

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

Thomas Weichlein<sup>a,\*</sup>, Shujun Zhang<sup>a</sup>, Pengzhi Li<sup>a</sup>, Xu Zhang<sup>b</sup>

<sup>a</sup> *University of Gloucestershire, United Kingdom*

<sup>b</sup> *University of Southampton, United Kingdom*

## 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, the data flow control for network load balancing in an 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 performed. The results show that enhancements for scheduled traffic (EST), strict priority queuing (SPQ), and the combination of SPQ with frame preemption 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 only alternatives for data load control in small networks or in conjunction with slow applications.

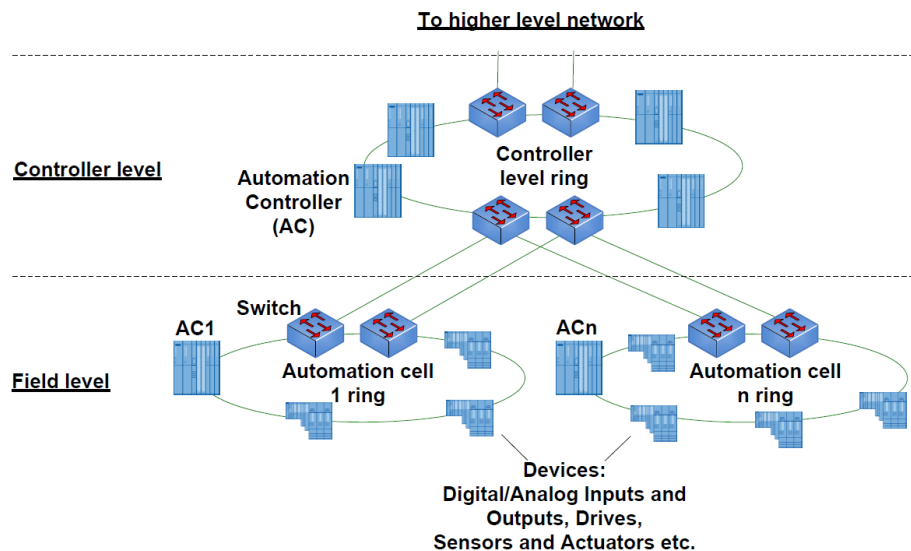
## Keywords

Time Sensitive Networks, Automation networks, Load Balancing, Data flow control.

## 1. 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 to the creation of a network standard framework called the “Time sensitive networks (TSN)” [1]. TSN is defined by the associated IEEE standards, extending the IEEE 802.1Q standard [2-12] which is still under further development. It defines various new functionalities and different traffic shapers and schedulers, such as the credit-based shaper (CBS), enhancements for scheduled traffic (EST), cyclic queuing and forwarding (CQF), asynchronous traffic shaper (ATS), classical strict priority queuing (SPQ), and frame preemption, to achieve highly efficient and deterministic data transport. The TSN project also allows for the use of multiple communication paths, primarily to provide redundancy. Classical non-TSN networks for Internet or campus communications, both wired and wireless, are typically set up as multipath 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 based on Open Systems Interconnection (OSI) layer 3 routing technology and dedicated load balancing methods, such as ISP networks [13], campus networks [14], and access networks of mobile connectivity [15]. In factory automation applications, networks with smaller spatial extensions 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 date, communication connections 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.

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 [16].



**Fig. 1. Typical automation communication network setup**

Controller-level rings usually contain a variety of higher-level ACs such as programmable logic controllers (AC) 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.

Comparing legacy layer 3 networks, three major load distribution strategies, also known as traffic engineering concepts, are visible [13]:

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 minimise the local load maxima.

Dynamic load control, as has been investigated in previous research on layer 3 ISP networks or campus networks [13, 15, 17, 18], usually differentiates between

- Flow control: The algorithm for controlling the data flow on a single path increases or decreases 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 automation networks control data (CD), as explained in Chapter 3. Regarding distribution and flow control, more research has been conducted on distribution control [13, 15, 17, 18] 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 tasks hosted by ACs. These application cycle times, sending data at each cycle, limit attainable control performance. 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 characterise 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 analysed. Furthermore, the impact of bandwidth reservation was investigated, and recommendations for load measurements are provided.

## 2. 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 and campus networks include common-case optimisation with penalty envelope (COPE) [13], multiprotocol label switching (MPLS) adaptive traffic engineering (MATE) [14], and traffic engineering explicit congestion protocol (TeXCP) [19]. Various controller types such as linear [13, 14, 19, 20], predictor-based [20], fuzzy [21, 22], ant colony algorithm [23], and stochastic control [18] have been deployed. Elwalid *et al.* [14] 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 using an extended gradient projection algorithm, averaging several load measurements over a period of time to compensate for asynchronism. Kandula *et al.* [19] 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.* [24] applied dynamic load balancing via software-defined networks (SDN) to machine-to-machine (M2M) networks. They work with a 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.* [17]. 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  $\delta = \frac{B}{R}$ , where  $\delta$  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 does not serve to control stability considerations.

However, research projects on distribution or flow control in the context of **TSN** are exceptions. Nayak *et al.* [25] investigated scheduling and routing possibilities using an IEEE 802.1Qbv EST traffic shaper with a central network controller. Ojewale and Yomsi [26] 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 [27] 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 [28] 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. Although this work does not directly cover load balancing, it is still of interest for this research because it provides detailed insights into the performance and behaviour of EST and ATS.

### 3. TSN automation networks

The communication data for automation 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 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 ranging 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 or twice on two disjoint paths because of seamless media redundancy, according to [29]. In the case of an automation ring topology, the two paths are in 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, although 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 with low bandwidth consumption, such as sensor data, are typically unsuitable for load control because of their low influence. It also contributes to the basic load of non-load-controllable data.

The TSN standards ensemble defines a variety of traffic shapers and schedulers to achieve optimal data transfer for CD which are briefly introduced here.

