http://dx.doi.org/10.25673/39568

Investigating the Inter-Domain Forwarding Offset in the Context of Dynamic End-to-End Stream Reservation in Multi-Domain Time Sensitive Networks

Martin Böhm, Diederich Wermser

Research Group Communication Systems
Ostfalia University of Applied Sciences
Salzdahlumer Str. 46/48
38302 Wolfenbüttel
ma.boehm@ostfalia.de
d.wermser@ostfalia.de

Abstract: The constantly growing TSN toolbox offers runtime reconfiguration for dynamic end-to-end stream reservation. While inter-domain communication in TSN has been identified to be required for various use cases, the control plane interactions are not defined yet. This paper presents Multi-Domain Time-Sensitive Networks (MDTSNs) integrating east-westbound communication in the TSN control plane to enable on-demand multi-domain end-to-end bounded-latency stream configuration. An inter domain forwarding offset (IDFO) has been identified to emerge when setting up streams in MDTSN. This paper investigates the IDFO, presents control plane mitigation mechanisms which are implemented and evaluated in a physical MDTSN test environment.

1 Introduction

Inter-domain communication for IEEE 802.1 Time-Sensitive Networking (TSN) has been identified to be required for various use cases gathered by the IEC/IEEE 60802 TSN Profile for Industrial Automation [TSIA] (TSN-IA Profile). E.g., one use case is described [IAUC] where preconfigured machines with tested and approved communication, require communication with other preconfigured machines located in different TSN-domains. So far, the TSN-Toolbox offers features working in single TSN domain i.e., within the domain boundaries. Towards the future of industrial automation networks, infrastructure components will be deployed and changed on demand at runtime [WSJ17]. This requires automation networks to provide dynamic reconfiguration of communication.

The IEEE 802.1Qcc [IEQC] standard offers runtime reconfiguration allowing users to specify stream requirements. The standard introduces three different configuration models for TSN. The *fully centralized* model, shown in Figure 1, which is in focus of this work, introduces a Centralized User Configuration (CUC) which handles end station stream requests and end station TSN feature configuration, while a Centralized Network Configuration (CNC) performs traffic scheduling and network configuration for TSN bridges. Applying the separation between the data plane and the control plane to TSN, as inherited from Software Defined Networking (SDN), the CUC and the CNC are part of the TSN control plane, while end stations and TSN bridges are part of the TSN data plane. As pointed out by Schriegel et al., for the Industrial Internet of Things (IIoT), the information technology (IT) and operational technology (OT) must coalesce. This requires an interface between the networks' control plane [SKJ18]. Such an interface, for the interaction between different TSN domains, has not been addressed by TSN standards yet.

In [BW21] we presented control plane mechanisms for MDTSN integrating an east-westbound protocol in the existing TSN control plane, achieving multi-domain on-demand end-to-end bounded-latency TSN stream configuration in the fully centralized model for unidirectional periodic traffic. In this work, we present a multi-domain use case, where different Manufacturing as a Service (Maas) provider, with machines located in different TSN domains have to cooperate with each other i.e., require streams between different TSN domains. We present control plane mechanisms for the integration of east-westbound communication in the MDTSN control plane. A main challenge in this context is the inter-domain forwarding offset (IDFO) on the MDTSN data plane, which will investigated in detail. Control plane mechanisms to mitigate the IDFO are presented and evaluated in an MDTSN test environment.

The paper is structured as follows: first, Chapter 2 presents TSN in the context of dynamic stream reservation. Use cases for MDTSN are presented in Chapter 3. Related work is presented in Chapter 4. Chapter 5 presents MDTSN and control plane mechanisms for dynamic stream reservation in MDTSN. Chapter 6 investigates the IDFO, as part of MDTSN. Control plane mechanisms to mitigate the IDFO are presented in Chapter 7. These mechanisms are tested and evaluated in an MDTSN test environment, as presented in Chapter 8. Chapter 9 concludes paper and identifies future work.

2 Time Sensitive Networking

Within IEEE 802.1, enhancements for Ethernet based networks are specified. These enhancements in particular provide real-time capabilities and can be combined in different ways. A bounded latency, low jitter and low packet loss can be achieved by the use time slots with cyclical repetition (IEEE 802.1Qbv [QBV]). This allows to grant specific traffic classes (TCs) exclusive use for the data transmission within a time slot. Traffic is assigned to different queues which open and close for a certain time, called schedule, which are specified Gate Control Lists (GCLs). A stream is an end-to-end connection in TSN, identified by, e.g., MAC addresses. A network-wide time synchronization (IEEE 802.1AS-rev [ASRe]) is required to synchronize GCLs of the devices to properly schedule traffic. The generalized precision time protocol (gPTP), which is based on the precision time protocol (PTP), is specified in the IEEE 802.1AS-rev standard in the context of TSN.

