IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 December 2022, accepted 24 January 2023, date of publication 8 February 2023, date of current version 15 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3243286

RESEARCH ARTICLE

Data Flow Control for Network Load Balancing in IEEE Time Sensitive Networks for Automation

THOMAS WEICHLEIN[1], (Member, IEEE), SHUJUN ZHANG[1], PENGZHI LI[1], (Member, IEEE), AND XU ZHANG[2]

[1]Cyber and Technical Computing, University of Gloucestershire, GL50 2RH Cheltenham, U.K.
[2]Boldrewood Innovation Centre, University of Southampton, SO16 7QJ Southampton, U.K.

Corresponding author: Thomas Weichlein (thomasweichlein@connect.glos.ac.uk)

ABSTRACT IEEE time sensitive networks (TSN) offer redundant paths for automation networks that are essential preconditions for network load balancing (NLB) or distribution. They also provide several traffic shapers and schedulers with different impacts on the data flow control. The selection of the right traffic shaper or scheduler for an automation network is challenging. Their influence depends on various network parameters such as network extension, network cycles, application cycles, and the amount of data per traffic class and network cycle. In this study, data flow control for NLB in automation TSN using different traffic shapers and schedulers was investigated. The effects of the network parameters on the shapers and schedulers were derived and imported into the data flow control model of the automation network. The sample networks were simulated, and performance comparisons were made. The results show that the enhancements for scheduled traffic (EST), strict priority queuing (SPQ), and the combination of SPQ with frame preemption (FP) are better scheduler selections in connection with larger networks, fast network cycles, and fast application cycles. The cyclic queuing and forwarding (CQF) shaper and asynchronous traffic shaper (ATS) are rather an alternative for load control in small networks or in conjunction with slow applications.

INDEX TERMS Automation networks, data flow control, load balancing, time sensitive networks.

I. INTRODUCTION

The continuously increasing communication demand in the industry has resulted mainly from the ''Industry 4.0'' industrial revolution. This implies a significant expansion in the digitalisation of the production process and vertical communication connectivity from cloud-based servers down to the sensor level in an industrial plant. This increase implies not only a growing demand for data volume and communication speed, but also a higher need for reliable and deterministic data transport. These developments have led in a first step to the develoment of the Audio Video Bridging (AVB) standard [1] and finally to the creation of a ''Time-Sensitive Networks (TSN)'' [2], [3] Task Group (TG) as part of the IEEE 802.1 Working Group (WG). TSN is defined by the associated IEEE standards extending the IEEE 802.1 standard [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] and is still under further development. AVB and TSN define various new functionalities and different traffic shapers and schedulers, such as the credit based shaper (CBS) [6], the enhancements for scheduled traffic (EST) [8], the cyclic queuing and forwarding (CQF) [11], the asynchronous traffic shaper (ATS) [13], the strict priority queuing (SPQ) [5], and frame preemption (FP) [7], to achieve highly efficient and deterministic data transport. TSN also allow for the use of multiple communication paths, primarily to provide seamless media redundancy according to IEEE 802.1CB [32], which defines ''Frame Replication and Elimination for Reliability (FRER)''. Classical, non-TSN networks for Internet or campus communications, both wired and wireless, are typically set up as multi-paths networks. In addition to the advantages of redundancy, the availability of multiple paths has led to the use of load-sharing and load-balancing concepts since the late 1990s. These non-TSN networks are usually

The associate editor coordinating the review of this manuscript and approving it for publication was Cesar Vargas-Rosales.

based on Open Systems Interconnection (OSI) layer 3 routing technology. Their dedicated load balancing methods include methods for ISP networks [15], campus networks [16], or access networks for mobile connectivity [17]. In factory automation applications, networks with smaller spatial extension are used to transport information between automation controllers (AC) and devices, such as drives, sensors, and actuators. They are typically based on the OSI layer 2 technology using switching.

To achieve redundant connections with minimum wiring effort, ring topology has become a prevalent topology in redundant industrial automation networks. Fig. 1 shows a typical industrial automation network setup, where several field-level rings are redundantly coupled to a controller-level ring. This, in turn, is redundantly coupled to a higher-level Information Technology (IT) or Operational Technology (OT) network [18]. Controller-level rings usually contain a variety of higher-level ACs such as programmable logic controllers (PLC) or motion controllers (MC). However, the field-level ring typically consists of only one AC which controls a variety of automation devices, such as drives, sensors, actors, or decentral peripherals providing digital and analogue inputs and outputs. Field-level ACs communicate with the controller-level ACs. To date, communication connections in automation networks typically have been set up redundantly primarily because of failure-safety rather than load sharing. Until now, there has been limited theoretical research and practical application work on efficient and effective load sharing and load balancing over multiple paths of TSN, which is particularly true in industry. Therefore, manufacturing automation networks, especially the more recent TSN automation networks, offer new grounds for research on network load distribution, which can be expected to contribute to enhancing the performance of these networks. Comparing legacy layer 3 networks, three major load distribution strategies, also known as traffic engineering concepts, are visible [15]:

1) oblivious routing, that is, routing on a fixed scheme without incorporating changes in the network load along various data paths;
2) traffic control using predicted traffic demands based on recorded traffic history;
3) adaptive or dynamic control using metrics on traffic load along the available data paths.

The first two methods, oblivious and predictive traffic control, are advantageous particularly in conjunction with uncertain network demand estimations. However, the communication demands of automation applications are comparably well-defined and predictable within certain limits in the network setup. Because of this advantage, they are suitable candidates for traffic distribution planning during the network setup phase in conjunction with adaptive or dynamic traffic control at runtime. Adaptive or dynamic control is based on routing decisions based on network-load metrics. A control algorithm constantly controls the load distribution on several paths to achieve an optimal or near-optimal load distribution and to minimize the local load maxima.

Dynamic load control, as has been investigated in previous research on layer 3 ISP networks or campus networks [15], [17], [20], [21], usually differentiates between:

- Flow control: The algorithm for controlling the data flow on a single path to increase or decrease throughput;
- Fairness control: This algorithm regulates the fair distribution of the reduction or increase in throughput among different data flows;
- Distribution control: An algorithm for allocating parts of a stream or several streams evenly to a number of paths.

Fairness control is of minor importance for control data (CD) of automation networks, as explained later in Section IV. Regarding distribution and flow control, more research has been conducted on distribution control [15], [17], [20], [21] than on the flow control subtask. Some network characteristics play a crucial role in flow control. One important influencing factor is the cycle time of the automation application's tasks hosted by ACs. These application cycle times, sending data at each cycle, limit attainable control performance, that is, the time to establish a new load distribution setpoint. The other is the underlying basic cycle time of the network communication, which must be long enough to transport the maximum amount of data but small enough to serve the fastest application. Furthermore, the network extension and applied traffic shaper or scheduler influence the path delays or latencies which represent dead-time elements that characterize the flow control circuit properties. In this study, data flow control for control data within TSN automation networks was investigated under the influence of different application cycles, communication cycles, network extensions, and traffic shaping and scheduling mechanisms. The influence of these parameters on the control dynamics and stability was analyzed. Furthermore, the impact of bandwidth reservation was investigated and recommendations for load measurements are provided.

II. RELATED WORK

Various control methods have been used for data load distribution control, to achieve a balanced load on OSI-layer 3 networks. Examples of ISP networks and campus networks include common-case optimization with penalty envelope (COPE) [15], multiprotocol label switching (MPLS) adaptive traffic engineering (MATE) [16], or traffic engineering explicit congestion protocol (TeXCP) [22]. Various controller types such as linear [15], [16], [22], [23], predictor-based [23], fuzzy [24], [25], ant colony algorithm [26], and stochastic control [21] have been deployed. Elwalid et. al. [16] introduced MATE, an adaptive traffic engineering concept in MPLS networks for best effort (BE) traffic. They used label switched path (LSP) statistics gathered by probe packets in the ingress node to determine the most loaded path. This load was decreased until the path loads were equalized by using an extended gradient projection algorithm, averaging several load measurements over a period of

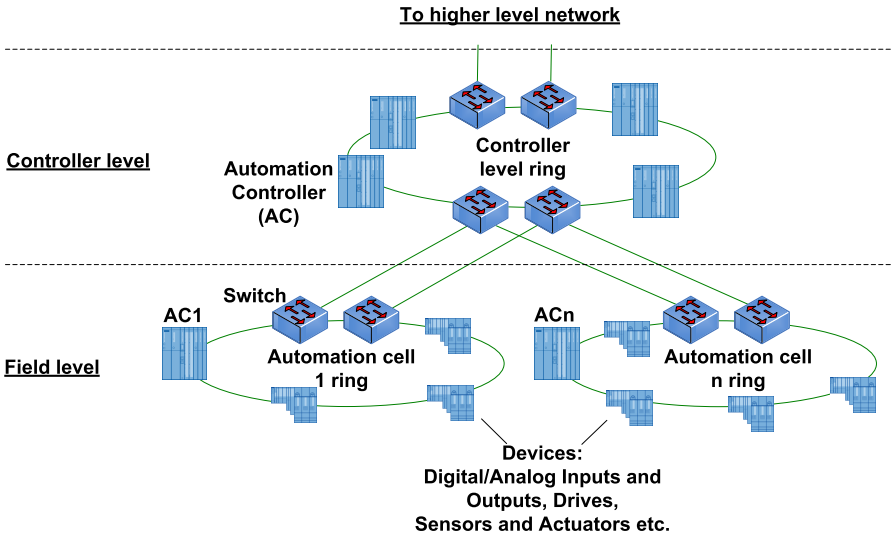


FIGURE 1. Typical automation communication network setup.

time to compensate for asynchronism. Kandula et. al. [22] used a linear control method called TeXCP which is applied within classical Ethernet. They measured the maximum load of several paths by sending probe frames from the TeXCP agent on the source side, which were then sent back by the edge router on the target side. A linear control algorithm decreased the load on the path with the current maximum load. It was shown that the load measurement path delays can be neglected for longer control cycles. Yu-Jia et. al. [27] applied dynamic load balancing via software-defined networks (SDN) in machine-to-machine (M2M) networks. They worked with a quality of service (QoS) violating threshold, where the network delay is derived from the measured load of the data sink, which is represented by a network service capability layer in the accessed server. The actual control algorithm implemented is an on-off controller rather than a linear controller such as a proportional–integral–derivative (PID) or predictor-based controller. The actual path delay times were not considered in the model. A further example using an on-off controller algorithm which is applied to wireless network load balancing between the LTE path and WLAN path, was provided by López-Pérez et al. [20]. They used a two-step decision algorithm to determine between the two possible paths from which they previously estimated the delay. This is achieved using Little's law $\delta = B/R$, where δ is the estimated delay, B is the mean number of bits in the system and R is the throughput of the system. B and R were obtained from the network status reports. Here, the delay information is used for the path selection decision but is not used for control stability considerations.