**Strict priority queuing or static priority queuing (SPQ)** assigns eight different quality of service (QoS) properties to various data classes. It is known from classical Ethernet switch ASICs defined in [3] and is used in general layer 2 networks and automation networks. It is also used in TSN. One example is the “PROFINET (PN) over TSN” [30, 31] data-forwarding system for 1 Gbit/s and 2.5 Gbit/s. PN over TSN combines SPQ with frame preemption for CD and non-CD. The SPQ provides one egress queue for each or 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 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 frame preemption 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 802.1Qcc [8] mainly for the purpose of transferring audio/video data without bursts and congestion. 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 because it is 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.

**Enhancements for scheduled traffic (EST)**, defined by IEEE 802.1Qbv [6], are also known as time-aware shapers (TAS) and assign gating windows to traffic classes. Each traffic class sending queue is then emptied at a defined time slot, 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 [9], 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 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)** [28, 32], defined by IEEE 802.1Qcr [11], 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 [5], 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 [33] for interspersing express traffic (IET) for the physical layer. Preemption can basically be applied in SPQ environments, but can, in principle, be also combined with EST, CQF, and ATS shapers.

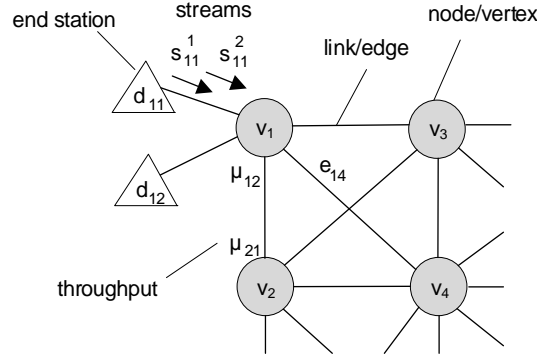
**Stream reservation (SR)** is another crucial feature offered by 802.1Q which can be used in the TSN domain. It is defined by the multiple reservation protocol (MRP)/multiple stream reservation protocol (MSRP) [3] and the currently emerging resource allocation protocol (RAP) [12]. SR in combination 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 their reserved bandwidth, an ingress limiter, as defined by IEEE 802.1Qci [10] can be deployed as supplementary protection.

Automation applications typically demand a variety of different application communication cycles, in which control loops and other automation tasks 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 has its own requirements for the communication speed with peripheral devices or other controllers. For instance, a slow temperature controller might exchange the setpoint and actual value with an analogue I/O card every 500 ms. On the other hand, a fast speed controller might need to exchange the setpoint and actual value with a drive in cycles of only a few  $10^{\text{th}}$  of  $\mu\text{s}$ . 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.

The majority of TSN networks, except for ATS or SPQ types, provide a network cycle which uses the timing information of bridges to synchronize data transport through the network domain. Thereby, a minimum of overall path latency is achieved. With synchronized bridges, the determining parameter for the overall path latency is network cycle time. It is determined by the shortest application cycle, and may not be longer than this. 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 on the dynamic performance and stability of the control circuit.

Fig. 2 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  $E(G)$ . Set  $V(G)$  to represent node  $v_i$  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 order  $n$ . The number of edges connected to a node determines its degree  $deg(v)$ .

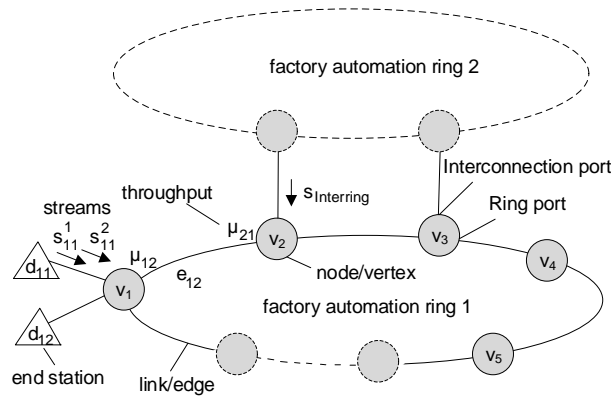


**Fig. 2. Abstracted TSN automation network**

Let  $\mathbf{D}_i = \{d_{i1}, \dots, d_{ij}\}$  be a set of devices (end stations) connected to the node  $v_i \in \mathbf{V} = \{v_1, \dots, v_n\}$ . Let furthermore be  $\mathbf{Ta}_{ij} = \{ta_{ij}^1, \dots, ta_{ij}^k\}$  a set of Talkers within  $d_{ij}$  and let  $\mathbf{Li}_{ij} = \{li_{ij}^1, \dots, li_{ij}^p\}$  be a set of Listeners within  $d_{ij}$ .  $\mathbf{Ta}_{ij}$  create a set of streams  $\mathbf{S}_{ij} = \{s_{ij}^1, \dots, s_{ij}^q\}$  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  $\mu_{ij}$  at the output port of node  $v_i$ . Each link  $e_{ij}$  provides two scalars of throughputs  $\mu_{ij}$  and  $\mu_{ji}$  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  $\mu$  of all links in the network can be formed as an instance  $\mathbf{M}$  of a distance matrix of graph  $G$ :

$$\mathbf{M} = \begin{bmatrix} \mu_{11} & \dots & \mu_{1n} \\ \vdots & \ddots & \vdots \\ \mu_{n1} & \dots & \mu_{nn} \end{bmatrix} \quad (1)$$

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. 3.



**Fig. 3. Automation ring graph**

The throughput distance matrix  $\mathbf{M}$  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,  $\mathbf{M}$  for a ring of five nodes results in

$$\mathbf{M}_{ring} = \begin{bmatrix} 0 & \mu_{12} & 0 & 0 & 0 \\ \mu_{21} & 0 & \mu_{23} & 0 & 0 \\ 0 & \mu_{32} & 0 & \mu_{34} & 0 \\ 0 & 0 & \mu_{43} & 0 & \mu_{45} \\ 0 & 0 & 0 & \mu_{54} & 0 \end{bmatrix} \quad (2)$$

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 the paths of the ring results from the optimization task to minimize the maximum throughput on the single links on the available paths:

$$\min \max_{i,j \in V} \mu_{ij} \quad (3)$$

Subject to:

$$\forall e \in E(G)$$

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 a 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.

