resource constraints," *IEEE Trans. Automat. Contr.*, vol. 59, no. 12, pp. 3282–3295, Dec. 2014.
- [18] S. Shenker, C. Partridge, and R. Guerin, "Specification of guaranteed quality of service," Internet Requests for Comments, IETF, RFC 2212, Sept. 1997. [Online]. Available: <http://tools.ietf.org/rfc/rfc2212.txt>
- [19] F. Baker, C. Iturralde, F. L. Faucheur, and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 reservations," Internet Requests for Comments, IETF, RFC 3175, Sept. 2001. [Online]. Available: <http://tools.ietf.org/rfc/rfc3175.txt>
- [20] N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862–876, Apr. 2010.
- [21] G. C. Walsh, H. Ye, and L. G. Bushnell, "Stability analysis of networked control systems," *IEEE Trans. Contr. Syst. Technol.*, vol. 10, no. 3, pp. 438–446, May 2002.
- [22] D. Nesic and A. R. Teel, "Input-output stability properties of networked control systems," *IEEE Trans. Automat. Contr.*, vol. 49, no. 10, pp. 1650–1667, Oct. 2004.
- [23] L. Greco, D. Fontanelli, and A. Bicchi, "Design and stability analysis for anytime control via stochastic scheduling," *IEEE Trans. Automat. Contr.*, vol. 56, no. 3, pp. 571–585, Mar. 2011.
- [24] R. Blind and F. Allgöwer, "Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process," in *Proc. 54th IEEE Conf. Decision and Control (CDC)*, Osaka, Japan, Dec. 2015, pp. 7510–7515.
- [25] O. Gettings, S. Quinton, and R. I. Davis, "Mixed criticality systems with weakly-hard constraints," in *Proc. 23rd Intl. Conf. Real Time and Networks Systems (RTNS'15)*. Lille, France: ACM, Nov. 2015, pp. 237–246.
- [26] J. C. Geromel, P. Colaneri, and P. Bolzern, "Dynamic output feedback control of switched linear systems," *IEEE Trans. Automat. Contr.*, vol. 53, no. 3, pp. 720–733, Apr. 2008.
- [27] 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 Trans. Networking*, vol. 1, no. 3, pp. 344–357, June 1993.
- [28] H. Lin and P. J. Antsaklis, "Switching stabilization and l_2 gain performance controller synthesis for discrete-time switched linear systems," in *Proc. 45th IEEE Conf. Decision and Control (CDC)*, San Diego, CA, USA, Dec. 2006, pp. 2673–2678.
- [29] —, "Stability and stabilizability of switched linear systems: A survey of recent results," *IEEE Trans. Automat. Contr.*, vol. 54, no. 2, pp. 308–322, Feb. 2009.
- [30] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-Time Markov Jump Linear Systems*. Springer, 2005.
- [31] G. S. Deaecto, M. Souza, and J. C. Geromel, "Discrete-time switched linear systems state feedback design with application to networked control," *IEEE Trans. Automat. Contr.*, vol. 60, no. 3, pp. 877–881, Mar. 2015.
- [32] Intel®, "Data Plane Development Kit (DPDK)." [Online]. Available: <http://dpdk.org>
- [33] S. Zinkler, "In-network packet priority adaptation for networked control systems," Master's thesis, University of Stuttgart, Aug. 2016, code available at <https://github.com/znsn/masterthesis>.