However, research projects on distribution or flow control in the context of **TSN** are exceptions. Nayak et. al. [28] investigated scheduling and routing possibilities using an IEEE 802.1Qbv EST traffic shaper with a central network controller. Ojewale and Yomsi [29] proposed two heuristics for routing flows for TSN distribution control but did not consider the influence of path delays introduced by different types of shapers. Instead, a user defined common proportional factor for path length was applied. Arif and Atia [30] provided a mathematical model for load-balancing routing in a general TSN by estimating long-term average path delays. However, the influence of different possible traffic shapers defined by IEEE TSN is not considered. Nasrallah [31] compared the performance of EST with ATS and introduced an adaptive bandwidth-sharing mechanism for EST, where the gating window size was adapted to the traffic load using a control algorithm. Zhao et. al. [34] provide a quantitive performance comparions of various traffic shapers. Although these two works do not directly cover load balancing, they are still of interest for this research because they provide detailed insights into the performance and behaviour of various traffic shapers and scheduler.

III. TSN AUTOMATION NETWORKS

The communication data for automation [19] can be classified as control data (CD) or noncontrol data (Non-CD). CD can be transferred either synchronized or unsynchronized and is always cyclic data, so-called ''streams'', within the TSN. Synchronized CD are also referred to as isochronous CD (I-CD) and offer the lowest guaranteed latency from the talker to the listener. Unsynchronized CD is also referred to as nonisochronous CD (NI-CD), and typically offers bounded low latency. I-CD is used for fast, highest precision control loops with I-CD cycle times from a few milliseconds to tens of microseconds. NI-CD is used for slower control loops with cycle times ranging from a few hundreds of milliseconds to a few milliseconds. Examples of non-CD include configuration, diagnosis, and monitoring data. Non-CD has typically no special timing requirements and is also named ''Best Effort'' (BE) data. Both I-CD and NI-CD can either be sent on a single path, also called nonseamlessly, or twice on two disjoint paths, also called seamlessly, because of

seamless media redundancy, according to IEEE 802.1 CB [32]. In the case of an automation ring topology, the two paths are represented by the two directions around the ring. For CD, load balancing makes sense only in conjunction with the singly transferred data. This is because the doubly transferred CD on the two paths can only be controlled by ingress limiting and not by traffic redirection. Ingress limiting is not an option for CD though because CD are subject to tight timing restrictions for transmission. Seamless and nonseamless I-CDs and NI-CD in TSN networks are typically separated by virtual local area networks (VLAN). Seamless CD contributes to the basic load of non-load-controllable data. Nonseamless CD with higher bandwidth consumption are available for load control. Nonseamless CD of low bandwidth consumption, such as sensor data, are typically unsuitable for load control because of their low influence. They also contribute to the basic load of non-load-controllable data.

The TSN TG defines a variety of traffic shapers and schedulers to influence the data transfer for CD which are briefly introduced here. A more detailed description is provided by Lo Bello et. al. [3].

The **strict priority queuing or static priority queuing (SPQ)** assigns eight different QoS properties to various data classes. It is known from traditional Ethernet switch ASICs defined in [3], and is used in general layer 2 networks and automation networks. It is also used in TSN. The SPQ provides one egress queue for each or for a selection from the eight QoS frame priorities. For I-CD data, it is common to use the highest or, in cases where management frames should have a higher priority, the second highest priority. For the NI-CD, the next lowest priority is used. The SPQ gains attractiveness when combined with FP for the highest-priority traffic class, thereby forming an express traffic class. SPQ is advantageous, particularly for higher bandwidth systems of 1 Gbit/s and above. The reason for this is, that with a higher bandwidth, the relevance of the maximum frame length transmission time, which can block the egress port, decreases.

The **credit-based shaper (CBS)** was introduced with IEEE 802.1Qav [6] mainly for the purpose of transferring audio/video data without bursts and congestions. The main feature of the CBS is that it stretches data bursts to achieve a continuous flow of the stream. Therefore, it is not suitable for CD as CD are intentionally sent in bursts by the AC at the beginning of a new application cycle. Therefore, it was not considered here for the application and analysis of automation networks.

The **enhancements for scheduled traffic (EST)**, defined by IEEE 802.1Qbv [8], also known sometimes as time-aware shaper (TAS), assign gating windows to traffic classes. Each traffic class send queue is then emptied at a defined time slot, called gating window or gate-open window, which is repeated in every network cycle. EST can be used to achieve synchronized gating times in all bridges of the TSN domain, with no other data interfering with the transmission during the gating window. This guarantees the unhindered transfer of data traffic and minimum network latency through the complete, synchronized EST network domain. If a synchronized talker sends synchronized to the beginning of the gating window, minimum network latencies can be achieved.

The **cyclic queuing and forwarding (CQF)** traffic shaper, defined by IEEE 802.1Qch [11], also follows a global network domain cycle. It stores the ingress traffic during one network cycle and forwards it in the next network cycle. Through this method, a certain amount of data traffic is handed from one bridge to the next, taking one hop per network cycle. Thus, with CQF, limited latency can be guaranteed which depends on the maximum number of hops in the CQF network domain. The latency per hop is identical to the length of the network cycle. The amount of admissible data per cycle depends on the configuration of the cycle length and can be restricted by reservation and ingress limits. Gating windows for the data of further traffic classes to be transported in parallel, extend the necessary network cycle.

The **asynchronous traffic shaper (ATS)** [31], [33], defined by IEEE 802.1Qcr [13], provides additional, shaped egress queues which feed the existing classical egress queue structure, as known from SPQ. The processing chain for a stream with ATS consists of per-stream filtering and policing (PSFP), shapers, egress queues, transmission selection, and gate control. An internal priority value (IPV) can be assigned to each traffic class within a bridge. The IPV is independent of the frame's tagged priority and does not influence it either, on its way through the bridge. It allows for dedicated prioritized frame handling per hop and traffic classes. ATS does not depend on synchronous bridges or synchronous communication and offers bounded latency for lower-performance control data such as NI-CD. The ATS shaper mechanism functions as a token bucket traffic shaper, which limits bursts to configurable sizes.

Frame preemption (FP), originally defined by IEEE 802.1Qbu [7], is another TSN feature in which streams are classified as either express or preemptable traffic. Express traffic can interrupt the transmission of a preemptable frame, and thus overtake preemptable frames. After the express frame is transmitted, preemptable frame transmission is resumed. Logically, only one traffic class can be classified as expressing traffic without spoiling the intention of preemption. The preemption feature, which is now integrated into IEEE 802.1Q [3] for the MAC layer, strongly correlates with the definitions of IEEE 802.3br [35] for interspersing express traffic (IET) for the physical layer (PHY). Preemption can basically be applied in SPQ environments, but can, in principle, be also combined with EST, CQF and ATS shapers.

Stream reservation (SR) is another crucial feature offered by 802.1Q which can be used in the TSN domain. It is defined by the multiple reservation protocol (MRP)/multiple stream reservation protocol (MSRP) [5] and the currently emerging resource allocation protocol (RAP) [14]. SR in combination with automation networks is mostly used as overload protection for the network because excess streams will not receive bandwidth reservation in the bridges. To protect against congestion, that is, against talkers which exceed

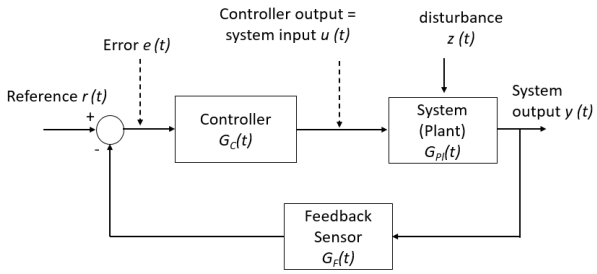


FIGURE 2. General control system.

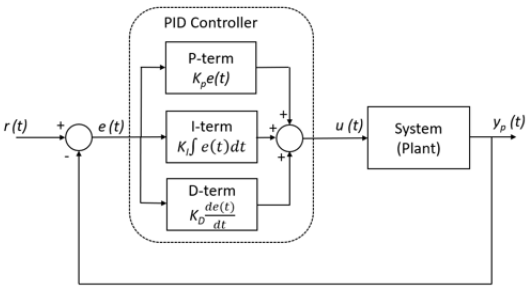


FIGURE 3. Elements of a PID controller.

their reserved bandwidth, an ingress limiter, as defined by IEEE 802.1Qci [12], can be deployed as supplementary protection.

Industrial automation TSN standardisation efforts [19] currently define particularly the application of EST, SPQ, FP, and SR. Nevertheless, also the CQF and ATS are potential TSN shaper candidates for industrial automation networks. The various shapers and schedulers have a different influence on network traffic distribution control though.

IV. NETWORK TRAFFIC DISTRIBUTION CONTROL

The dynamic control of the network load distribution demands the application of control engineering, which is shortly introduced here. Fig. 2 shows a general abstracted control system. It consists of three major parts:

1) The system under control, in control engineering terminology also called ''the plant'', with its output $y(t)$ to be controlled. In the case of NLB the system is the network and the output is the traffic load.
2) A sensor module to feed back the systems output value to be compared with a reference value $r(t)$, which represents the wished output value.
3) The controller to control the systems input $u(t)$ based on the difference between the reference and the feedback of the output.

The control theory defines linear and non-linear systems. TSNs are typically linear systems as the measured mean of a traffic load changes linearly. Both the controller and the system consist of typical control elements. These are either proportional (P), time-constant (T), differential (D), or integral (I) elements. Furthermore, systems cause delays for outputs or inputs which are called dead-time elements. The traditional controller type for linear systems, where the dead-time is small compared to the inherent time constants, is the PID controller. It consists of a proportional, an integral, and a differential element as depicted in Fig. 3. The optimal design of the controller can be reached by an optimized application of the controller parameters which are the proportional amplification K_P, the integral amplification K_I, and the differential amplification K_D. The controller parameters must be optimized for the different types and extensions of TSN as these change the systems characteristics, such as the delay times, that is, the system dead times from control theory point of view.

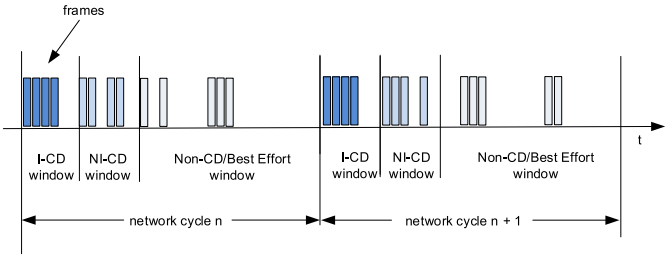
Different mathematical methods can be used to analyze dynamic systems. The traditional approach is the analysis in the time domain using differential equations. The Fourier transform and the Laplace transform can be used to transform the often complicated differential equations of the time domain to achieve simpler mathematical equations and solutions in the frequency domain. We use the Laplace transform to describe the TSN system.

Important quality criteria for a control are dynamic performance and stability. The dynamic performance, or performance for short, here means the time required to bring the output back to a constantly stable position after a reference change, that is, a change in network traffic load which must be distributed. Stability or also stability reserve here means the reserve in the controller and system parameters before uncontrolled output changes occur, e. g. in the form of output oscillations. To determine the dynamic performance of a control circuit it is common to simulate an input step of the reference $r(t)$ and to plot the output $y(t)$ step response to measure the settling time T_S needed for the output to settle to the reference value.