#### 4. TSN automation network analysis

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 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. Furthermore, 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, automation data traffic is partly planned and well defined 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  $ta_{ij}$  which create stream sets  $s_{ij}$ , 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  $\mu$ s 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 timespan. 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.

$$T_{Mean} \geq m (\max_i T_{App\ i}) \quad (4)$$

where  $T_{Mean}$  is the recommended integration time for the calculation of the mean link load or throughput, and  $m$  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.  $T_{App\ i}$  is the application cycle of all the applications in the network domain.

Fig. 4 depicts the control structure of a single network path within the plant. It consists of all network hop latency times and LAN propagation delays that appear as dead-time elements. The feedback path contains a PT1 element caused by the rolling mean calculation of the feedback. It further contains 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. 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, it contained a parasitic PT1 element.

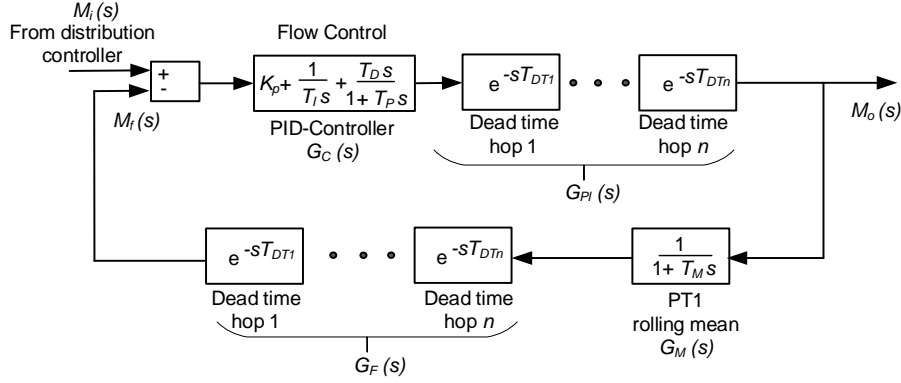


Fig. 4. Network path flow control structure

$M_i(s)$ ,  $M_o(s)$  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  $G_{Pl}(s)$ , that is, the network path in the frequency domain, is given by:

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

With:

$$T_{DTP} = \sum_{i=1}^m T_{DTi} \quad (6)$$

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  $G_{CL}$  is then

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

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 [34] or Chien-Hrones-Reswick [35] 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 overshoot 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 overshoot 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 overshoot would mean an oscillation of the load between two paths, which would only create unnecessary disturbances. The cost of this overshoot 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 protocols [2] time information;
2. Measuring the 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. 4 provides separate overall dead times for path and feedback. This is because 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 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 [3]. 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.

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

**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

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

$$T_{BL} = T_{S\&F} + T_{Tr} \quad (8)$$

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:

$$T_{Tr} = \text{MaxFrameSize [Byte]} \frac{1}{B} 8 \text{ Bit} \quad (9)$$

where  $\text{MaxFrameSize}$  is the maximum SDU size (Service Data Unit - net data load [3]) plus header (usually 42 Bytes),  $B$  is the bandwidth (normally 1 Gbit/s for automation networks), and  $\text{Bit}$  counts the bits of a byte. The store-and-forward delay depends on bridge design. According to [8] 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 [36]. A 100 m Ethernet copper cable corresponds therefore to 0.5  $\mu\text{s}$  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 synchronisation peer-to-peer delay measurement [2].

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.

To evaluate the actual dead times introduced by bridges with various traffic-shaping technologies for CD, a dedicated analysis is necessary.

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 [3] 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

$$T_{BL} = T_{S\&F} + T_{Tr} + T_Q \quad (10)$$

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 [3]. 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 used 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:

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

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** [3] 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 gating window length depends on the overall data to be transported per link. This is caused by  $Ta_{ij}$  stream demand  $S_{ij}$  from each end station of  $D_i$  at each node  $v_i \in V$  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 through SR. SR is achieved using either the MSRP **Fehler! Verweisquelle konnte nicht gefunden werden.** or RAP [12]. 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 follows:

$$T_{DTP\ EST} = \frac{T_{GW}}{n_{max\ ring}} n_{max\ \mu} \quad (12)$$

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 unsynchronised 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:

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

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  $\mu_{max}$ . 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** [9] 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 (i) the number of hops to be traversed through the network, (ii) the store and forward delay (no token bucket delays

for CD), (iii) the transmission time and queuing time of one maximum frame, (iv) one network cycle per hop, and (v) 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 is must be considered, a single additional network cycle length is added.

If **stream reservation (SR)** is applied, it has an influence. Stream reservation requires time for the reservation process to be completed before a stream can flow. With decentralised 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 full network capacity. Because distribution control is never ideal, admissible overbooking must be limited. A proposal for a possible limitation is

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

where  $B_{MultiRes}$  denotes the maximum admissible bandwidth reservation per path for multipath overbooking.  $B_{SingleRes}$  is the maximally reservable bandwidth for a single-path network and  $n$  is the number of available paths. Parameter  $J$  is the quality of the distribution control. 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.  $\eta$  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% of the 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

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

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 [34]:

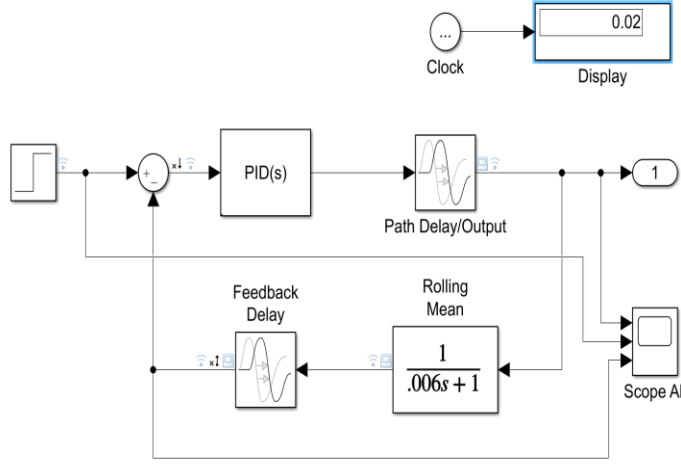
$$\tau = \frac{T_{DT}}{T_{DT} + T_n} \quad (18)$$

where  $\tau$  is the normalised 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  $\tau$  is near 1, usually  $\geq \frac{2}{3}$  [34], 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. 4 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 [20, 37] or a model predictive controller [34]. 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.