The IEEE 802.1Qcc standard introduces enhancements for the Stream Reservation Protocol (SRP) (IEEE 802.1Qat) for dynamic stream reservation. Besides performance improvements, it presents three different architectural models. Within the *fully distributed* model, application request streams by propagating a stream request directly over the network. In a distributed manner, each TSN bridge along a path configures itself with the communication parameters of the stream request. The second presented model is the *centralized network/distributed user* model. Here, a central entity called Centralized Network Configuration (CNC) is introduced. Stream requests are still propagated over the network, while the first bridge forwards the request to the CNC. For scheduled traffic (IEEE 802.1Qbv), the CNC, with a centralized view on the network, takes care of path finding, traffic scheduling and configures all related TSN bridges affected by the stream request. For more complex use cases, where end stations (talker and listener) also require configuration, the *fully centralized* model, shown in Figure 1, introduces a Centralized User Configuration (CUC). The CUC discovers end stations and their capabilities. Streams are requested directly at the CUC. The CUC communicates the stream requests with the CNC. Furthermore, the CNC provides configurations for the end stations, configured by the CUC. The communication interface between the CUC and the CNC, called user network interface (UNI), is specified by the IEEE while the communication between the end stations and the CUC is application specific. For example, the OPC Foundation specified their PubSub architecture to be compatible with TSN [Fou16]. They also specified a CUC called *PubSub TSN Configuration Broker* (PTCB) [Fou17].

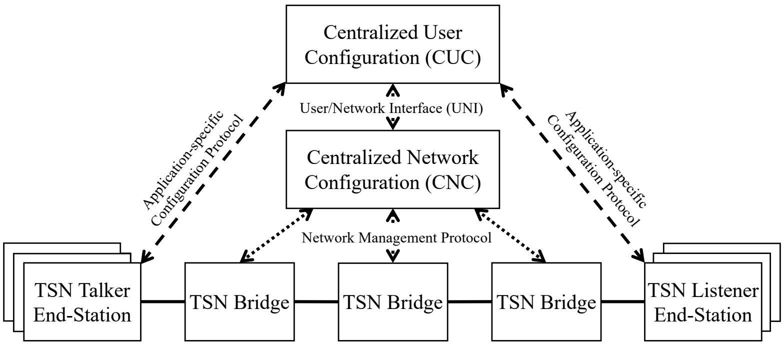


Figure 1: TSN in the fully centralized model

In the TSN-IA Profile an Industrial Automation Management Entity (IA-ME) is presented, which is adapted from the fully centralized model to serve industrial automation use cases and partitions the CUC and the CNC for better understanding. The IA-ME introduces new entities besides the already specified CUC and the CNC. A Topology Discovery Engine (TDE), an Industrial Automation Path Establishment Entity (IA-PE), responsible for path management, and a Best Management Entity Algorithm (BMEA) for the management of multiple IA-ME for e.g., failover, are added to the IA-ME. Within IA-ME, end stations request streams at a Query Stream Server (QSS) which handles the communication with the CUC. The authors of the profile note, that the TDE and the IA-PE could be considered as part of the CNC as well as the QSS could be part of the CUC. This paper assumes the architecture described in IEEE 802.1Qcc.

2.1 Stream Parameter

In a time-sensitive stream request, as described in IEEE 802.1Qcc, there are lots of control plane (communication) parameters involved. The most relevant parameters are further described. Parameters for time awareness are specified within a network cycle, which is a repetitive time interval used in the network for e.g., opening and closing gates of queues. In a stream request from talker to the CUC, the talker specifies the following parameters.

- Source and destination MAC address: MAC address of the talker and the listener of a requested stream.
- Interval: Interval for transmitting data (e.g., 125µs). Usually within predefined traffic classes.
- Maximum frames per interval: Maximum number of frames that will be sent during the interval.
- Maximum frame size: Maximum frame size sent by the talker for the requested stream.
- Earliest Transmit Offset (ETO): Earliest offset within the network cycle where talker is capable of transmitting data.
- Latest Transmit Offset (LTO): Latest offset within the network cycle where the talker is capable of transmitting data.
- Maximum Latency: Maximum latency from talker to listener for a single frame.

After a successful stream request, in the talker configuration, a *Transmit Offset* (also called time aware offset) is specified by the CNC. The value is between the requested ETO and LTO. It specifies a point of time within the network cycle for the transmission of the first frame of the requested stream.

The listener configuration includes an accumulated latency, which specifies a point of time within the network cycle, when the first frame of the configured stream arrives. It is calculated by the CNC by adding up the configured bridge delays and the propagation delays of each bridge along the stream path.

2.2 Industrial Traffic Types

The TSN-IA Profile specifies traffic types for different functionalities. They have different characteristics and requirements on the network. Table 1 depicts a subset of the traffic types. The table shows the name of the traffic type, the periodicity (periodic or sporadic) and the typical period of the application cycle (data transmission interval). Furthermore, it specifies the requirements for the data delivery guarantee. This can be for example a required maximum latency, bandwidth, or no guarantees at all. Some traffic types require the application to be synchronized to the network clock for e.g., scheduled traffic. The table also specifies the typical application data sizes and the criticality.

This work focusses on the most critical traffic types, isochronous and cyclic-synchronous. Isochronous traffic, typically used for control-loop communication, is required to be synchronized to the network clock to reduce jitter, which requires this traffic type to not interfere with other traffic. Messages may be discarded when delivered too late. Cyclic-synchronous traffic has slightly less strict communication requirements compared to isochronous traffic. Its period with slightly larger and it can handle a small jitter.