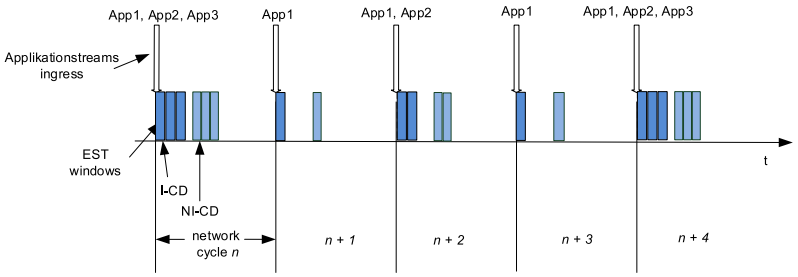
The stabiity or stability reserve of the control circuit can be determined using the Nyquist diagram. This is the system frequency response plot of the magnitude and phase of the open loop system output $G_0(s)$ over a frequency sweep of the sinusoidal input signal $\mu(t)$. Both, step response and Nyquist diagram are produced using the MATLAB/Simulink simulation tools.

A further property of a control circuit is the quality of the control J which is the deviation of the output value compared to the input value. For example, it is given as an integral of the time-weighted absolute error (ITAE) value with $J = \int_0^\infty |e(t) - e(\infty)|\, t\, dt$, where e is the control deviation, that is, the deviation of the actual value from the setpoint or reference of the control.

For NLB, the network traffic load is the system output to be controlled. It consists of automation applications CD and non-CD. Automation applications typically [19] demand a variety of different application communication cycles, in which automation tasks, such as a temperature control, a speed control, or a digital or analogue input/output control, are processed cyclically. These application cycles are determined by various automation application requirements that operate on one or more ACs in the network. Each application



(a) Succession of network cycles with I-CD, NI-CD and Non-CD traffic in EST windows



(b) Succession of frame ingress from applications with different application cycles

FIGURE 4. Relations between communication cycle, EST windows, and application cycles.

has its own requirements for the communication speed with peripheral devices or other ACs. For instance, a slow temperature controller application might exchange the setpoint and actual value with an analogue I/O card every 500 ms. On the other hand, a fast speed controller application might need to exchange setpoint and actual value with a drive in cycles of a few microseconds. Typically, data exchange between the application on the AC and a connected device occurs once at the beginning of the application cycle in both directions, to transport setpoints or references and actual values.

Besides the application cycle, nearly all types of TSN based network types, except for SPQ and ATS, provide a network cycle time. This uses the timing information of the bridges among each other to synchronize the data transport throughout the TSN domain. Thus, particularly with EST, a minimum of latency is achieved. The network cycle time is determined by the shortest application cycle in the network domain and may not be longer than this. Fig. 4 shows how application cycles and network cycles are correlated within an automation network based on the EST traffic scheduler. Fig. 4 (a) shows a snapshot of two network cycles with EST windows for I-CD, NI-CD and Non-CD or BE traffic classes. The I-CD data transport is fed in synchronized to the network cycle and its EST window start. It is transported immediately at window start without any delay. NI-CD and Non-CD queues are emptied at window start and there might come additional unsynchronized data during the window duration time. Fig. 4 (b) shows a succession of five network cycles and three example applications App1, App2, and App3 feeding in I-CD streams and NI-CD streams of about the same size. Network cycle length, EST window length, and application cycles are important parameters for the traffic delay, that is, for the dead time elements, and for the time constants for NLB.

The classification of the general term load control within classical OSI layer 3 networks into more specific terms of flow control, fairness control and distribution control is also sensible for load control in automation networks. Fairness control is of secondary importance for automation networks. This follows from the fact that the proportion of timely rather uncritical data flows of non-CD, whose throughputs could be evenly reduced, such as TCP/IP flows, is low. Instead, automation networks must part time-critical streams onto different paths without being allowed to reduce the overall throughput of the stream. Therefore, congestion control, where ingress data are either dropped or the sender is informed to reduce the throughput, is not an option for automation-data traffic such as CD.

This study focused on the flow control of CD in TSN automation networks. The aim is to analyze the influence of different traffic-shaping mechanisms, the application cycles, and the network extension on the dynamic performance and stability of the control circuit.

CD in TSN consist of data traffic streams transmitted by the automation applications. Fig. 5 shows a section of an abstracted fully meshed automation network. It is represented by the graph $G = (V, E)$ with a set of vertices $V(G)$ and a set of edges $E(G)$. The set $V(G)$ represent the nodes v_i of the graph. In automation networks, these are either pure network switches or automation devices with integrated switches. Set $E(G)$ with edges e_{ij} represents the links between nodes v_i and node v_j. The number of nodes in the graph determines its

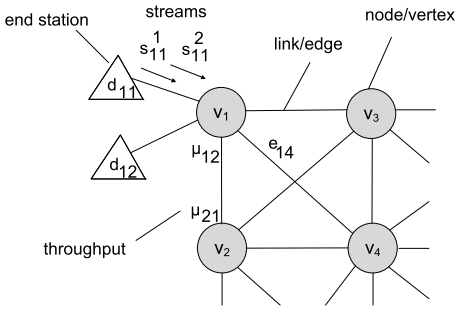


FIGURE 5. Abstracted TSN automation network.

order n. The number of edges connected to a node determines its degree $deg(v)$. Let $\boldsymbol{D_i} = \{d_{i1}, \ldots, d_{ij}\}$ be a set of devices (end stations) connected to the node $v_i \in \boldsymbol{V} = \{v_1, \ldots, v_n\}$. Let furthermore be $\boldsymbol{Ta_{ij}} = \{ta_{ij}^1, \ldots, ta_{ij}^k\}$ a set of Talkers within d_{ij} and let $\boldsymbol{Li_{ij}} = \{li_{ij}^1, \ldots, li_{ij}^p\}$ be a set of Listeners within d_{ij}. $\boldsymbol{Ta_{ij}}$ create a set of streams $\boldsymbol{S_{ij}} = \{s_{ij}^1, \ldots, s_{ij}^q\}$ being sent to v_i. The paths which the streams can take from a Talker ta_{ij}^k to one or more Listeners li_{ij}^p located somewhere in the network, are derived from automation applications running in the devices d_{ij}. The sum of the directed streams on link e_{ij} creates a throughput μ_{ij} at the output port of node v_i. Each link e_{ij} provides two scalars of throughputs μ_{ij} and μ_{ji} which represent the current output data rates at node v_i in the direction of v_j and vice versa. Thus, the edges describing the throughputs are directed edges. The individual throughputs μ of the all links in the network can be formed as an instance

$$\boldsymbol{M} = \begin{bmatrix} \mu_{11} & \ldots & \mu_{1n} \\ \ldots & \ldots & \ldots \\ \mu_{n1} & \ldots & \mu_{nn} \end{bmatrix} \tag{1}$$

of a distance matrix of graph G, where n is the order of the graph, which represents the number of nodes within the network domain. Automation applications with redundant networks are implemented almost without exception in the ring topology, as illustrated in Fig. 6. The throughput distance matrix M for a ring topology is reduced to a doubly diagonal filled matrix, provided that the nodes of the ring are numbered clockwise or counterclockwise in succession. For example, M for a ring of five nodes results in

$$\boldsymbol{M} = \begin{bmatrix} 0 & \mu_{12} & 0 & 0 & 0 \\ \mu_{21} & 0 & \mu_{23} & 0 & 0 \\ 0 & \mu_{32} & 0 & \mu_{34} & 0 \\ 0 & 0 & \mu_{43} & 0 & \mu_{45} \\ 0 & 0 & 0 & \mu_{54} & 0 \end{bmatrix}. \tag{2}$$

The ring nodes v_i provide the measured throughputs on their ring ports as feedback for flow control within the ring. Owing to various applications with talkers ta_{ij}^k connected to the ring nodes v_i and possible inter-ring communication $s_{interlink}$, the individual link throughputs along a path from a controller talker to listeners can be different. The distribution control task using one of either paths of the ring results from the

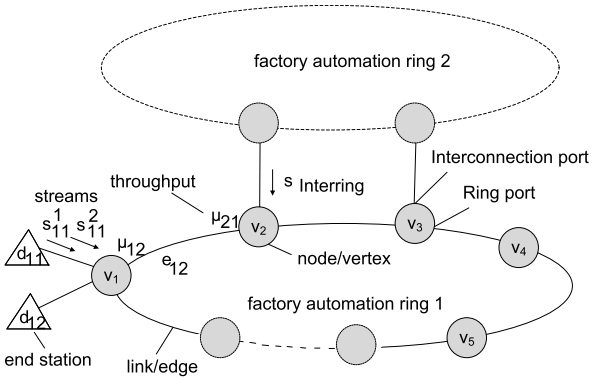


FIGURE 6. Automation ring graph.

optimization task

$$\begin{aligned} &\min \max_{i,j \in V} \quad \mu_{ij}, \\ &\text{subject to:} \quad \forall e \in E(G), \end{aligned} \tag{3}$$

to minimize the maximum throughput on the single links on the available paths. Data flow control is a subtask of distribution control within an influential controller (AC). The task is to reduce the load on a certain path and shift it to an alternate path, either completely or partially. An influential controller is an AC which transmits sufficient data which can be redirected to contribute to a significant change in load distribution. This flow control task in switched layer 2 automation TSN under the influence of different shapers and application cycles is the focus of this study.

V. DATA TRAFFIC FLOW CONTROL IN TSN FOR AUTOMATION

The selection of the flow control method strongly depends on the characteristics of the data traffic and network properties. Automation networks and automation-specific data have different characteristics, see IEC/IEEE 60802 TSN profile [19], from those of ISP or campus networks. First, they are based on Ethernet OSI layer 2 traffic switching, rather than OSI layer 3 traffic routing. Second, the type of data traffic differs because the data frames are typically smaller. Data-transport intervals are much faster, and data traffic is often generated in bursts instead of a homogenous distribution over time. On the other hand, an automation plant has a manageable and known extension unlike an ISP network. Therefore, the data traffic volume is known or at least more predictable. Unlike ISPs, network properties can be calculated in terms of the sum of bridging delays and LAN propagation delays along a defined network path from the talker to the listener. These preconditions suggest the application of linear dynamic control rather than oblivious or predictive traffic control which is more appropriate under uncertain conditions. ACs d_{ij}, hosting talkers $\boldsymbol{Ta_{ij}}$ which create stream sets $\boldsymbol{S_{ij}}$, typically have a variety of applications running with different send cycles. With a direct and immediate link load or throughput measurement and its feedback and control calculation on the network cycle speed, the controller output oscillates with the interference of

all different application cycle data transmissions. Moreover, it creates a considerable central processing unit (CPU) load on the AC to calculate the control loop in every network cycle which is usually selected within the range of 100 μs to 4 ms. Its length depends on the applied traffic-shaping method and applications. Furthermore, it is difficult to collect all the actual values of the throughputs at each link in the network within one network cycle. Therefore, the mean throughput at a link must be measured over a suitable time-span. It is evident, that this time span T_{mean} minimum length is determined by the slowest application cycle T_{App} that sends data over the TSN domain. Under these conditions, we propose the following calculation.

$$T_{mean} \geq m\,(\max_{i} T_{Appi}), \tag{4}$$

where T_{mean} is the recommended integration time for the calculation of the mean link load or throughput. It should therefore be m times larger than the largest application time T_{App}. Parameter m is an empirical factor which should be selected sufficiently long to smoothen local peaks, but sufficiently short to reach sufficient control dynamics. For the simulations of this research task, m was chosen as 5, which seems to be a reasonable starting point. T_{Appi} is the application cycle of all the applications in the network domain.