## 5. Network control simulation and results

One result of the data traffic analysis in Section 4 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. 5, was simulated using MATLAB/Simulink.



**Fig. 5. Network control simulation model**

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 the controller to the link with the current maximum throughput  $\mu_{ij \max}$  was assumed, which is half the typical maximum ring diameter of 50 hops [30, 38]. The average cable length between the hops was assumed with 100 m Ethernet CAT6 cable which have a typical propagation delay of about  $0,5 \mu s$  [36]. Thereby,  $T_{LPD} = 24 \times 0,5 \mu s = 12 \mu s$ , 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 Byte Ethernet header was assumed. For a single frame, this leads to a transmission delay according to (9):

$$T_{Tr} = 242 \text{ Byte} \frac{\left(8 \frac{\text{Bit}}{\text{Byte}}\right) 10^{-9} s}{\text{Bit}} = 1.936 \mu s$$

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

$$T_{BL} = T_{S\&F} + T_{Tr} = 0.800 \mu s + 1.936 \mu s \approx 2.75 \mu s \quad (19)$$

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):

$$\begin{aligned} T_{DTP \text{ SPQ}} &= n_{\max \mu} (T_{S\&F} + T_{Tr} + T_Q) + \sum_{i=1}^{n_{\max \mu}} T_{LPG \ i} \\ &= 24 \left( 0.8 \mu s + 1.936 \mu s + 1530 \text{ Byte} \frac{\left(8 \frac{\text{Bit}}{\text{Byte}}\right) 10^{-9} s}{\text{Bit}} \right) \\ &\quad + 12 \mu s \\ &= 371.42 \mu s \\ &\approx 370 \mu s \end{aligned} \quad (20)$$

The path delay with Frame Preemption, where only 64 Byte instead of 1530 Byte are to be calculated for  $T_Q$ , would result in  $89,9 \mu s \approx 90 \mu s$  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:

$$T_{Q ICI} = 99 \times 1.936 \mu s \approx 200 \mu s \quad (21)$$

would have to be added. This resulted in an SPQ with an ICI dead-time of approximately 570  $\mu s$ .

For **EST** with in-class interference (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  $\mu s$  (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  $\mu s$ . This time also represents the worst-case delay for the I-CD data if the talker transmits synchronized with the network gating window. For unsynchronised talkers for NI-CD, one network cycle of worst-case waiting time must be added, which would then result in a delay of 1207  $\mu s$ , assuming a network cycle time of 1 ms. For EST without ICI, the delay would be according to (13):

$$\begin{aligned} T_{DTP EST} &= n_{\max \mu} (T_{S\&F} + T_{Tr}) + \sum_{i=1}^{n_{\max \mu}} T_{LPG i} \\ &= 25(0.8 \mu s + 1.936 \mu s) + 12 \mu s \\ &= 80.4 \mu s \\ &\approx 80 \mu s \end{aligned} \quad (22)$$

For **CQF**, one network cycle is required to transfer data over one hop. According to (19) 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 the controller to the link with the current maximum throughput  $\mu_{ij \max}$  is then, according to (14):

$$\begin{aligned} T_{DTP CQF} &= n_{\max \mu} T_{NC} + \sum_{i=1}^m T_{LPG i} \\ &= 25 \times 195 \mu s + 12 \mu s \\ &= 4,887 \mu s \\ &\approx 4,890 \mu s \end{aligned} \quad (23)$$

As with SPQ, if the maximum ICI is considered, the dead time would have to be increased by a further 200  $\mu s$ . 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 (15), the worst-case path delay for the network path under simulation must be calculated as follows:

$$\begin{aligned} T_{DTP ATS} &= n_{\max \mu} (T_{S\&F} + T_{Tr} + T_Q + T_{NC}) + \sum_{i=1}^{n_{\max \mu}} T_{LPG i} \\ &= 25 \left( 0.8 \mu s + 1.936 \mu s + 1530 \text{ Byte} \frac{\left(8 \frac{\text{Bit}}{\text{Byte}}\right) 10^{-9} s}{\text{Bit}} + 195 \right) \\ &\quad + 12 \mu s \\ &= 5,141.4 \mu s \\ &\approx 5,140 \mu s \end{aligned} \quad (24)$$

As with SPQ and CQF, if the maximum ICI is considered, the dead time must be increased by 200  $\mu s$ .

**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 for the simulated network.

**Table 2. Path dead times for the different traffic shapers and schedulers**

Traffic shaper and traffic type	Worst case path dead time ( $\mu\text{s}$ )
SPQ without ICI	370
SPQ with Preemption and without ICI	90
SPQ with maximum ICI	570
EST without ICI	80
EST with maximum ICI	280
CQF without ICI	4,890
CQF with maximum ICI	5,090
ATS without ICI	5,140
ATS with maximum ICI	5,340