Table 1: Industrial automation traffic types [IEE, Ind]

Traffic type name	**Periodicity**	**Typical period**	**Data delivery guarantee**	**Synchronized to network**	**Typical application data sizes (Bytes)**	**Criticality**
Isochronous	Periodic	100µs-2ms	~1 application cycle	Yes	Fixed: 30-100	High
Cyclic-Synchronous	Periodic	500µs-1ms	~1 application cycle	Yes	Fixed: 50-1000	High

Cyclic-Asynchronous	Periodic	2ms-20ms	~½ application cycle	No	Fixed: 50-1000	High
Alarms and Events	Sporadic	N/A	100ms-1s	No	Variable: 50-1500	High
…						
Best Effort	Sporadic	N/A	None	No	Variable: 30-1500	High

3 Use Cases for TSN Inter-Domain Communication

In [IAUC], over 35 use cases are specified by the IEC/IEEE 60802 project. Several use cases have been identified, which require inter-domain communication within TSN. E.g., one use case focussing on redundancy is described, where multiple TSN-domains are interconnected via a ring topology. Another use case for machine to machine (M2M) communication and controller to controller (C2C) communication is described for inter-domain communication. Machines are grouped and located in their own TSN domains, because they e.g., run different complex schedules with different network cycles. A communication between machines located in other TSN domains is required. Production cells, which connect the TSN domains of the machines, are also located in their own TSN domains. These production cells are interconnected in a production line TSN domain while the Operation Control HMI is also located in its own TSN domain. New devices, e.g., automatic guided vehicles (AGVs), are plugged/unplugged automatically, requiring communication streams on demand. Technically, the document describes, that the TSN domains share a single OSI layer 2 broadcasting domain.

Figure 2 presents our vision of the future factory, where machines of different MaaS providers have to work together. Within MaaS, manufacturing processes are outsourced to a MaaS provider. To prevent vendor lock-in, when relying on a single MaaS provider, while also benefitting on different specialities and pricing of different MaaS providers, a combination of multiple MaaS provider should be intended. Open standard solutions such as TSN and OPC UA, allow communication between machines of different vendors.

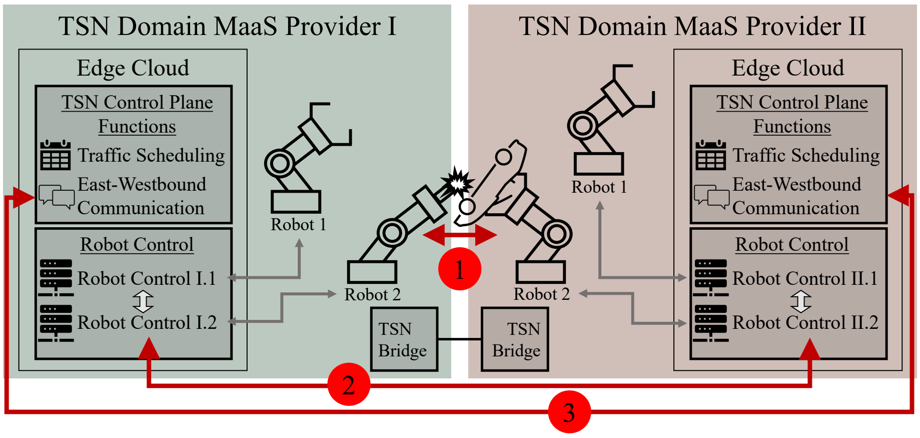


Figure 2: Cooperating robots of Manufacturing as a service (MaaS) providers in interconnected TSN domains

As visualized in Figure 2, each MaaS provider is located in its own TSN domain. A provider specific edge cloud provides TSN control plane function such as traffic scheduling as well as the robot controls. Each MaaS providers solution is part of their business secret and has to be kept confidential. The interaction for the combination of different MaaS providers is highlighted in red color in Figure 2. 1) Robots physically cooperate with each other. 2) Communication with bounded latency between the robot control systems of the providers is required. 3) Dynamic streams are negotiated and configured in the TSN control plane over an east-westbound protocol.

Our developed use case has different requirements compared to the use cases identified by the IEC/IEEE 60802 project. To preserve the internal configuration of the MaaS providers, the TSN domains cannot form an OSI layer 2 broadcast domain. Thus, topology hiding is required. Each multi-domain stream has to be requested and configured separately, as further described in this paper.

4 Related Work

Traffic scheduling, as part of the CNC, is an NP-complete problem. Depending on the size of the network and the streams, the traffic schedule calculation may require hours [CO16]. When streams are scheduled in runtime, the calculation uses the existing configurations to add new streams. Compared to a complete traffic schedule calculation, the runtime schedule calculation is faster as presented in [RP17], where the proposed heuristic scheduling approach is done within seconds. Their scheduling algorithm uses an as-soon-as-possible (ASAP) approach, where streams are scheduled at the earliest point in time, that is feasible. In the worst case, the scheduler destroys the current configuration and reschedules all flows.

In SDN, east-westbound protocols are used for the communication between distributed domain controllers to enable e.g., coordinated flow setups. While there is OpenFlow, the most common standardized southbound protocol for SDN, so far, there has been an expired IETF draft in 2012, called SDNi [SDNi], for an east-westbound protocol in SDN. The sketched protocol proposes e.g., domain-specific policies, reachability information exchange and coordinated flow setups including QoS. While there has been lots of research on east-westbound communication in SDN [ZC18], there is no standardized protocol yet.

In [SKJ18], a distributed control plane for heterogenous TSN is proposed, which requires inter-domain communication. The authors describe the idea of cascaded CNCs which perform stream configurations. Without a coordinated stream setup, which requires communication between the CNCs, the end-to-end latency increases. Mechanisms for the interaction between the CNCs are not covered.