Fig. 7 depicts the control structure according to Fig. 2 of a single network path within the plant in the form of a control block diagram. The blocks contain their transfer functions in the Laplace transform representation. The control structure consists of all network hop latency times and LAN propagation delays T_{DT1} to T_{DTn} that appear as dead-time elements and form the plant $G_{Pl}(s)$. The controller is designed as a PID-Controller for the reasons mentioned above. The feedback path contains a PT1 (proportional with one time constant) element $G_M(s)$ caused by the rolling mean calculation of the feedback. It further contains in the feedback path all hop dead-time elements caused by the transition time of the feedback data from the relevant link back to the flow controller in the AC.

This model of the control structure is valid for all forms of TSN. Different TSN only change the size of the dead times and time constants in the following way:

1) The use of different shapers or schedulers change the dead time elements T_{DT1} to T_{DTn} in the plant and in the feedback path;
2) The network extension changes the number of the network hops and thus the number of the dead time elements in the plant and in the feedback path;
3) The longest of all application cycles in the TSN domain assignes T_{mean} according to (4) und thus the PT1 time constant T_M.

The relevant position of the link with the current maximum throughput, and thereby the number of hops between the controller and link, is determined by the maximum throughput of all links along the path from the talker to the last listener. The controller was designed as a real PID controller, that is,

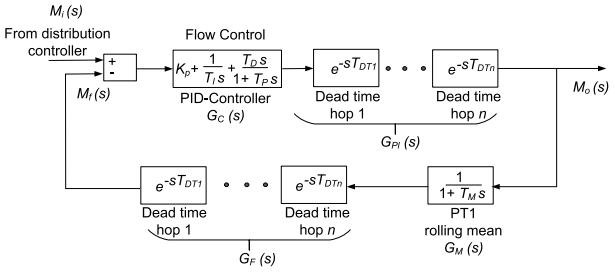


FIGURE 7. Network path flow control structure.

it contained a parasitic PT1 element which reflects the time constants of non-ideal controller elements.

$M_i(s)$, $M_o(s)$ and $M_f(s)$ are the Laplace transforms of the throughput $\mu(t)$. These are the setpoint or input throughput $\mu_i(t)$, the output throughput $\mu_o(t)$, and the feedback throughput $\mu_f(t)$. K_P is the proportional gain, T_I the integral time, and T_D the derivative time of the PID controller. The T_P in the PT1 element in the denominator of the derivative part of the PID controller represents the real PID behavior that contains parasitic filters. Therefore, the transfer function of the plant, that is, the network path in the frequency domain is given by:

$$G_{Pl}(s) = e^{-T_{DTP}s}, \tag{5}$$

where

$$T_{DTP} = \sum_{i=1}^{m} T_{DTi}, \tag{6}$$

and where $m \in \mathbb{N}$ is the number of hops from the controller to the link with the current maximum throughput along the path. T_{DTP} is the sum of the dead-times of these hops, consisting of the bridge latency and the LAN propagation delay. The transfer function of the closed loop is then

$$\begin{aligned} G_{CL}(s) &= \frac{M_o(s)}{M_i(s)} \\ &= \frac{G_c(s)G_{pl}(s)}{1 + G_c(s)G_{pl}(s)G_M(s)G_F(s)} \\ &\quad \frac{(K_p + \frac{1}{T_I s} + \frac{T_D s}{1+T_P s})e^{-T_{DTP}s}}{1 + (K_p + \frac{1}{T_I s} + \frac{T_D s}{1+T_P s})e^{-T_{DTP}s}\frac{1}{1+T_M s}e^{-T_{DTF}s}}, \end{aligned} \tag{7}$$

where the product $G_C(s)G_{Pl}(s)G_M(s)G_F(s)$ in the denominator is the transfer function $G_0(s)$ of the open loop. To assign the PID controller parameters, tuning according to Ziegler-Nichols [36] or Chien-Hrones-Reswick [37] was applied to a plant involving feedback and simulating the step response at the open loop at $M_f(s)$. One important goal of automation-data control is that no or only a minimum of data frames may be lost to avoid bumps in the controlled process. Therefore, an overshot of $M_o(s)$ over the reference level $M_i(s)$ must be avoided, because the operating point can be near the maximum bandwidth. An overshot would then mean congestion loss. As the plant consists of only dead-time elements, this limitation is equivalent to the requirement

for proportional gain: $K_P \leq 1$. Another reason for this limitation is the practical aspect; an overshot would mean an oscillation of load between two paths, which would only create unnecessary disturbances. The cost of this overshot avoidance is slower dynamic performance.

Generally, dead-time elements increase the difficulty of controlling the loop and promote its tendency toward instability. Because of the PT1 dampening effect of the rolling mean calculation in the feedback, the instability of the control loop can be counteracted if the sum of the dead-time elements is small compared with the T_{mean} of the rolling mean calculation. T_{mean} increases with the longest application cycle and is thereby determined by the slowest application, as stated in (1). The sum of dead-times depends on the selection of the traffic shaping-technology, the number of hops between the controller, and the location of the current throughput maximum. It further depends on the LAN propagation delays of the links between the hops. With certain traffic-shaping methods, bridge delays can be assumed to be nearly constant. Others imply variable bridge delays, and thereby variable dead-time elements in the control circuit. A nearly constant bridge delay and thereby a constant dead-time element, as given by, for example, EST traffic shaping, has the advantage that T_{DTP} does not need to be measured and transferred to the controller continuously. Instead, T_{DTP} can be calculated if the constant single dead-time per hop and the number of hops are known. If the dead-time needs to be measured, it is recommended to perform this continuously in parallel with the actual throughput control to obtain instant dead-time values for load control. Suitable methods for dead-time assignments are as follows:

1) Using the time synchronisation protocol [4] time information;
2) Measuring the ring round-trip delay, divided by the total number of hops, and multiplied by the distance of the maximum throughput in the number of hops;
3) To use a special frame to collect the accumulated latency to be updated and stored in the nodes.

The general control structure in Fig. 7 provides separate overall dead-times for path and feedback. The reason is that the dead-time elements on the plant and feedback paths are not always identical. The paths to be followed in the two directions to and from the relevant link are not necessarily the same. Furthermore, they could have different delays owing to the influence of the interfering traffic. The local maximum of throughput $\max_{i,j\in V} \mu_{ij}$ can be at different locations in the network domain at each distribution control loop sample time. This results in different path characteristics and, therefore, different controller parameters for flow control if optimal flow control with regard to dynamic performance and stability is to be achieved. Therefore, the controlling instance located within the AC must provide and use dedicated plant models for each possible location of $\max_{i,j\in V} \mu_{ij}$.

The dead-time element of one hop consists of the bridge transit delay or latency T_{BL} and the LAN propagation delay T_{LPD} from the bridge egress port to the next bridge ingress port. The actual transit delay through a bridge depends on several factors. TSN offers a variety of traffic shapers and schedulers for bridge internal MAC forwarding services as defined in IEEE 802.1Q [5]. Depending on the forwarding method used, the bridge internal forwarding delay, or, in the case of certain shapers or schedulers the overall path latency, is defined.

TABLE 1. Bridge to bridge delay components.

Delay type	Meaning/remark	Relevant?
Input queuing	not relevant here, as there are no input queues in the IEEE 802.1 bridge architecture that constitutes the basis for the bridges underlying this article.	no
Interference	depends on the number of nonring input ports and traffic ingress and is relevant for some of the investigated traffic shapers as queuing delay.	yes
Frame transmission	is the time it takes to transmit one frame at the transmit rate, which is assumed to be 1 $Gbit/s$ for the networks underlying this article	yes
LAN propagation	represents the time it takes to send the frame over the LAN to the next bridge depending on the media and distance.	yes
Store-and-forward	consists of all other bridge-internal forwarding elements assuming empty send queues.	yes
Output queuing	is caused by other frames waiting in the output queue to be sent before a frame is due to be sent.	yes

According to IEEE 802.1Q [5], Annex L.3, the worst-case latency for a frame for a single hop from bridge to bridge can be broken out into the components, as listed in Table 1, together with a statement of its relevance.

The pure **single-bridge latency**, without traffic which depends on the output queuing delay, can be calculated as the store-and-forward delay plus the transmission delay for a frame. The latter depends on frame size and link speed. Furthermore, no input queues are assumed because these are usually not common in standard switch ASIC designs. Bridge latency is calculated as

$$T_{BL} = T_{S\&F} + T_{Tr}, \tag{8}$$

where T_{BL} is bridge latency. $T_{S\&F}$ is the store-and-forward delay which is the time required to forward the frame in the bridge, and T_{Tr} is the transmission delay which is the time required to send the frame to the output port. Here, the bridge forwarding mode to the output port must be assumed to be the store-and-forward mode as the worst case. This is because more than one input port usually forwards data to the output port, and therefore, the faster cut-through mode is no alternative.

The **transmission delay** T_{Tr} is calculated as

$$T_{Tr} = MaxFrameSize[Byte]\frac{1}{B}8\ Bit, \tag{9}$$

where $MaxFrameSize$ is the maximum SDU size (Service Data Unit - net data load [5]) plus header (usually 42 bytes), B is the bandwidth (normally 1 Gbit/s for automation networks), and Bit counts the bits of a byte. The store-and-forward delay depends on bridge design. According to [10], a typical value can be assumed to be 700 - 800 ns.

The **LAN propagation delay** T_{LPD} represents the cable delay from the output port to the next input port. Automation networks are typically set up using a copper Ethernet CAT 6 cable with a specific delay of approximately 5 ns/m [38]. A 100 m Ethernet copper cable corresponds therefore to 0.5 μs cable delay. For precise LAN propagation delay assignment, the actual LAN propagation delay from an output port to the next input port can be retrieved from the clock synchronization peer-to-peer delay measurement [4].

The **output port queuing delay** T_Q is another element that can have a delaying influence during frame transfer through a bridge. Whether the queuing delay has an influence depends on the forwarding method, that is, the TSN traffic-shaping concept used.