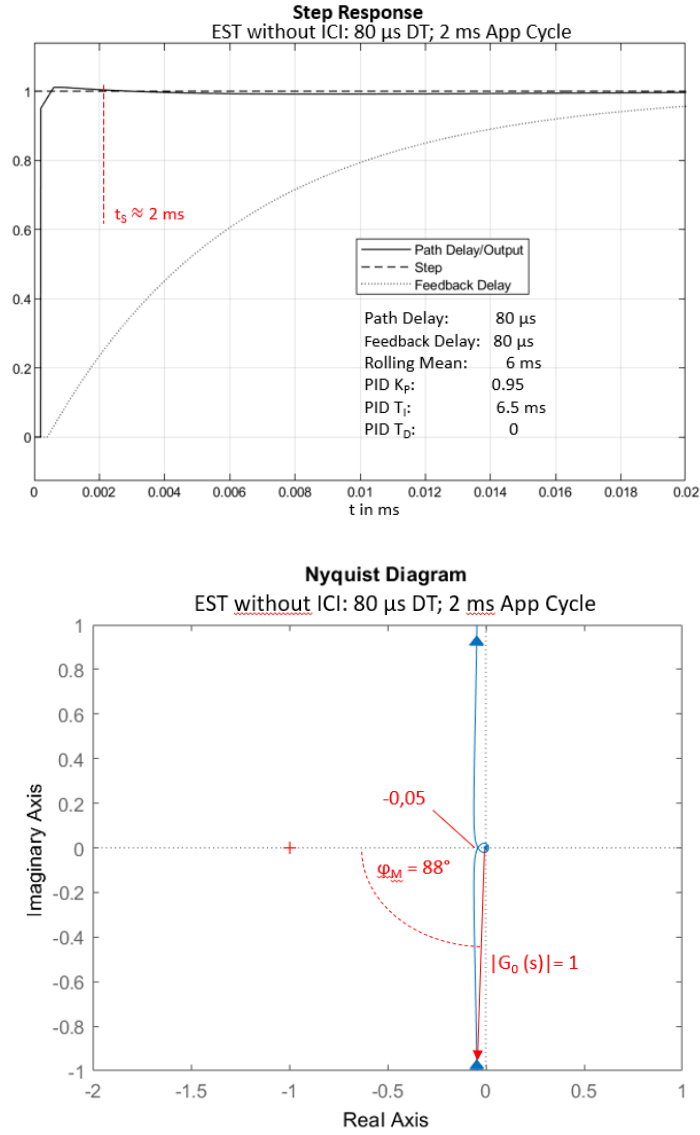
As outlined in Section 4, the influence of these dead-times is dominant only in networks which are not informed by slow applications, forcing a long integration time for the rolling mean calculation.

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 overshoot rather than for fast setpoint approximation, for the reasons mentioned above. The dynamic behaviour was analyzed using the reference step response. 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  $\text{Im}(G_0(s)) = 0$ , the closed loop is stable. The gain factor at  $\text{Im}(G_0(s)) = 0$  to reach  $|G_0(s)| = 1$ , that is, the gain margin  $g_M$ , should not be smaller than  $2 \triangleq 6 \text{ dB}$  for a robust control design stability reserve. The second stability criterion is the phase margin, which represents the angle of  $G_0(s)$  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  $\varphi_M$  should be  $\geq 45^\circ$ . A Padé approximation of order 16 was applied to linearize the dead time elements. Fig. 6 to Figure 8 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.

**Table 3. Simulation parameters**

Traffic shaper and traffic type	Worst case path dead time ( $\mu\text{s}$ )	Simulation 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

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 is investigated here for reasons of flow control behaviour analysis.



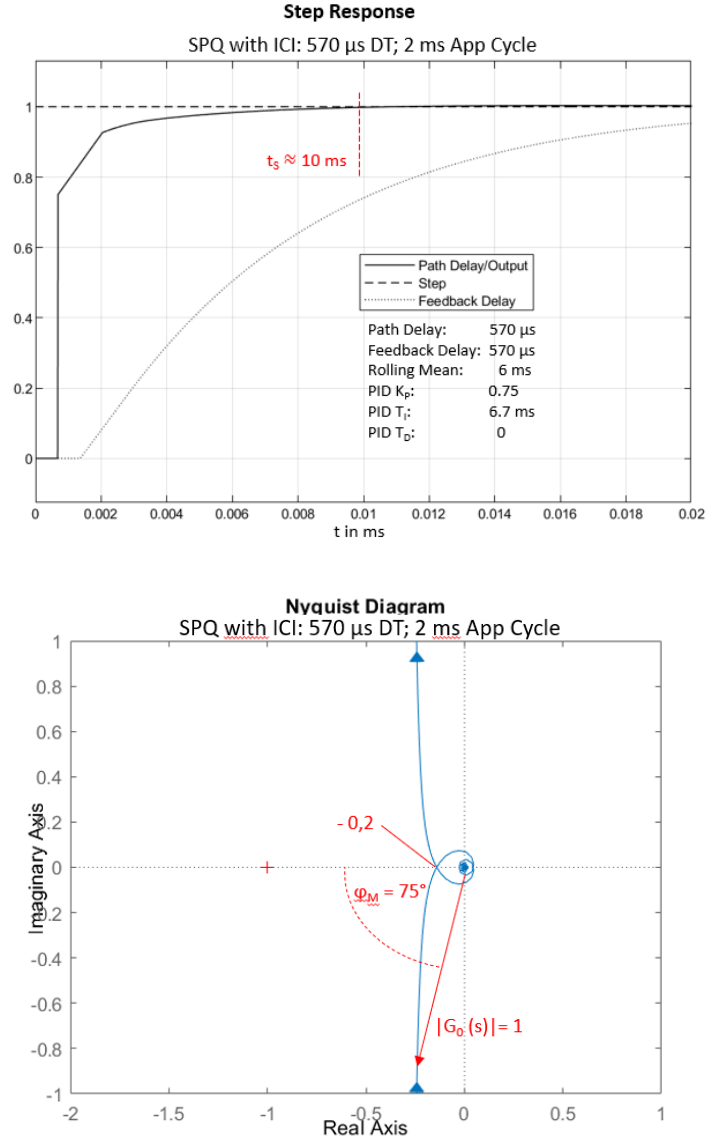
**Fig. 6. Step response and Nyquist diagram for EST**

Fig. 6 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  $\mu$ s for both path dead time and feedback dead time and thereby the network with the least dead time. According to (18), with:

$$\tau = \frac{T_{DT}}{T_{DT} + T_{Mean}} = \frac{2 \times 80 \mu s}{2 \times 80 \mu s + 6000 \mu s} = 0.03 \quad (25)$$

The control circuit was strongly lag-dominant. The control circuit features a fast settling time of  $t_s = 2$  ms, a gain margin of  $\frac{1}{0.05} = 20 \triangleq 26$  dB, and a phase margin of about  $88^\circ$ , thereby representing a fast and robust control design.