In the IEC/IEEE 60802 project, there have been some contributions about inter-domain communication. In [Lih], an idea for a protocol between CNCs is proposed, called CCP (Config-entity to Config-entity Protocol or CNC to CNC Protocol). The authors point out, that protocol, procedures, and managed objects must be specified for CCP.

5 Multi-Domain Time-Sensitive Networks (MDTSNs)

The audio video bridging (AVB) standard (IEEE 802.1BA [IEBA]) specifies an automatic AVB domain detection mechanism. Domain boundaries are detected by a bridge when a different SRP domain on a network port is detected. This mechanism works for the fully distributed TSN configuration model. In the fully centralized TSN configuration model, TSN domains are formed by connected TSN bridges to their respective CNC.

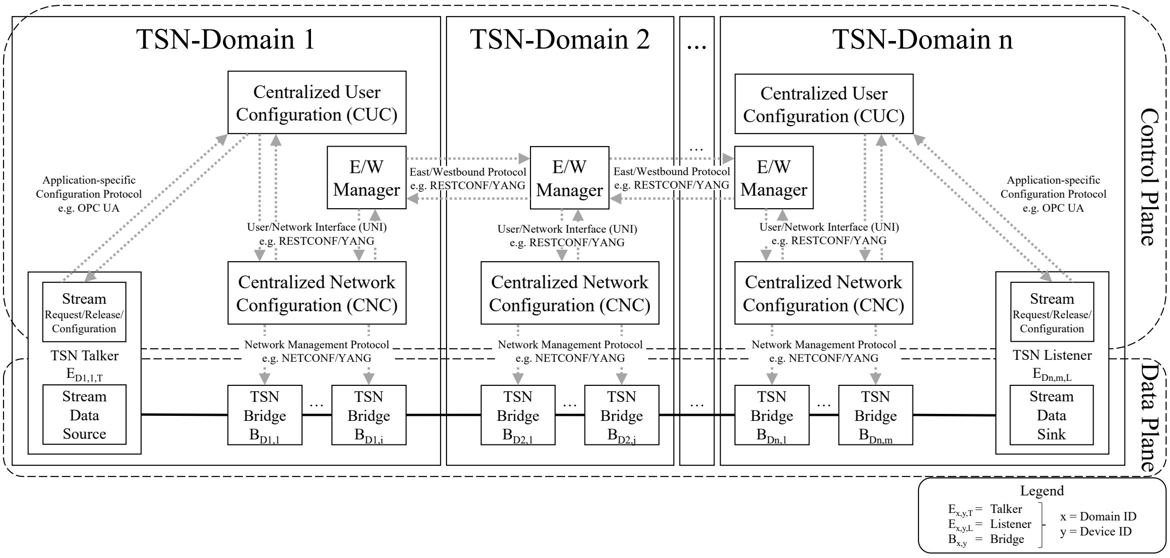


Figure 3: East-westbound communication in the multi-domain time-sensitive networks (MDTSN) control plane

In MDTSN, on-demand end-to-end TSN streams are set up between different TSN domains. Thus, the interaction on the TSN control plane must be defined. Figure 3 shows MDTSN in a horizontal, also called peer-to-peer architecture. Another architecture, not in focus of this work, is the hierarchical architecture, where control plane communication for inter domain communication is handled by a root controller, as presented for SDN in [ZC18]. The horizontal architecture has been chosen because of its advantages with respect to topology hiding. Figure 3 shows an *east/westbound manager* (E/W Manager) added in the TSN control plane enabling inter domain communication. Compared to SDN, an east-westbound protocol in TSN requires to exchange timing information for dynamic stream configuration.

5.1 Stream Segments

In MDTSN an end-to-end stream consists of multiple stream segments, which are streams provided by each TSN domain as part of the end-to-end stream. We identified three different types of TSN domains which are involved in an MDTSN stream request. The *source domain* is the domain, where the talker end station of a stream request is located. Stream are provided between the end station and the *boundary network port*. Boundary network ports interconnect two TSN domains. The *forwarding domains* are domains, where neither the talker end-station, nor the listener end-station is located. Therefore, the forwarding domain provides a stream between boundary ports. A forwarding domains purpose may only be forwarding i.e., it has no end-stations and no CUC. Within an MDTSN stream, there may be multiple forwarding domains involved. The *destination domain* is the domain, where the listener end-station is located. Stream are provided between a boundary port and the listener end-station. Within the MDTSN stream request, timing information of that stream are forwarded to the next domain.

Due to topology hiding, the CNC, which usually is supposed to configure streams between two end-stations, has no knowledge about the existence of all end-stations involved in an MDTSN stream request. Here, two new parameters are introduced for stream requests: the *ingress MAC address* and the *egress MAC address*. These parameters specify the boundary network port MAC addresses between two adjacent domains. The E/W Manager adds the parameters if required.

5.2 Reachability Information Exchange

Only the CNC has information about the network's topology. Due to topology hiding, this information is limited to the local domain. To enable MDTSN stream requests, the E/W Manager requires reachability information exchange over the east-westbound protocol. Due to topology hiding, this information should be limited. Information required for MDTSN with respect to topology hiding is the topology of interconnected domains, the directly connected boundary network port MAC addresses, a list of local end-stations (available for multi-domain communication) and a list of (disclosed) end-stations of directly and indirectly connected domains.