VI. TSN SHAPER AND SCHEDULER DELAYS FOR HIGH PRIORITY CONTROL DATA

It is necessary to assign the actual dead-times introduced by various bridge's traffic-shaping and schudling technologies for CD. The focus of this study is to analyze the principal tendency of influence of the different shapers and scheduler in connection with different application types on the NLB. The most accurate determination of the delay of the different data traffic types in combination with all different shapers and schedulers is not important here. We concentrated on highest priority CD as the most important traffic type for automation applications and compare the influence by focusing on one fixed traffic type and one example of process data frame length and frames per network cycle and application cycle. The **SPQ** transmission selection for CD must assign the highest or second highest QoS priority to the CD to achieve privileged frame handling. This is necessary to achieve the minimum reliable bridge latency to guarantee the determinism necessary for control tasks. Assuming the highest priority for CD and no interfering traffic of the same highest traffic class (In-Class-Interference - ICI) from other controllers or interconnection links along the path, the worst-case situation would be if a maximum-sized frame of 1530 bytes [5] would already be in the send process in each hop before the CD frame could be forwarded. This frame could not be interrupted with pure SPQ handling capabilities and would delay the forwarding of the CD. To calculate the delay time per hop, (8) is expanded by the output port queuing delay T_Q for this disturbing frame, and is thus

$$T_{BL} = T_{S\&F} + T_{Tr} + T_Q. \tag{10}$$

Therefore, for the CD with the highest priority applying SPQ, the maximum output port queuing delay T_Q is identical to the transmission time of the longest frame transmission time T_{Tr}. If SPQ is combined with FP, the delay T_Q is reduced to the transmission time T_{Tr} of the minimum fragment size, typically 64 bytes [5]. If the CD is assigned only the second highest priority, it is only acceptable if the requirements for determinism are relaxed. In this case, the highest priority is only for sporadic network management traffic. For the evaluations in this study, the highest QoS priority of seven was assumed. The overall path dead-time with SPQ under the conditions stated above is determined by the number of hops to be traversed through the network, delay per hop, and the sum of the LAN propagation delays from the talker to the link of the maximum throughput. It is thus

$$T_{DTP\ SPQ} = n_{max\ \mu}\,(T_{S\&F} + T_{Tr} + T_Q) + \sum_{i=1}^{n_{max\ \mu}} T_{LPG\ i}, \tag{11}$$

where $T_{DTP\ SPQ)}$ is the sum of the dead-times of the SPQ path defined in (6) from the controller to the maximum throughput, $n_{max\ \mu}$ is the hop count from the controller to the maximum throughput, and $T_{LPG\ i}$ is the LAN propagation delay between the hops.

The **EST or TAS** [5] timing calculation is based on the assumption that with EST, the data can transition through the complete network within a defined gating window. This gating window is synchronized among all nodes in the network domain and reserved for one or more dedicated traffic classes. A network cycle can be divided into several gating windows for the different traffic classes. The remaining time of the network cycle which is not consumed by gating windows is usually left to the non-CD and best-effort (BE) data traffic with lower timing requirements. Therefore, with EST, the queuing delay is not relevant because it must be ensured that the cyclic data traffic fits into the gating window. Thus the necessary length of the gating window length depends on the overall data to be transported per link. This is caused by $\boldsymbol{Ta_{ij}}$ stream demand $\boldsymbol{S_{ij}}$ from each end station of $\boldsymbol{D_i}$ at each node $v_i \in \boldsymbol{V}$ along the path. Furthermore, it depends on the maximum number of hops of all possible paths, which is usually limited by the maximum network diameter and LAN propagation delays between the hops. The maximum data calculation can be achieved through network traffic pre-planning and/or dynamic limitations throgh SR. SR is achieved using either the MSRP [5] or RAP [14]. As an alternative to delay measurement, the path delay with EST can also be calculated. A good approximation is the distance of the throughput maximum relative to the complete ring length calculated in hop counts. Provided that the LAN propagation delay differences can be neglected, it is calculated as

$$T_{DTP\ EST} = \frac{T_{Tr}}{n_{max\ ring}} n_{max\ \mu}, \tag{12}$$

where $T_{DTP\ EST}$ is the overall EST path dead-time from the controller to the maximum throughput, T_{GW} is the length of the gating window, $n_{max\ ring}$ is the maximum hop count of the ring, and $n_{max\ \mu}$ is the hop count from the controller to maximum throughput. The crucial advantage of EST in terms of flow control is the possibility of reserving dedicated gating windows for different application data with different application cycles. Thus, the disadvantage of the slowest application cycle determining the control dynamics of faster applications can be avoided. If data are sent unsynchronized from the talker to the edge bridge, an additional worst-case

waiting time of one network cycle time for the next gating window to start must be added. If interfering traffic of the same traffic class from other controllers along the path can be excluded (no ICI), the overall dead-time along the path is reduced to pure bridge latencies plus LAN propagation delays without any queuing delay. It is thus:

$$T_{DTP\ EST} = n_{max\ \mu}\,(T_{S\&F} + T_{Tr}) + \sum_{i=1}^{n_{max\ \mu}} T_{LPG\ i}, \quad (13)$$

where $T_{DTP\ EST}$ is the sum of the dead-times of the EST path defined in (6) from the controller to the maximum throughput μ_{max}. If the maximum ICI is to be considered, the dead-time increases to the length of the gating window because this is derived from the maximum data transport demand of that traffic class.

The timing of the **CQF scheduler** [11] is determined by the number of hops to be traversed through the network, the length of the cycle time, and the LAN propagation delay from the talker to the link of the maximum throughput:

$$T_{DTP\ CQF} = T_{NC}\ n_{max\ \mu} + \sum_{i=1}^{n_{max\ \mu}} T_{LPG\ i}, \quad (14)$$

where $T_{DTP\ CQF}$ is the sum of the dead-times of the CQF path defined in (6) from the controller to the maximum throughput. T_{NC} is the length of the network cycle, $n_{max\ \mu}$ is the hop count from the controller to the maximum throughput, and $T_{LPG\ i}$ is the LAN propagation delay between the hops. The network cycle time with CQF can be selected to be smaller than that with EST because only one hop must be traversed instead of the complete network in the worst case.

ATS is the most complex shaper among the various TSN shapers. It offers a variety of configurations, which makes timing analysis complex. However, the special properties of CD reduce the permissible configuration combinations. First, CD must be transported with the highest priority of cyclical frames, sharing this only with the highest absolute priority of sporadic management frames. Therefore, ATS IPV must be selected as the highest priority. Second, a burst of CD must also be transported as a burst. This implies that it must not be stretched. The committed burst size parameter of the token bucket shaper must be sufficiently large to guarantee this. CD data must be assigned to a reserved stream gate. Unlike the EST, the bridges in the ATS domain are not synchronized. Therefore, unhampered data transport is not possible. In the best case, all gates in the bridges along a path are opened by accident at the same absolute point in time. This would result in a timing similar to the EST timing. In the worst case, all waiting times for gate opening when reaching the next hop are maximal. In this case, the waiting time per hop is equivalent to the network cycle time. This would result in a timing similar to that of CQF. Therefore, the worst-case overall path dead-time with ATS for high-priority CD without ICI is determined by:

1) the number of hops to be traversed through the network,
2) the store and forward delay (no token bucket delays for CD),
3) the transmission time and queuing time of one maximum frame,
4) one network cycle per hop, and
5) the sum of the LAN propagation delays from the Talker to the link of the maximum throughput

$$T_{DTP\ ATS} = n_{max\ \mu}\,(T_{S\&F} + T_{Tr} + T_Q + T_{NC}) + \sum_{i=1}^{n_{max\ \mu}} T_{LPG\ i}, \quad (15)$$

where $T_{DTP\ ATS}$ is the sum of the dead-times of the ATS path for the highest priority CD. If the maximum ICI must be considered, a single additional network cycle length is added.

If **Stream Reservation (SR)** is applied, it has an influence. SR requires time for the reservation process to be completed before a stream can flow. With decentralized reservation, this time consumption is caused by the reservation protocols of MSRP/MRP or RAP Talker Advertise and Listener Join frame transitions through the network. These typically follow the VLAN-controlled paths. With a central configuration, a delay is required for reservation via, for example, the simple network management protocol (SNMP) or network configuration protocol (NETCONF) from a central network controller (CNC). Two consequences are possible for load balancing.

1) **Pre-reservation:** All possible network path options for a stream to flow are reserved with 100 per cent of the stream bandwidth demand. However, only a fraction is used per path or a different path may be used completely, following the load control calculation result. This has the advantage of highly dynamic path change. The disadvantage is that bandwidth overbooking must be considered when using the full network capacity. Because distribution control is never ideal, admissible overbooking must be limited. A proposal for a possible limitation is

$$B_{MultiRes} = B_{SingleRes}(1 + \frac{n-1}{1+\eta J}), \quad (16)$$

where $B_{MultiRes}$ denotes the maximum admissible bandwidth reservation per path for multipath overbooking. $B_{SingleRes}$ is the maximum reservable bandwidth for a single-path network, and n is the number of available paths. Parameter J is the quality of the distribution control as introduced in Section IV. η is an empirical weighting factor that amplifies the influence of control quality. Pre-reservation is a compulsory precondition if streams are divided into several paths instead of completely shifting them between paths. Overbooking must be limited conservatively to ensure that the load deviations do not exceed 100 % load per path.
2) **Dynamic reservation:** The re-reservation process just before the shift of a stream completely from a previous path to a new path involves a new reservation

process for the new path just before the shift. This process implies an additional time span and a slower path change. In the case of a decentralised reservation, this time span consists of

$$T_{Res} = n\,(T_{TAdv} + T_{LJoin}), \tag{17}$$

where T_{Res} is the overall reservation time from the talker to the relevant listener, T_{TAdv} is the time a Talker Advertise needs to transition over one hop, T_{LJoin} is the time a Listener Join needs to transition over one hop, and n is the number of hops from the talker to the relevant listener. T_{Res} would appear as an additional dead-time element in the distribution control.

The sum of dead-times in a control circuit is synonymous with the difficulty in controlling it. A common way to identify the influence of dead-time is to use normalized dead-time. This is related to the time constants of the delaying elements, that is, the PT1 element in this case, such that [36]

$$\tau = \frac{T_{DT}}{T_{DT} + T_n}, \tag{18}$$

where τ is the normalized dead-time ($0 \leq \tau \leq 1$), T_{DT} is the real dead-time of the plant, and T_n is the delay time constant of the plant. If τ is near 1, usually $\geq \frac{2}{3}$ [36], a system, as a rule of thumb, is said to be dead-time-dominant. Otherwise, it is said to be lag-dominant. The PID controller of the control structure in Fig. 7 is sufficient if the dead-times are relatively small in comparison to the PT1 rolling mean element, that is, the system is rather lag-dominant. If the system is dead-time-dominant or demands enhanced dynamics, the PID controller should be replaced using a predictive controller, that is, either a Smith predictor [23], [39] or a model predictive controller [36]. In summary, it can be stated that different traffic shapers and schedulers introduce different dead-time elements into the flow control circuit, thus influencing the control characteristics. Furthermore, the network and application cycles in the network domain play a crucial role. They determine the rolling mean time constant of the throughput measurement and thus change the ratio between the dead-times and delay times. This ratio is, in turn, responsible for the maximally achievable control performance. If stream reservation is used, it is recommended to work with pre-reserved resource reservations to meet stringent dynamic requirements.