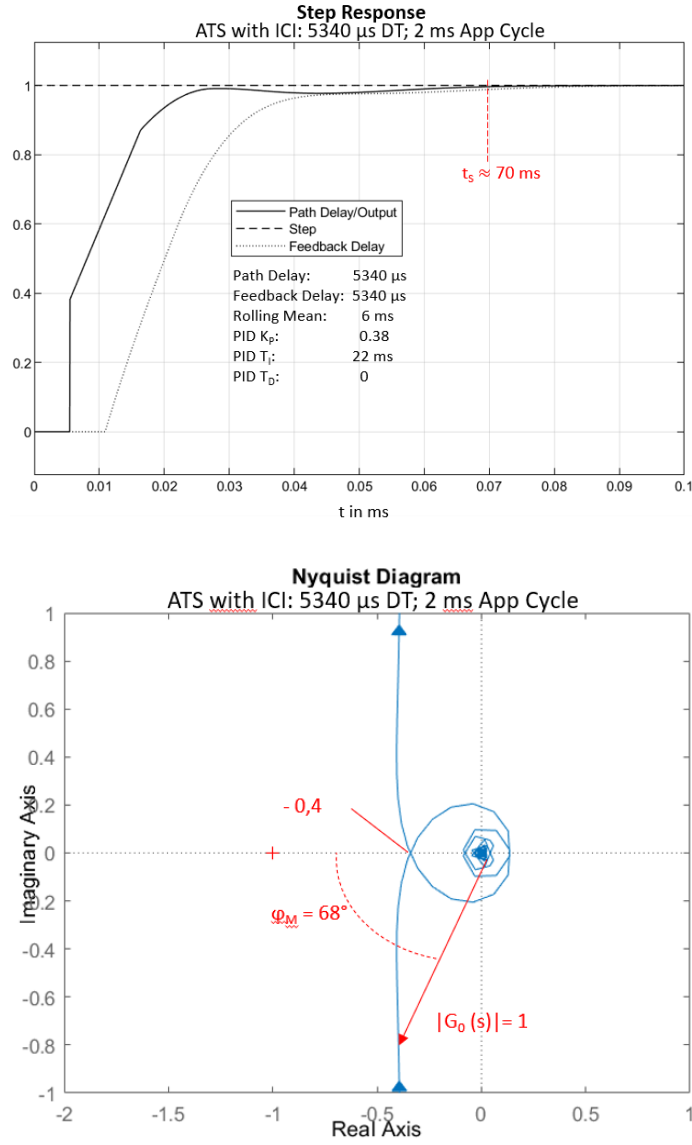


**Fig. 7. Step response and Nyquist diagram for SPQ with ICI**

Fig. 7 shows the step response and Nyquist diagram for SPQ with maximum ICI. It represents a control circuit with medium dead time of 570  $\mu$ s for both path dead time and feedback dead time. According to (18), with

$$\tau = \frac{T_{DT}}{T_{DT} + T_{Mean}} = \frac{2 \times 570 \mu s}{2 \times 570 \mu s + 6000 \mu s} = 0.16 \quad (26)$$

the control circuit remains a lag-dominated network. It features a quite fast settling time of  $t_s = 10 \text{ ms}$ , a gain margin of  $\frac{1}{0,2} = 5 \triangleq 14 \text{ dB}$ , and a phase margin of approximately  $75^\circ$ , representing a fast and robust control design.



**Figure 8. Step response and Nyquist diagram for ATS with maximum ICI**

Figure 8 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  $\mu$ s for the path dead time and feedback dead time. According to (18), with

$$\tau = \frac{T_{DT}}{T_{DT} + T_{Mean}} = \frac{2 \times 5340 \mu s}{2 \times 5340 \mu s + 6000 \mu s} = 0.64, \quad (27)$$

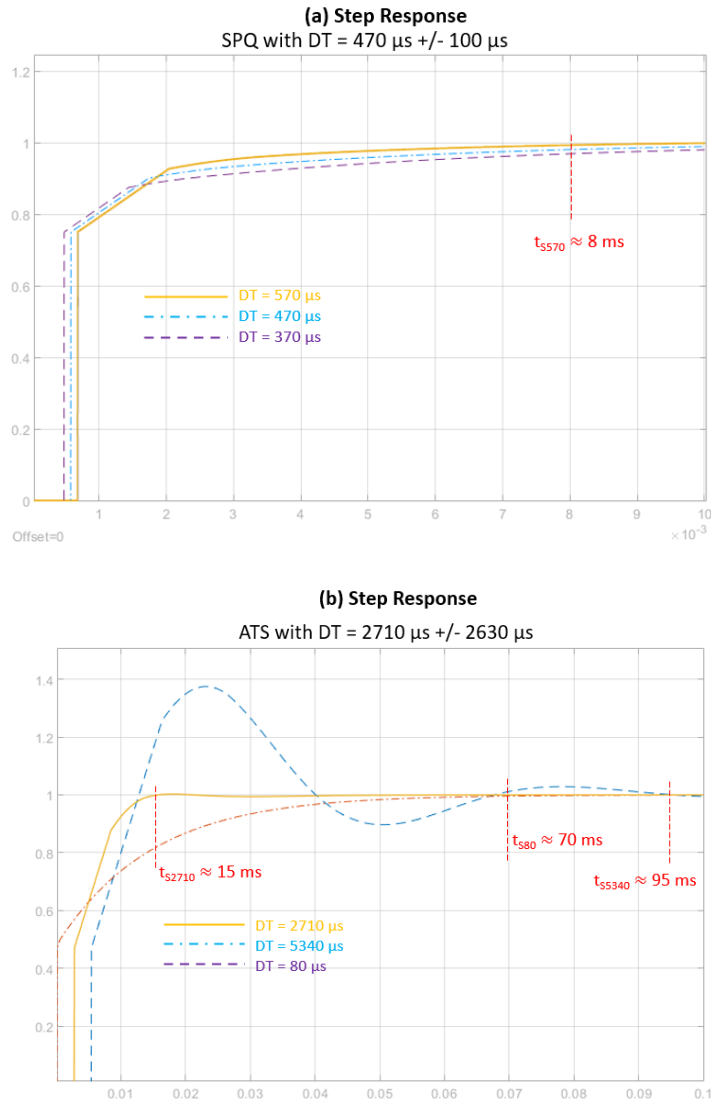
the control circuit is at the border of being dead-time dominant. The settling time has worsened to 70 ms, the gain margin to  $\frac{1}{0,4} = 2,5 \triangleq 8$  dB, and the phase margin to  $68^\circ$ , representing a control design at the border of robustness.