5.3 MDTSN Stream Request

The sequence diagram in Figure 4 illustrates the stream request procedure for a forwarding domain in MDTSN. It shows the interaction between the previous domain and the next domain of the stream request. (1) An E/W Manager receives a stream continuation requested by a neighboring E/W Manager. (2) Stream parameters are determined by the E/W Manager. This includes validating the existence of the end-stations, the determination of the next domain and its boundary network port. (3) A stream segment reservation request is sent to the local CNC. Note, that this stream request is a reservation request, without a direct configuration. This reduces data plane reconfigurations in case a domain later in the chain of the MDTSN stream request is not able to provide a stream. In (4), the CNC performs traffic scheduling and calculation of network configurations. (5) Scheduling results are provided as a response of the CNC to the E/W Manager. (6) Stream parameters for the next domain are determined including the recalculation of the remaining requested maximum latency, the ingress MAC address, ETO and LTO. (7) A stream continuation request is sent to the E/W Manager of the next domain. (8) Further domains on the path to the listener end-station are trying to provide a stream. (9) When successful, the stream configuration is confirmed to the E/W Manager. (10) The E/W Manager requests the CNC to configure the reserved stream segment. (11) The CNC configures the affected TSN bridges with a new configuration. (12) The successful configuration is confirmed to the E/W Manager. (13) The E/W Manager confirms the successful stream configuration from the listener end-station back to the current domain.

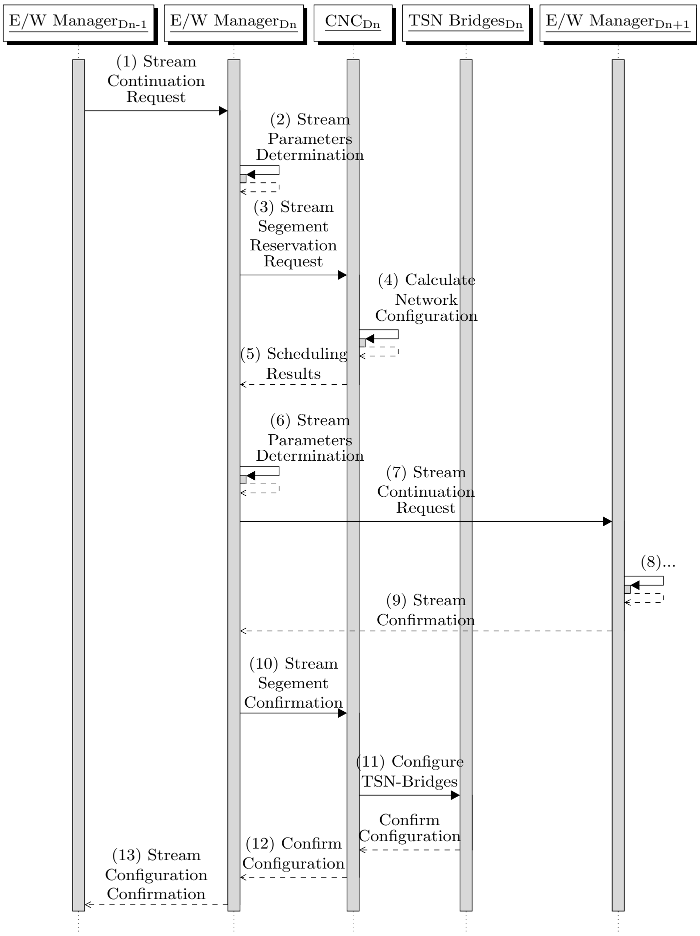


Figure 4: Stream request procedure for a forwarding domain in MDTSN

6 Inter-Domain Forwarding Offset (IDFO)

Figure 5 shows a configured stream on the MDTSN data plane exemplified with three TSN domains. Each domain involved in this stream is responsible for proper scheduling of its stream segment, while notifying the next domain about the periodical arrival time to continue the end-to-end stream setup. Figure 5 depicts, that domain 2 and domain 3 are *wait for forwarding* at the point in time, when messages arrive at its domain, as scheduled, and notified by the previous domain. Depending on the current utilization of the TSN bridges at the point of time of the stream setup, traffic may not be directly scheduled, due to already reserved resources within the network cycle. This idle time leads to an inter-domain forwarding offset (IDFO). When requesting two streams with identical Stream Request Parameter Sets (SRPS) at a different Stream Request Time (SRT), due to different utilizations of the TSN bridges and different sizes of the IDFOs, the streams actually provided end-to-end latency is different. When TSN domains have high traffic loads, this may even lead to unsuccessful stream configurations, where the requested maximum end-to-end latency cannot be achieved. Each domain within an MDTSN stream may have an IDFO for that stream. The size of the IDFOs are not limited and may have the size of multiple network cycles. Depending on the traffic class and the required maximum latency, for cyclic-synchronous traffic, a sum of large IDFOs may work. For isochronous traffic, the required end-to-end latency has the size of one application cycle, which's size is close to the network cycle. Thus, large IDFOs lead to unsuccessful stream requests. Control plane mechanisms to mitigate the size of the IDFOs will further be investigated.