VII. NETWORK CONTROL SIMULATION AND RESULTS

One result of the data traffic analysis in Section VI is that the actual dead-time for networks with a similar number of hops and throughput depends on different traffic shaping or scheduling methods. To compare their influences, a sample network model, as shown in Fig. 8, was simulated using MATLAB/Simulink. The parameters for the path delay and feedback delay are calculated as follows and summarized in Table 2. The simulation parameters are listed in Table 3. A network of 25 hops from controller to the link with the current maximum throughput μ_{ijmax} was assumed, which is half

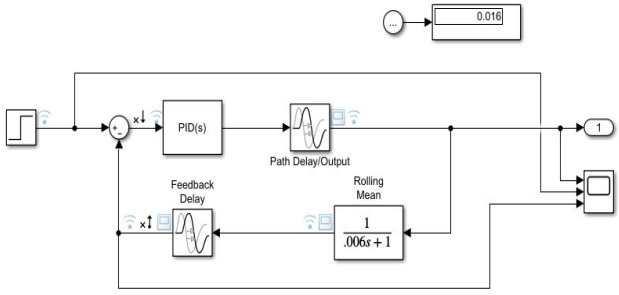


FIGURE 8. Network control simulation model for MATLAB.

TABLE 2. Path dead-times for the different traffic shapers and schedulers.

Traffic shaper and traffic type	Worst case path dead-time (μs)
SPQ without ICI	370
SPQ with Preemption and without ICI	90
SPQ with maximum ICI	370
EST without ICI	80
EST with maximum ICI	280
CQF without ICI	4890
CQF with maximum ICI	5090
ATS without ICI	5140
ATS with maximum ICI	5340

TABLE 3. Simulation parameters.

Traffic shaper and traffic type	Worst case path dead-time (μs)	Simul. time (ms)	PID K_p	PID T_I(ms)	PID T_D(ms)
EST without ICI	80	20	0.75	7.7	0
SPQ with maximum ICI	570	20	0.95	6.5	0
ATS with maximum ICI	5340	100	0.38	22	0

the typical maximum ring diameter of 50 hops [40]. The average cable length between the hops was assumed with 100 m Ethernet CAT6 cable which have a typical propagation delay of about 0.5 μs [38]. Thereby, $T_{LPD} = 24 \cdot 0,5\mu s = 12\mu s$, under the assumption that the controller is near the first bridge with insignificant LAN propagation delay. A maximum data amount of 100 streams with a maximum of 200 bytes of net SDU data load plus a 42 bytes Ethernet header was assumed. For a single frame, this leads to a transmission delay according to (9) of

$$T_{Tr} = 242Byte\frac{(8\frac{Bit}{Byte})10^{-9}s}{Bit}, \tag{19}$$

and thereby to a Bridge Latency time according to (8) of

$$T_{BL} = T_{S\&F} + T_{Tr} = 0.8\mu s + 1.936\mu s \approx 2.75\mu s. \tag{20}$$

For SPQ without considerable ICI, the dead-time needed to shift one frame from the talker to the link with the maximum

throughput is given by (11),

$$\begin{aligned} T_{DTP\ SPQ} &= n_{max\ \mu}\,(T_{S\&F} + T_{Tr} + T_Q) + \sum_{i=1}^{n_{max\ \mu}} T_{LPG\ i} \\ &= 24(0.8\mu s + 1.936\mu s \\ &\quad + 1530Byte\frac{(8\frac{Bit}{Byte})10^{-9}s}{Bit}) + 12\mu s \\ &= 371.42\mu s \\ &\approx 370\mu s. \end{aligned} \tag{21}$$

The path delay with FP, where only 64 bytes instead of 1530 bytes are to be calculated for T_Q, would result in 89.9 μs $\approx$ 90 μs path delay. These low path delay values for SPQ are a result of the assumption that no other interfering ICI enters the path which would raise T_Q. If the worst case is assumed for this example, the rest of the maximum load enters the ring at a ring interconnection to a coupled ring in between, and this data is in front of the control data, one further $T_{Q\ ICI}$ of

$$T_{Q\ ICI} = 99 \cdot 1.936\mu s \approx 200\mu s \tag{22}$$

would have to be added. This resulted in an SPQ with an ICI dead-time of approximately 570 μs.

For **EST with ICI**, calculation of the necessary gating window length is required. To shift the maximum data of 24200 bytes through the network along the path, one T_{BL} of 195 μs (as reception and forwarding of bytes from bridge to bridge occur nearly simultaneously) plus the complete LAN propagation delay of $T_{LPD} = 12\mu s$ is to be calculated. This resulted in a minimum gating window time T_{GW} of 207 μs. This time also represents the worst-case delay for the I-CD data if the talker transmits synchronized with the network gating window. For unsynchronized talkers for NI-CD, one network cycle of worst-case waiting time must be added, which would then result in a delay of 1207 μs, assuming a network cycle time of 1 ms. For EST without ICI, the delay would be according to (13),

$$\begin{aligned} T_{DTP\ EST} &= n_{max\ \mu}\,(T_{S\&F} + T_{Tr}) + \sum_{i=1}^{n_{max\ \mu}} T_{LPG\ i} \\ &= 25(0.8\mu s + 1.936\mu s) + 12\mu s \\ &= 80.4\mu s \\ &\approx 80\mu s. \end{aligned} \tag{23}$$

For **CQF**, one network cycle is required to transfer data over one hop. According to (20), this needs to be at least $T_{BL} \approx 195\mu s$ for all 100 streams of this example, assuming that this data is the only traffic class to be transported within the network cycle. The LAN propagation delay must be added to reach the next hop. The overall delay from controller to the link with the current maximum throughput $\mu_{ij\ max}$ is then, according to (14),

$$\begin{aligned} T_{DTP\ CQF} &= T_{NC}\ n_{max\ \mu} + \sum_{i=1}^{n_{max\ \mu}} T_{LPG\ i} \\ &= 25 \cdot 195\mu s + 12\mu s \\ &= 4887\mu s \\ &\approx 4890\mu s. \end{aligned} \tag{24}$$

As with SPQ, if the maximum ICI is considered, the dead-time must be increased by a further 200 μs.

For **ATS**, the same network cycle time as that of CQF was assumed, and that it was the only CD traffic class to be transported. According to (25), the worst-case path delay for the network path under simulation must be calculated as

$$\begin{aligned} T_{DTP\ ATS} &= n_{max\ \mu}\,(T_{S\&F} + T_{Tr} + T_Q + T_{NC}) \\ &\quad + \sum_{i=1}^{n_{max\ \mu}} T_{LPG\ i} \\ &= 25(0.8\mu s + 1.936\mu s \\ &\quad + 1530Byte\frac{(8\frac{Bit}{Byte})10^{-9}s}{Bit} + 195\mu s) + 12\mu s \\ &= 5141.4\mu s \\ &\approx 5140\mu s. \end{aligned} \tag{25}$$

As with SPQ and CQF, if the maximum ICI is considered, the dead-time would have to be increased by 200 μs.

Bandwidth reservation reconfiguration dead-times were not considered in the simulations, as dynamic changes in reservation have practically no relevance because they are too time consuming. Table 2 summarizes the path dead-time results for different traffic shapers and schedulers for the simulated network. As outlined in Section VI, the influence of these dead-times is dominant only in networks which are not informed by slow applications, forcing a long integration time for the rolling mean calculation. To visualize this influence, a high-performance application with an application cycle of only 2 ms was simulated. The integration time for the rolling mean calculation of the actual value feedback is selected to be five times the application cycle of 2 ms; that is, m of (4) is 5. This is equivalent to a time constant of approximately 6 ms for the PT1 time constant T_{mean}. The PID controller is optimised for minimum overshot rather than for fast setpoint approximation, for the reasons mentioned above. The dynamic behaviour was analyzed using the reference step response diagram. This shows the settling time T_S when the output has reached the input reference step value. The control circuit contains dead-times which introduce nonrational elements into the transfer function. Therefore, stability analysis via the poles and zeros of closed-loop or open-loop systems is not available. Instead, the Nyquist criteria for the open-loop provide evidence of the stability and robustness of closed-loop flow control. If the magnitude of the transfer function of the open loop $|G_0(s)| = |G_C(s)G_{Pl}(s)\ G_M(s)\ G_F(s)| < 1$ (compared to (7)), at $Im(G_0(s)) = 0$, the closed-loop is stable. This can be read directly from the Nyquist diagram as it is the x-value where the graph intersects with the x-axis. The gain factor at $Im(G_0(s)) = 0$ to reach $|G_0(s)| = 1$, that is, the gain margin g_M, should not be smaller than $2 \mathrel{\hat{=}} 6\ dB$ for a robust control design stability reserve. The second stability criterion is the phase margin φ_M, which represents the angle of $G_0(s)$

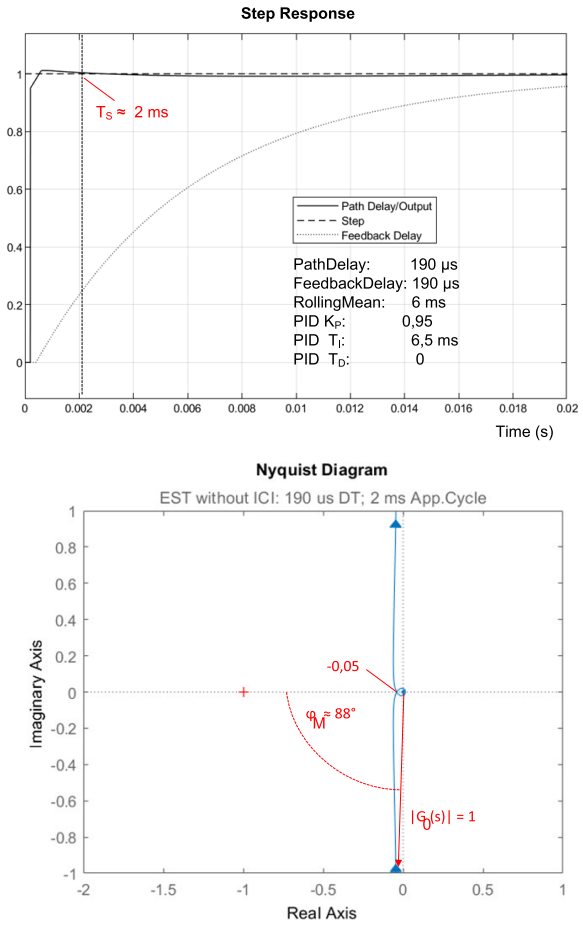


FIGURE 9. Step response and Nyquist diagram for EST.