Fig. 6 to Figure 8 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. Table 4The results are summarised in Table 4.

**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 $T_{DT}$ ( $\mu$ s)	settl. time $t_s$ (ms)	gain margin $g_M$ (dB)	phase margin $\varphi_M$ ( $^\circ$ )	ctrl robustness
EST without ICI	80	2	26	88	high
SPQ with maximum ICI	570	10	14	75	medium
ATS with maximum ICI	5340	70	8	68	low

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. 9 (a) shows an example of the step response for the SPQ with 50 per cent ICI. A dead-time of  $470 \mu$ s 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. 9. Dynamic performance deviation depending on dead time uncertainties**

Fig. 9 (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  $t_s = 10$  ms for both  $370 \mu$ s and  $470 \mu$ s dead time compared to 8 ms

for the  $570\ \mu\text{s}$  tuned control circuit. Fig. 9 (b) shows the result if the same test case is applied to a tuned control circuit featuring ATS. An average medium dead time of  $DT = 2,710\ \mu\text{s}$ , a possible deviating minimum dead time of  $DT = 80\ \mu\text{s}$  and a maximum dead time of  $DT = 5,340\ \mu\text{s}$  is assumed. The considerably deviating dead times provoke substantial deviations in settling time of  $t_s = 70\ \text{ms}$  for  $T_{DT} = 80\ \mu\text{s}$ , and  $T_s = 95\ \text{ms}$  for  $T_{DT} = 5,340\ \mu\text{s}$ , compared to the tuned settling time of  $t_s = 15\ \text{ms}$ . In addition, the higher actual dead time produces a considerable overshoot of 40 %.

Summarizing the influence of ICI, it can be stated, as Fig. 9 (a) shows, that ICI has a very low influence on the control quality for SPQ. The small dead time deviation tolerance band of only  $\pm 20\%$  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. 9 (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 poor settling times and a high overshoot of the flow control circuit.

## 6. 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 frame preemption 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 the 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  $2\ \text{ms}$  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 optimised 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] N. Finn, "Introduction to Time-Sensitive Networking," *IEEE Communications Standards Magazine* 2, 2018.
- [2] *IEEE Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications*, IEEE 802.1AS, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2020.
- [3] *IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks*, IEEE 802.1Q, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2018.
- [4] *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks, Amendment: Forwarding and Queuing Enhancements for Time-Sensitive Streams*, IEEE 802.1Qav, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2009.
- [5] *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks, Amendment: Frame Preemption*, IEEE 802.1Qbu, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2015.
- [6] *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks, Amendment: Enhancements for Scheduled Traffic*, IEEE 802.1Qbv, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2015.
- [7] *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks, Amendment: Path Control and Reservation*, IEEE 802.1Qca, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2015.
- [8] *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks, Amendment: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements*, IEEE 802.1Qcc, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2018.
- [9] *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks, Amendment 29: Cyclic Queuing and Forwarding*, IEEE 802.1Qch, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2019.
- [10] *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks, Amendment: Per-Stream Filtering and Policing*, IEEE 802.1Qci, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2016.
- [11] *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks, Amendment: Asynchronous Traffic Shaping*, IEEE 802.1Qcr, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2020.
- [12] *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks, Amendment: Resource Allocation Protocol*, IEEE 802.1Qdd, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2022.
- [13] H. Wang et al., "COPE traffic engineering in dynamic networks. *ACM SIGCOMM Computer Communication Review*, vol. 36, p. 99, 2006.
- [14] A. Elwalid, C. Jin, S. Low, I. Widjaja, "MATE: multipath adaptive traffic engineering," *Computer Networks* vol. 40, pp. 695-709, 2002.
- [15] I. Ahmad, S. N. Karunaratna, M. Ylianttila, A. Gurtov, "Load Balancing in Software Defined Mobile Networks" in *Software Defined Mobile Networks (SDMN) : Beyond LTE Network Architecture*, John Wiley and Sons, Ltd : Chichester, UK, 2015, pp. 225-245.
- [16] *TSN Profile for Industrial Automation, Use cases*, IEC/IEEE 60802, 2018.
- [17] D. Lopez-Perez et al., "Long Term Evolution-Wireless Local Area Network Aggregation Flow Control," *IEEE Access* vol. 4, 2016.
- [18] M. J. Neely, C. P. Li, E. Modiano, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Transactions on Networking* vol. 16, pp. 396-409, 2008.
- [19] S. Kandula, D. Katabi, B. S. Davie, A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering", in *Computer Communication Review*, vol. 35, pp. 253-264, 2005.
- [20] S. Mascolo, "Smith's principle for congestion control in high-speed data networks, " *IEEE Transactions on Automatic Control*, vol. 45, 2000.