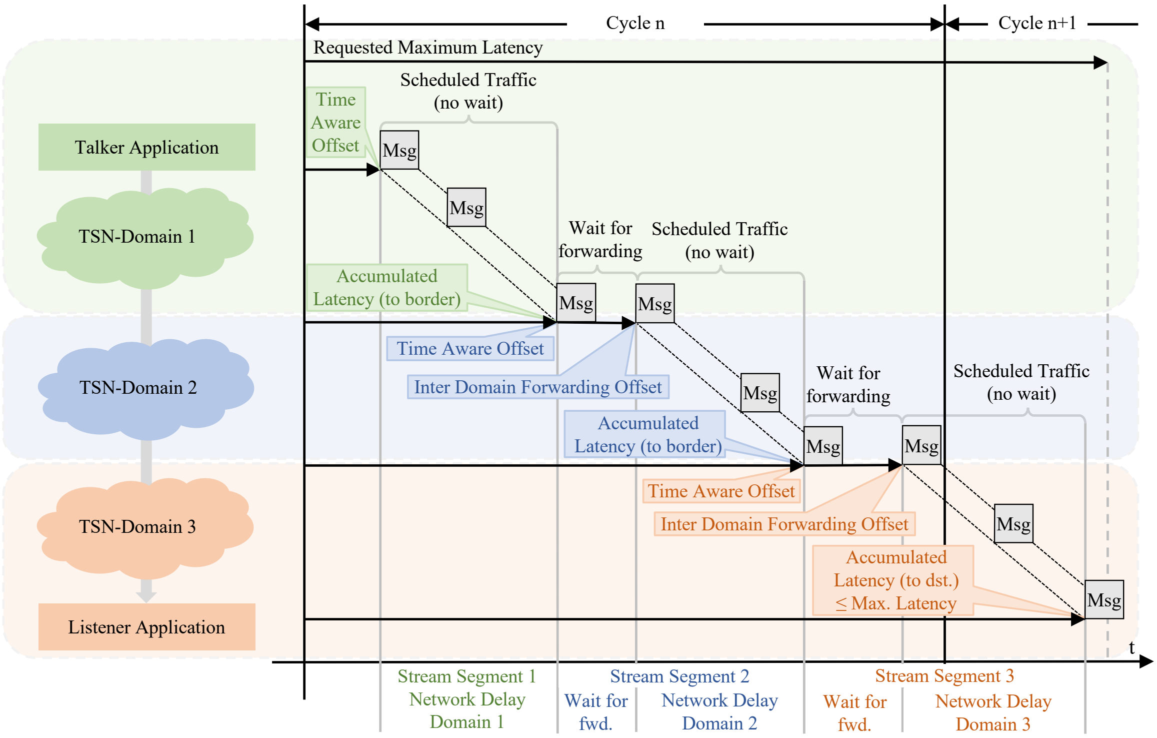


Figure 5: Data plane configuration of an MDTSN stream exemplified with three TSN domains

7 IDFO Mitigations

In MDTSN, each domain provides a stream segment of an end-to-end stream. As presented in the previous chapter, IDFOs emerge in MDTSN, increasing the end-to-end latency. Each IDFO reduces the remaining maximum latency for the next domains. Within the goal of successful stream configurations, the sum of the IDFOs should be minimized. Further, we present different control plane mechanisms to mitigate the IDFO:

1. **Maximum latency sharing**: Sharing the requested maximum latency of a stream request by e.g., diving the requested maximum latency by the number of remaining domains. This mechanism restricts each domain to provide a stream within the reduced maximum latency.

2. **Iterative stream requests**: Whenever a stream request is rejected by a domain, the stream is iteratively requested again by the first domain. Here, the ETO and LTO are restricted for the first domain within the goal to achieve different ETO/LTO for the next domain. Note, that the ETO and LTO increase the flexibility for scheduling, while in MDTSN, only the first domain can make use of this flexibility. For all further domains, the arrival time is fixed, i.e., the ETO and LTO are identical.

3. **Multiple stream configuration offers**: Similar to iterative stream requests, the CNC of the first domain is requested with different ranges of the ETO and LTO to achieve multiple stream configuration offers. Further domains have increased flexibility.

Further, the presented control plane mechanisms are discussed.

1. **Maximum latency sharing**: For scheduling approaches using ASAP scheduling [RP17], where streams are scheduled at the earliest point in time that is feasible, the sharing mechanism forces the scheduler to find a stream reservation with reduced latency. For scheduling approach intending to reduce the end-to-end latency, this mechanism may be ineffective.
2. **Iterative stream requests**: The computational complexity for the schedule calculation rises with the number of existing streams and the size of the network. Nevertheless, due to dynamic stream reservation, adding new streams is less complex than a full schedule calculation [RP17]. Iterative stream requests may lead to longer end-to-end stream configurations when requesting streams repeatedly.
3. **Multiple stream configuration offers**: Providing multiple offers by each domain may lead to a higher flexibility for the schedule calculation for the next domains, increasing the chance of a successful stream configuration. On the other side, this mechanism temporary blocks lots of scheduling

resources. Domain internal stream requests and other MDTSN stream requests, which are parallelly processed, while waiting for active MDTSN stream configurations to finish, have a reduced chance of a successful stream configuration due to less available scheduling resources.

8 Implementation and Evaluation

We developed an MDTSN test environment to prove the viability of the MDTSN control plane mechanisms, focusing on those presented in Chapter 7. All three presented mechanisms, *maximum latency sharing*, *iterative stream requests* and m*ultiple stream configuration offers* have been implemented and tested in the MDTSN test environment.

As a basis for this Proof of Concept (PoC) implementation, single TSN domains, including TSN bridges and end-stations on the data plane as well as CUC and CNC as part of the control plane, are needed. The technical basis implementation partly implements IEEE 802.1AS, IEEE 802.1Qbv and IEEE 802.1Qcc. For the data plane, the TSN bridges TrustNodes of the vendor InnoRoute are chosen, because they already support the required standards.