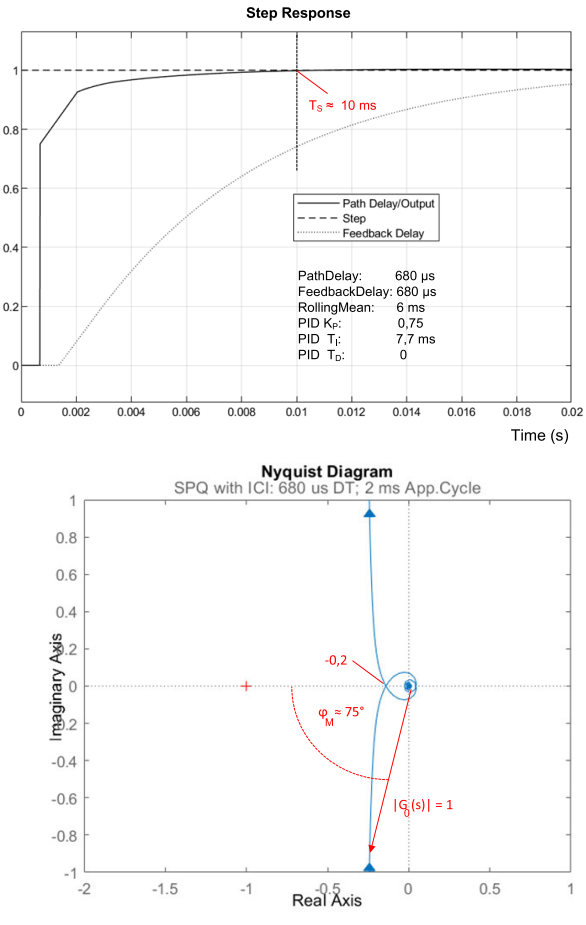


FIGURE 10. Step response and Nyquist diagram for SPQ with ICI.

with the negative real axis at the point of intersection with the unitary circle $|G_0(s)| = 1$. For robust control design, the phase margin φ_M should be $\geq 45°$. A Padé approximation of order 16 was applied to linearize the dead-time elements.

Fig. 9 to Fig. 11 show the simulation results for the step response and Nyquist diagrams for the selection of three networks and traffic situations featuring EST, SPQ, and ATS. The simulation parameters are listed in table 3.

Strictly speaking, the use of ATS in combination with a fast application cycle of 2 ms makes little sense from an application control point of view. This is because the data transport for the setpoint and actual value would be longer than the overall available time to calculate an application control algorithm, including the data transport times. Nevertheless, this was investigated here for reasons of flow control behaviour analysis.

Fig. 9 shows the step response and Nyquist diagram for EST without ICI. It represents the least possible dead-time (DT or T_{DT}) solution of 80 μs for both path dead-time and feedback dead-time and thereby the network with the least dead-time. According to (18), with

$$\tau = \frac{T_{DT}}{T_{DT} + T_n} = \frac{2 \cdot 80\mu s}{2 \cdot 80\mu s + 6000\mu s} = 0.03, \quad (26)$$

the control circuit was strongly lag-dominant. The control circuit features a fast settling time of $t_S = 2ms$, a gain margin of $1/0.05 = 20 \mathrel{\hat{=}} 26\ dB$, and a phase margin of about 88°, thereby representing a fast and robust control design.

Fig. 10 shows the step response and Nyquist diagram for SPQ with maximum ICI. It represents a control circuit with medium dead-time of 570 μs for both path dead-time and feedback dead-time. According to (18), with

$$\tau = \frac{T_{DT}}{T_{DT} + T_n} = \frac{2 \cdot 570\mu s}{2 \cdot 570\mu s + 6000\mu s} = 0.16, \quad (27)$$

the control circuit remains a lag-dominated network. It features a quite fast settling time of $t_S = 10ms$, a gain margin of $1/0,2 = 5 \mathrel{\hat{=}} 14\ dB$, and a phase margin of approximately 75°, representing a fast and robust control design. Fig. 11 shows the step response and Nyquist diagram for the ATS with maximum ICI. It represents the traffic shaper and traffic type with the worst dead-time of 5340 μs for the path dead-time and feedback dead-time. According to (18), with

$$\tau = \frac{T_{DT}}{T_{DT} + T_n} = \frac{2 \cdot 5340\mu s}{2 \cdot 5340\mu s + 6000\mu s} = 0.64, \quad (28)$$

the control circuit is at the border of being dead-time-dominant. The settling time has worsened to 70 ms, the gain

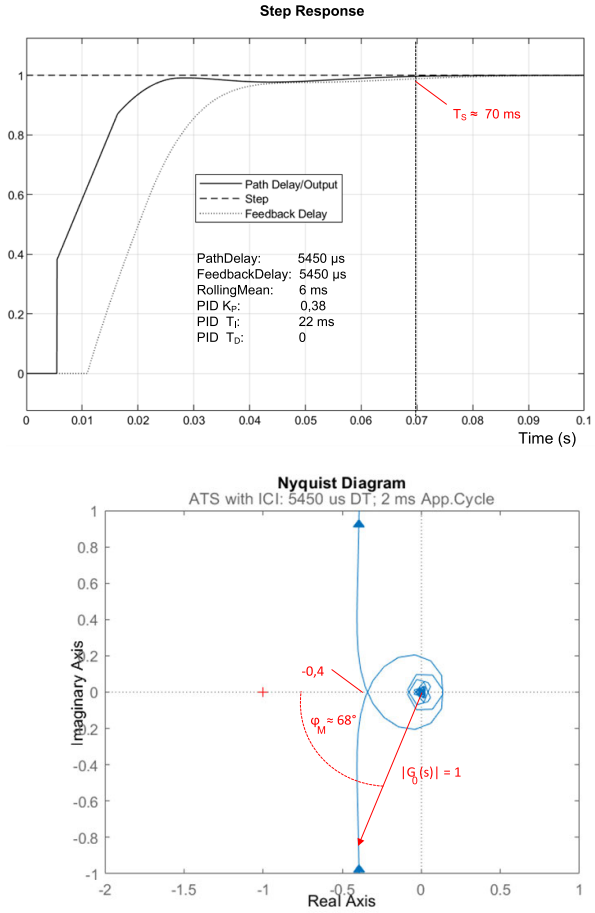


FIGURE 11. Step response and Nyquist diagram for ATS with maximum ICI.

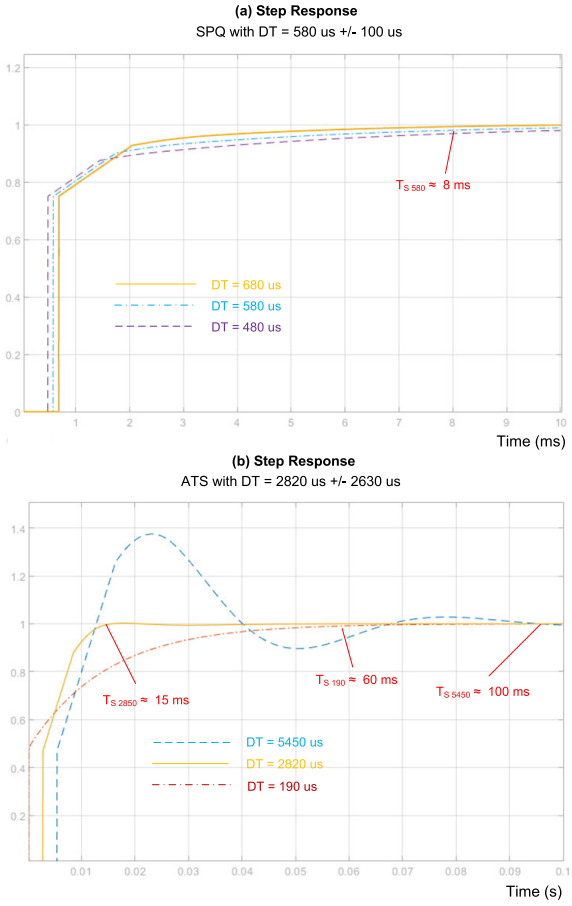


FIGURE 12. Dynamic performance deviation depending on dead-time uncertainties.

TABLE 4. Simulation results for shaper/scheduler examples for a fast 2 ms application cycle dominated network.

Traffic shaper and traffic type	Worst case path dead-time (μs)	Settl. time $t_S(ms)$	Gain margin $g_M(dB)$	Phase margin $\varphi_M()$	Ctrl robust-ness
EST without ICI	80	2	26	88	high
SPQ with max-imum ICI	570	10	14	75	medium
ATS with max-imum ICI	5340	70	8	68	low

margin to $1/0,4 = 2,5 \hat{=} 8\ dB$, and the phase margin to 68°, representing a control design at the border of robustness.

Fig. 9 to Fig. 11 clearly show the influence of the path dead-time and feedback dead-time. With increasing dead-time, the necessary control-loop settling time t_S grows approximately proportionally. At the same time, the intersection of the Nyquist diagrams with the negative real axis shifts with increasing dead-times toward -1, which is the critical point for stability. This results in lower gain margins and lower phase margins, and thereby less robust flow control circuits. The results are summarized in Table 4.

Because dead-time is either calculated or measured over an appropriate time span, the actual dead-time can differ. The possible uncertainty in the dead-time calculation or dead-time measurement makes a tuned flow-control circuit imprecise or unstable. To illustrate the effect of dead-time deviation, Fig. 12 (a) shows an example of the step response for the SPQ with 50 per cent ICI. A dead-time of 470 μs was assumed for both the path delay and feedback delay. The PID controller was optimized for a dead-time of $DT = 470\mu s$. In this case, the maximum deviation is represented by either no ICI or maximum ICI, leading to either $DT = 370\mu s$ dead-time or $DT = 570\mu s$ dead-time.

Fig. 12 (a) shows that the effect of the error is only a slightly mis-tuned control circuit. It provokes in this case a rather acceptable slower settling time of $t_S = 10ms$ for both 370 μs and 470 μs dead-time compared to 8 ms for the 570 μs tuned control circuit. Fig. 12 (b) shows the result if the same test case is applied to a tuned control circuit featuring ATS. An average medium dead-time of $DT = 2710\mu s$, a possible deviating minimum dead-time of $DT = 80\mu s$ and a maximum dead-time of $DT = 5340\mu s$ is assumed. The considerably deviating dead-times provoke substantial deviations in settling time of $t_S = 70ms$ for $T_{DT} = 80\mu s$, and $t_S = 95ms$ for $T_{DT} = 5340\mu s$, compared to the tuned settling

time of $t_S = 15ms$. In addition, the higher actual dead-time produces a considerable overshot of 40 per cent. Summarizing the influence of ICI, it can be stated, as Fig. 12 (a) shows, that ICI has very low influence on the control quality for SPQ. The small dead-time deviation tolerance band of only +/− 20 per cent has hardly any noticeable control performance consequences for the sample network. The tolerances for EST and CQF, as summarized in Table 2, were also uncritical. However, ATS, as illustrated in Fig. 12 (b), has a high uncertainty of approximately 90 per cent in this case. This was caused by the asynchronous gating times between the bridges, which resulted in bad settling times and a high overshot of the flow control circuit.