- [21] M. Collotta, "FLBA: A fuzzy algorithm for load balancing in IEEE 802.11 networks," in *Journal of Network and Computer Applications*, vol. 53, pp. 183-192, 2015.
- [22] D. Pompili, F. D. Priscoli, "A closed-loop fuzzy traffic controller for fair bandwidth sharing", in *ACM SIGBED Review*, vol. 5, pp. 1-6, 2008.
- [23] Z. Zhang, X. Zhang, "A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation", in *The 2nd International Conference on Industrial Mechatronics and Automation*, pp. 240-243, 2010
- [24] C. Yu-Jia, W. Li-Chun, C. Meng-Chieh, H. Pin-Man, C. Pei-Jung, "SDNEnabled Traffic-Aware Load Balancing for M2M Networks", in *IEEE Internet of Things Journal*, vol. 5, 2018.
- [25] N. G. Nayak, F. Dürr, K. Rothermel, "Routing algorithms for IEEE802.1Qbv networks", *SIGBED Rev.* vol. 15, pp. 13–18, 2018.
- [26] M. A. Ojewale, P. M. Yomsi, "Routing heuristics for load-balanced transmission in TSN-based networks", *ACM SIGBED Review*, vol. 16, pp. 2025, 2020.
- [27] F. A. R. Arif, T. S. Atia, S., "Load balancing routing in Time-Sensitive Networks", in *Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S and T) (IEEE)*, pp. 207-208, 2016.
- [28] A. Nasrallah, "Performance Comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS)", *IEEE access*, vol. 7, pp. 44165-44181, 2019.
- [29] *IEEE Standard for Local and Metropolitan Area Networks—Frame Replication and Elimination for Reliability*, IEEE 802.1CB, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2017.
- [30] *Industrial communication networks –Fieldbus specifications*, IEC 611585-10, Part 5-10: Application layer service definition – Type 10 elements, International Electrotechnical Commission, 2019.
- [31] *Industrial communication networks –Fieldbus specifications*, IEC 611586-10, Part 6-10: Application layer protocol specification – Type 10 elements, International Electrotechnical Commission, 2019.
- [32] Z. Zhou, Y. Yan, M. Berger, S. Ruepp, "Analysis and modeling of asynchronous traffic shaping in time sensitive networks", in *14th IEEE International Workshop on Factory Communication Systems (WFCS) (IEEE)*, pp. 1-4, 2018.
- [33] *IEEE Standard for Ethernet - Amendment 5: Specification and Management Parameters for Interspersing Express Traffic*, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2016.
- [34] J. E. Normey-Rico, E. F. Camacho, "Control of dead-time processes," *Advanced textbooks in control and signal processing*, Springer, London, 2007.
- [35] K. L. Chien, Hrons, J. A., Reswick, J. B., "On the Automatic Control of Generalized Passive Systems," in *Transactions of the American Society of Mechanical Engineering*, vol. 74, pp. 175-185, 1972.
- [36] *Commercial Building Telecommunications Cabling Standard*, ANSI/TIA568.1-D, American National Standards Institute, Washington, D.C., USA, 2015.
- [37] L. De Cicco, S. Mascolo, S.-I. Niculescu, "Robust stability analysis of Smith predictor-based congestion control algorithms for computer networks", in *Automatica*, vol. 47, pp. 1685-1692, 2011.
- [38] *Industrial communication networks - High availability automation networks - Part 2: Media Redundancy Protocol (MRP)*, IEC 62439-2, International Electrotechnical Commission, 2021.

## Author Biographies

**Thomas Weichlein** received his Diplom-Ingenieur (FH) degree in Electrical Engineering from the Georg-Simon-Ohm University of Applied Sciences in Nürnberg, Germany, in 1990. Since 2016, he is a PhD student at the University of Gloucestershire, UK. Since 1990, he worked in various positions within research and development for industrial automation systems at the companies Siemens and Danfoss. He currently holds the position of a system architect for industrial automation and communication systems at Siemens Digital Industries in Nürnberg, Germany. His research and professional focus are on media redundancy, time synchronisation, and time sensitive networks and their coexistence with 5G and DetNet networks. In addition, he is involved in the definition of stream reservation protocols and network user interfaces.

**Dr Shujun Zhang** is the professor of applied computing and technology at the School of Computing and Engineering at the University of Gloucestershire, UK. He has worked at the University of Warwick more than 10

years, Salford University for 3 years and Jilin University for 6 years. He has published about 150 papers and undertaken more than 40 funded projects. His research interests are in the areas of applied computing and bionics engineering and their applications to both engineering and business management, covering (1) the various smart system design and development, (2) intelligent algorithms, (3) big data and data mining, (4) engineering system modelling, analysis, simulation optimisation and control, (5) physiological signal capturing and analysis (such as cell pulsation wave forms, frequency and strength, ECG, EMG and EEG etc), and (6) bio-inspired physiological signal-based algorithms and systems for both healthy and cybersecurity applications.

**Dr Pengzhi Li** is currently a lecturer in Industrial Systems, Mechatronics and Robotics at the University of Gloucestershire, UK. Before joining the University of Gloucestershire, he was a Postdoctoral Research Associate in Robotics for Extreme Environments Lab at the University of Manchester after receiving his PhD from University of Chinese Academy of Sciences in 2019. Before that, he had worked in Changchun Institute of Optics, Fine Mechanics and Physics of the Chinese Academy of Sciences as a researcher from 2011 to 2019. His research interests are mainly focused on precision mechatronics, robotic manipulation and machine learning, autonomous mobile robots, and advanced control systems.

**Dr Xu Zhang** is currently working as a Senior Research Fellow in the Rolls-Royce UTC at the University of Southampton after receiving his PhD from the University of Gloucestershire and MSc with Distinction from the University of Warwick. He is carrying out the multidisciplinary research to provide solutions to aero-engine optimisation challenges for Rolls-Royce. His research interests cover artificial intelligence, image/video processing, and algorithm in Computer Science as well as model parameterisation, design optimisation, and computational fluid dynamics in Applied Engineering.