For the control plane, no implementations are currently available on the market. Therefore, we extended the open source, python based SDN controller Ryu [Ryu] with a *Ryu application* to add functionality for the CUC and the CNC. The CNC supports reservation and configuration of unidirectional streams using exclusive gating i.e., each stream is assigned its own queue. The scheduling algorithm uses the ASAP approach. A GCL is generated as a YANG configuration file for TSN bridges and configured using Netconf. OpenFlow is used to match traffic to queues by the source and destination MAC address of the stream. The interface for the CUC to request and release streams by end stations is implemented as a REST API.

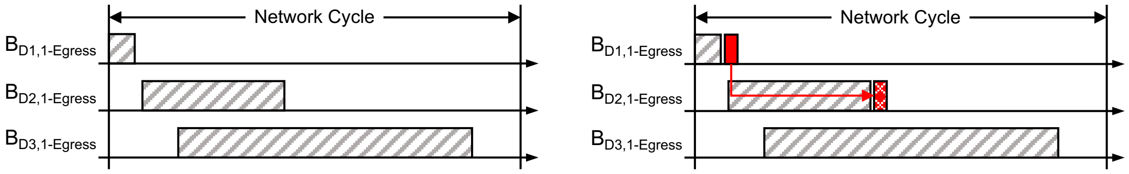
For the east-westbound communication, the E/W Manager has been implemented which communicates with other E/W Managers using a REST API with JSON encoding. It uses the python API to communicate with the CNC. A list of local devices, remote devices and their domain, directly and indirectly connected domains and their boundary network port, as well as contact information of other domain's E/W Managers is set manually in the PoC implementation.

The test environment is similar to the architecture of Figure 3, comprising three TSN domains. Because each domain at least requires an ingress network port and an egress network port on the data plane, a single TSN bridge per domain is sufficient.

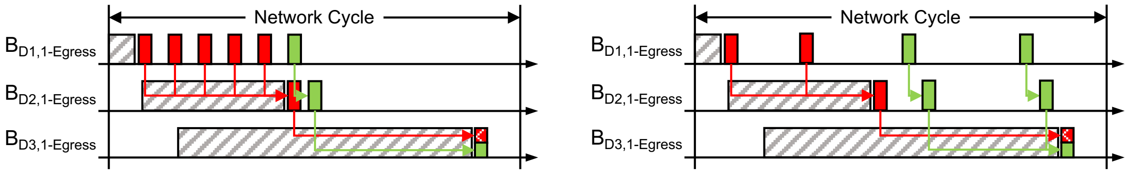
8.1 Test scenario

All three control plane mechanisms have been tested in different kinds of scenarios. One scenario has been selected and is simplified shown in Figure 6a. The figure shows the scheduled traffic of the egress network port of each TSN bridge of the MDTSN test environment. Grey boxes show already reserved time slots within the network cycle. The test scenario has been executed for each presented control plane mechanism with a network cycle of 125µs. A cyclic stream is requested with an end-to-end latency of 62,5µs, a PDU size of 100 bytes, an ETO of 0 and an LTO of 125.

Figure 6b shows the stream reservation process of the maximum latency sharing mechanism. The maximum end-to-end latency is divided by the number of remaining domains. The second domain is not able to provide a stream within the required latency. Figure 6c shows the iterative stream request mechanism. It shows that the stream is rejected by domain 3 multiple times while domain 1 is requested with an incremented ETO. In our implementation, the stream has been requested over 60 times within this scenario, until a feasible configuration has been found. Figure 6d shows the multiple stream configuration offer mechanism. We divided the range between the ETO and LTO into 4 parts (0-31,25; 31,25-62,5; …). 2 of the 4 stream offers were successful.



(a) Initial scheduling situation (b) Maximum latency sharing mechanism



(c) Iterative stream request mechanism (d) Multiple stream configuration offers

Figure 6: MDTSN stream configuration process of a stream with a requested maximum latency of 62,5µs within a network cycle of 125µs using different control plane mechanisms as presented in Chapter 7

8.2 Evaluation of IDFO Mitigation Approaches

Further the presented and tested control plane mechanisms are discussed focussing on the impact on the IDFO.

1. **Maximum latency sharing**: Within the presented test scenario, this mechanism is not able to provide a stream at all. Nevertheless, in more complex scenarios (larger network with more streams), this mechanism is able to restrict the scheduler, by requesting streams with a lower maximum latency than the requested end-to-end latency. Without the sharing, there may be domains consuming a majority of the requested end-to-end latency. In respect to the IDFO, this mechanism is not effective.
2. **Iterative stream requests**: Within the test scenario, this mechanism is able to find a feasible configuration after over 60 retries. The IDFO in domain 2 has been reduced for this stream. In more complex scenarios, this mechanism may consume lots of control plane computation resources.
3. **Multiple stream configuration offers**: Within the test scenario, this mechanism is able to find multiple stream configurations (2 of 4). The third offer had an IDFO in domain 3 which consumes most of the end-to-end latency, still resulting in a successful stream configuration. The fourth offer resulted in a small IDFO in domain 3. While offers are reserved for a certain stream, these resources are blocked and can not be used for other stream requested arriving in the process of the MDTSN stream request, which may lead other streams to unsuccessful stream configuration.