VIII. CONCLUSION

In this study, the influence of the selection of the TSN traffic shaper or scheduler on data flow control performance in automation networks was investigated. The shaper and scheduler latency timing models for the control data were compared in terms of their impact on data path delays. A thorough investigation was conducted on the dependency of the latency timing model on parameters such as network extension, network communication cycles, application cycles, amount of data per traffic class, and network communication cycle. Moreover, the impact of stream bandwidth reservation and application cycle times on data flow control was shown. The results show that EST, pure SPQ, and SPQ with FP are the best selections from a flow control point of view because of their low absolute dead-times. In addition, by increasing the amount of data traffic and number of hops, the CQF and ATS delay times increase more than those of the EST and SPQ. Therefore, CQF and ATS can still be good selections within smaller networks, lower-loaded networks, or both. Furthermore, it can be stated that the overall possible interfering traffic load extends the gating window sizes for EST, CQF, and ATS, and thereby, the resulting dead-times. This has a negative effect on the achievable control performance. A further finding is that the slowest application cycle within a traffic class of a network domain assigns the minimum integration time of the rolling mean calculation of the throughput feedback. A longer integration time is equivalent to a slower dynamic control performance. Therefore, along with slow applications, the higher dead-time shapers CQF and ATS can still be acceptable selections. This is true if the condition ''slowest application time > two times longest path dead-time'' is true. For fast applications with a 2 ms application cycle time or faster, only the low-dead-time shapers EST and SPQ are recommended. EST and SPQ are also preferred in conjunction with load control for extensive automation network rings. If shapers with a higher variance of dead-time, such as ATS, are to be used with data flow control, the control circuits should be optimized by assuming maximum dead-times to avoid overshoots during operation. Another unique advantage of EST, in addition to its low absolute dead-time and low dead-time uncertainty, is the possibility of separating data traffic in a timely manner. This allows the decoupling of the data transport and control of fast applications from those of slower applications. Thus, tailored flow control circuits can be implemented for different application groups, that is, fast dynamic control for fast applications, and slow control for slow applications. The actual distribution control in a ring topology, which is a prevalent network topology for redundant automation networks, will be evaluated in our future work. A further challenging task is to find a proper collaboration method for several controllers that apply load distribution to the same ring, thereby influencing each other.

REFERENCES

[1] *IEEE Standard for Local and Metropolitan Area Networks-Audio Video Bridging (AVB) Systems*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1BA, 2011.

[2] N. Finn, ''Introduction to time-sensitive networking,'' *IEEE Commun. Standards Mag.* vol. 2, no. 2, pp. 22–28, Jun. 2018.

[3] L. LoBello and W. Steiner, ''A perspective on IEEE time-sensitive networking for industrial communication and automation systems,'' *Proc. IEEE*, vol. 107, no. 6, pp. 1094–1120, Jun. 2019.

[4] *IEEE Standard for Local and Metropolitan Area Networks-Timing and Synchronization for Time-Sensitive Applications*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1AS, 2020.

[5] *IEEE Standard for Local and Metropolitan Area Networks-Bridges and Bridged Networks*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1Q, 2018.

[6] *IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks, Amendment: Forwarding and Queuing Enhancements for Time-Sensitive Streams*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1Qav, 2009.

[7] *IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks, Amendment: Frame Preemption*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1Qbu, 2015.

[8] *IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks, Amendment: Enhancements for Scheduled Traffic*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1Qbv, 2015.

[9] *IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks, Amendment: Path Control and Reservation*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1Qca, 2015.

[10] *IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks, Amendment: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1Qcc, 2018.

[11] *IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks, Amendment 29: Cyclic Queuing and Forwarding*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1Qch, 2019.

[12] *IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks, Amendment: Per-Stream Filtering and Policing*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1Qci, 2016.

[13] *IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks, Amendment: Asynchronous Traffic Shaping*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1Qcr, 2020.

[14] *IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks, Amendment: Resource Allocation Protocol*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1Qdd, 2022.

[15] H. Wang, H. Xie, L. Qiu, Y. R. Yang, and Y. Zhang, ''COPE: Traffic engineering in dynamic networks,'' in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, 2006, pp. 99–110.

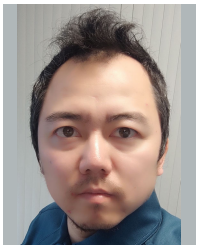
[16] A. Elwalid, C. Jin, S. Low, and I. Widjaja, ''MATE: Multipath adaptive traffic engineering,'' *Comput. Netw.*, vol. 40, pp. 695–709, Dec. 2002.

[17] I. Ahmad, S. N. Karunarathna, M. Ylianttila, and A. Gurtov, "Load balancing in software defined mobile networks," in *Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture*. Chichester, U.K.: Wiley, 2015, pp. 225–245.
[18] *TSN Profile for Industrial Automation, Use cases*. Accessed: Nov. 12, 2022. [Online]. Available: http://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf
[19] *TSN Profile for Industrial Automation*, document IEC/IEEE 60802, Accessed: Nov. 12, 2022. [Online]. Available: https://1.ieee802.org/tsn/iec-ieee-60802/
[20] D. Lopez-Perez, D. Laselva, E. Wallmeier, P. Purovesi, P. Lundén, and E. Virtej, "Long term evolution-wireless local area network aggregation flow control," *IEEE Access*, vol. 4, pp. 9860–9869, 2016.
[21] M. J. Neely, C. P. Li, and E. Modiano, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. Netw.*, vol. 16, pp. 396–409, 2008.
[22] S. Kandula, D. Katabi, B. S. Davie, and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 253–264, 2005.
[23] S. Mascolo, "Smith's principle for congestion control in high-speed data networks," *IEEE Trans. Autom. Control*, vol. 45, no. 2, pp. 358–364, 2000.
[24] M. Collotta, "FLBA: A fuzzy algorithm for load balancing in IEEE 802.11 networks," *J. Netw. Comput. Appl.*, vol. 53, pp. 183–192, Jul. 2015.
[25] D. Pompili and F. D. Priscoli, "A closed-loop fuzzy traffic controller for fair bandwidth sharing," *ACM SIGBED Rev.*, vol. 5, pp. 1–6, Jul. 2008.
[26] Z. Zhang and X. Zhang, "A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation," in *Proc. 2nd Int. Conf. Ind. Mechatronics Autom.*, May 2010, pp. 240–243.
[27] Y.-J. Chen, L.-C. Wang, M.-C. Chen, P.-M. Huang, and P.-J. Chung, "SDN-enabled traffic-aware load balancing for M2M networks," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1797–1806, Jun. 2018.
[28] N. G. Nayak, F. Dürr, and K. Rothermel, "Routing algorithms for IEEE802.1Qbv networks," *ACM SIGBED Rev.*, vol. 15, no. 3, pp. 13–18, Jun. 2018.
[29] M. A. Ojewale and P. M. Yomsi, "Routing heuristics for load-balanced transmission in TSN-based networks," *ACM SIGBED Rev.*, vol. 16, no. 4, pp. 20–25, Jan. 2020.
[30] F. A. R. Arif and T. S. Atia, "Load balancing routing in time-sensitive networks," in *Proc. 3rd Int. Sci.-Practical Conf. Problems Infocommunications Sci. Technol. (PIC ST)*, Oct. 2016, pp. 207–208.
[31] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. Elbakoury, "Performance comparison of IEEE 802.1 TSN time aware shaper (TAS) and asynchronous traffic shaper (ATS)," *IEEE Access*, vol. 7, pp. 44165–44181, 2019.
[32] *IEEE Standard for Local and Metropolitan Area Networks-Frame Replication and Elimination for Reliability*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.1CB, 2017.
[33] Z. Zhou, Y. Yan, M. Berger, and S. Ruepp, "Analysis and modeling of asynchronous traffic shaping in time sensitive networks," in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Jun. 2018, pp. 1–4.
[34] L. Zhao, P. Pop, and S. Steinhorst, "Quantitative performance comparison of various traffic shapers in time-sensitive networking," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 3, pp. 2899–2928, Sep. 2022.
[35] *IEEE Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, Standard 802.3br-2016, 2016.
[36] J. E. Normey-Rico and E. F. Camacho, "Control of dead-time processes," in *Advanced Textbooks in Control and Signal Processing*. London, U.K.: Springer, 2007.
[37] K. L. J. A. Chien Hrons and J. B. Reswick, "On the automatic control of generalized passive systems," *Trans. Amer. Soc. Mech. Eng.*, vol. 74, pp. 175–185, Feb. 1972.
[38] *Commercial Building Telecommunications Cabling Standard*, American National Standards Institute, Washington, DC, USA, Standard Ansi/TIA-568.1-D, 2015.
[39] L. De Cicco, S. Mascolo, and S.-I. Niculescu, "Robust stability analysis of Smith predictor-based congestion control algorithms for computer networks," *Automatica*, vol. 47, no. 8, pp. 1685–1692, Aug. 2011.
[40] *Industrial Communication Networks—High Availability Automation Networks—Part 2: Media Redundancy Protocol (MRP)*, International Electrotechnical Commission, document IEC 62439-2, 2021.

THOMAS WEICHLEIN (Member, IEEE) received the Diplom-Ingenieur (FH) degree in electrical engineering from the Georg-Simon-Ohm University of Applied Sciences, Nürnberg, Germany, in 1990. He is currently pursuing the Ph.D. degree with the University of Gloucestershire, U.K. Since 1990, he has been working in various positions within research and development for industrial automation systems at the companies Siemens and Danfoss. He is also holding the position of a System Architect for industrial automation and communication systems at Siemens Digital Industries, Nürnberg. His research interests include network media redundancy, time synchronization, and time sensitive networks and their cooperation with 5G and DetNet networks. In addition, he is involved in the definition of stream reservation protocols and network user interfaces.

SHUJUN ZHANG is currently a Professor in applied computing and technology at the School of Computing and Engineering, University of Gloucestershire, U.K. He has worked at the University of Warwick more than ten years, Salford University for three years, and Jilin University for six years. He has published about 150 papers and undertaken more than 40 funded projects. His research interests include the areas of applied computing and bionics engineering and their applications to both engineering and business management, covering the various smart system design and development, intelligent algorithms, big data and data mining, engineering system modeling, analysis, simulation optimization and control, physiological signal capturing and analysis (such as cell pulsation wave forms, frequency and strength, ECG, EMG, and EEG), and bio-inspired physiological signal-based algorithms and systems for both healthy and cybersecurity applications.

PENGZHI LI (Member, IEEE) received the Ph.D. degree from the University of Chinese Academy of Sciences, in 2019. He is currently a Lecturer in industrial systems, mechatronics and robotics at the University of Gloucestershire, U.K. Before joining the University of Gloucestershire, he was a Postdoctoral Research Associate with the Robotics for Extreme Environments Laboratory, The University of Manchester. Before that, he had worked with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, as a Researcher, from 2011 to 2019. His research interests include precision mechatronics, robotic manipulation and machine learning, autonomous mobile robots, and advanced control systems.



XU ZHANG received the M.Sc. degree (Hons.) from the University of Warwick and the Ph.D. degree from the University of Gloucestershire. He is currently working as a Senior Research Fellow with the Rolls-Royce UTC, University of Southampton. He is also carrying out the multidisciplinary research to provide solutions to aero-engine optimization challenges for Rolls-Royce. His research interests include artificial intelligence, image/video processing, algorithm in computer science and model parameterization, design optimization, and computational fluid dynamics in applied engineering.