Some of the presented control plane mechanisms were able to mitigate the size of IDFOs in an MDTSN stream. In the peer-to-peer MDTSN architecture, where the stream configuration is parted into stream segments, control plane mechanism to compensate the missing global end-to-end view while preserving topology hiding, is important. Iterative stream requests and multiple stream configuration offers increase the chance of a successful stream request, while increasing the load on the control plane. The maximum latency sharing mechanism has its benefits and may be used in combination with the other mechanisms.

9 Conclusion and Future Work

This paper presented developed control plane mechanisms for MDTSN to achieve on-demand multi-domain end-to-end bounded-latency stream reservation which are required for inter-domain communication in TSN needed for various use cases. We presented an MDTSN architecture in a peer-to-peer like architecture for the fully centralized TSN configuration model. The control plane mechanisms for the integration of an east-westbound protocol can be implemented without modifying existing TSN standards while considering topology hiding. IDFOs have been identified to emerge within this distributed solution when domains configure stream segments as part of an end-to-end stream. Different control plane mechanisms to mitigate the IDFOs have been presented, discussed, and evaluated in an MDTSN test environment. These mechanisms are able to increase the stream configuration success rate, while the size of emerging IDFOs has been decreased. As a drawback, the mechanisms increase the computational load on the control plane i.e., more east-westbound communication and more traffic scheduling.

Investigating these mechanisms in more complex scenarios while comparing different scheduling approaches should be addressed in the future. Further control plane mechanisms for MDTSN should be developed to support e.g., different network cycles of TSN domains on the data plane. Control plane performance should be analyzed for use cases with frequent setup and release of MDTSN streams.

10 Acknowledgment

This work was funded by the Federal Ministry for Education and Research within the KMU-innovativ program as a part of MONAT (16KIS0782) and the Ministry for Science and Culture of Lower Saxony as a part of the research project SecuRIn (VWZN3224).

11 References

[BW21] Martin Böhm, and Diederich Wermser. Multi-domain time sensitive networks—control plane mechanisms for dynamic inter-domain stream configuration. Electronics 10, no. 20: 2477. 2021 https://doi.org/10.3390/electronics10202477

[CO16] Silviu S Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried Steiner. Scheduling real-time communication in ieee 802.1 qbv time sen-sitive networks. In Proceedings of the 24th International Conference on Real-Time Networks and Systems, pages 183–192, 2016.

[Fou16] OPC Foundation. OPC Unified Architecture Specification, Part 14: Pubsub, Draft 1.04.21. 2016.

[Fou17] OPC Foundation. PubSub TSN Configuration Broker, Whitepaper Version 0.5. 2017.

[IAUC] IEC CD / IEEE 802.1 TSN TG ballot. Use Cases IEC/IEEE 60802. https://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf, accessed on 22.09.2021.

[IEBA] Ieee standard for local and metropolitan area networks–audio video bridging (avb) systems. IEEE Std 802.1BA-2011, pages 1–45, 2011.

[IEQC] Ieee standard for local and metropolitan area networks – bridges and bridged networks – amendment 31: stream reservation protocol (srp) enhancements and performance improvements. IEEE Std 802.1Qcc-2018 (amendment to IEEE Std 802.1Q-2018 as amended by IEEE Std 802.1Qcp-2018), pages 1–208, Oct 2018.

[Ind] Industrial Internet Consortium (IIC). Time sensitive networks for flexible manufacturing testbed Characterization and mapping of converged traffic types. https://iiconsortium.org/pdf/IIC_TSN_Testbed_Char_Mapping_of_Converged_Traffic_Types_Whitepaper_20180328.pdf, accessed on 22.09.2021

[Lih] Lihao Chen. TSN Configuration Interaction. https://www.ieee802.org/1/files/public/docs2019/new-chen-TSN-Configuration-Interaction-0719-v01.pdf, accessed on 22.09.2021.

[QBV] IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks – Amendment 25: Enhancements for Scheduled Traffic. IEEE Std 802.1Qbv-2015, pages 1–57, March 2016.

[RP17] Michael Lander Raagaard, Paul Pop, Marina Gutiérrez, and Wilfried Steiner. Runtime reconfiguration of time-sensitive networking (tsn) schedules for fog computing. In 2017 IEEE Fog World Congress (FWC), pages 1–6. IEEE, 2017.

[Ryu] Ryu SDN Framework Community. Ryu sdn framework. https://ryu-sdn.org/, accessed on 22.09.2021.

[SDNi] H Yin, H Xie, T Tsou, D Lopez, P Aranda, and R Sidi. Sdni: A message exchange protocol for software defined networks (sdns) across multiple domains. IETF draft, work in progress, 2012.

[SKJ18] Sebastian Schriegel, Thomas Kobzan, and Jürgen Jasperneite. Investigation on a distributed sdn control plane architecture for heterogeneous time sensitive networks. In 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), pages 1–10. IEEE, 2018.

[TSIA] IEEE 802.1 Working Group. IEC/IEEE 60802 TSN Profile for Industrial Automation. https://1.ieee802.org/tsn/iec-ieee-60802/, accessed on 22.09.2021.

[WSJ17] Martin Wollschlaeger, Thilo Sauter, and Juergen Jasperneite. The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. IEEE industrial electronics magazine, 11(1):17– 27, 2017.

[ZC18] Yuan Zhang, Lin Cui, Wei Wang, and Yuxiang Zhang. A survey on software defined networking with multiple controllers. Journal of Network and Computer Applications, 103:101–118, 2018.