Performance analysis of cost effective multi-hop Time Sensitive Network for IEEE 802.1Qbv and IEEE 802.1Qbu standards

Asha G Hagargund1,2, Muralidhar kulkarni1 and Hariram S Satheesh3

1 Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, India.
2 Department of Electronics and Communication Engineering, B.M.S Institute of technology and management, Bangalore, India.
3 ABB Global Industries and Services Private Limited, Bangalore, India.

E-mail: ashagh@bmsit.in, mkul@nitk.edu.in, hariram.satheesh@in.abb.com

Abstract. Time-Sensitive Networking (TSN) is an emerging technology, which enables advancements in applications like industrial automation, automatic vehicle-to-vehicle communication, etc. which hosts various time-critical applications, ensuring bounded latency. The novel idea of this paper is to present OMNET++ simulation-based complex multi-hop TSN network using the native VLAN concept to bring out a cost-effective model for inter-TSN and Intra-TSN domains. This paper investigates the performance of hybrid IEEE standards, i.e., IEEE 802.1Qbu and IEEE 802.1Qbv standards. The simulation results show that the combination of these standards, when effectively scheduled in switches will reduce the latency by 3.3 µseconds in time-critical applications. Further, it is observed that in Best effort traffic, frame loss is also very less in the range of 2-5 frames out of 1385 frames. These results certainly will be of great value in more complex TSN deployments.

1. Introduction

Various Ethernet-based industrial 4.0 use cases [1], emphasize the need for efficient communication methods. The Time-Sensitive Networking (TSN) standards assert the protocols and mechanisms to scale and control the Ethernet network[2][25]. TSN finds applications in Industrial automation, Electrical power generation and distribution, Automatic vehicle-to-vehicle communication, Professional audio-video studios, Cellular radio, Multimedia communication [3]. The standard Ethernet network fails to provide the bounded latency and may result in data loss due to legacy scheduling mechanisms when both Operation Technology (OT) and Information Technology (IT) data are involved. OT data is high-priority data that involves hardware and software to monitor and control events with in an industry. IT data is noncritical enterprise-level data within industry. Hence to handle both IT and OT data IEEE has proposed various standards such as IEEE 802.1AS (gPTP, an extension of PTP protocol), IEEE 802.1Qbu (Frame pre-emption), IEEE P802.1Qci (traffic filtering and policing), IEEE 802.1CB redundant frame transmission, IEEE 802.1Qcc (configuration aspects), etc. [4][24][26]. In TSN, based on the type, data is classified into various classes and handled by algorithms specific to traffic classes. Data is classified into 8 classes, which correspond to the 8 priorities set by the
PCP value of the VLAN tag. The details of traffic classes and the priorities assigned to the class are presented in [5][6][7]. To simulate TSN topology and check the performance of the network simulation tools like OMNET’s CoRE4INET [8][9], Riverbed [10], NeStiN g[11][5][27], NS-3[12], TSNS[7], DeNS[13] are available. Few papers show the validation of simulation work with hardware setup[28]. We choose to use the NeStiNg framework of OMNET++ 5.5.1 and INET 4.1.2. We have extended the work carried in the NeStiNg simulator by considering a complex network with multiple switches and implementing Virtual Local Area Networks (VLANs) to create TSN domains.

1.1. Details on IEEE 802.1Qbu
This standard is also known as frame pre-emption mechanism and it specifies the port utilization and low latency data transmission of real-time and time-sensitive data. The pre-emption mechanism enables the switch to pre-empt the non-time critical data and transmit the high priority time- critical data when they are to be transmitted through a common port. This may introduce delay in low priority data but guarantees the bounded latency of time critical data. Each port in a TSN switch supports 8 queues and hence 8 traffic classes with different priority levels. The traffic classes are Scheduled traffic (ST), Audio Video Bridging (AVB) traffic with bounded latency and Best-Effort (BE) data with default priority. In our work, scheduled traffic with PCP value 7 is considered for which the schedule is generated offline in Gate Control List (GCL) [7][14][15][16], sensor data with the priority of 6, video data of priority 5 and the BE traffic with default priority or PCP value 0 is considered. Which has the lowest priority. Each Ethernet traffic class has a priority and is mapped to either express or preemptible interface. Traffic class mapped to express Queue cannot be pre-empted by other traffic classes. Whereas non-express or preemptible class of traffic can be preempted by high priority frames. For a frame to be pre-emptible the minimum frame size should be 102 Bytes excluding header [7]. Recent studies show that a non-pre-emptive switch introduces the delay of 2336 µseconds by a less priority frame and with pre-emption, the same network experiences the delay of 1016 µseconds per 1000 Mbps switch [7].

1.2. Details on IEEE 802.1Qbv
This standard is also called time aware shaper. It mentions the GCL which enables the highest priority control data in the industrial applications to be delivered to with bounded latency. When the arrival of periodic time-critical data is known the GCL pattern can be efficiently designed. As the nodes and bridges in the TSN network can be time-synchronized using IEEE 802.1AS [5][17][18][19][20][21], the 802.1Qbv can be effectively used for critical data delivery. We have assumed that each port of a TSN switch consists of 8 queues. Each port can be scheduled based on the traffic class. In a single port if GCL is 10000000 then the gate corresponding to Queue 7 (when Queue numbers are 0-7) is open and gates corresponding to 0-6 are closed. Hence for the entire duration of gating where GCL has this pattern the frame of queue 7 can only be sent through the switch and all the remaining data frames are halted in the switch. GCL changes periodically and this period is called the gating cycle. The same cycle repeats throughout the simulation duration. The gating mechanism ensures that only one queue gets access to the egress(exit) port. However, the transmission of non-ST data will stop before the time slot of ST data. This time slot or window is known as a protected window. The switch internals for GCL:10000000 is as shown in figure 1.

The time slot where non-time-critical data frames are allowed through the egress(exit) port is called the unprotected window and also an additional time slot or window during which no frame transmission is allowed and the egress port is in idle state is known as a guard window. The details on all these time window slots and switch gating cycle is shown in figure 2.
2. Literature review

Bello[4], this paper provides a detailed introduction of TSN standards which are highly relevant to industrial communication, and also provides the future direction of research. According to the author future scope for research are as follows: While looking at the configuration TSN network, scaling the network schedules with fault tolerance is to be addressed, there is a need for improvement in deploying the applications. Concerning the security of TSN applications, there is a need of detecting the threats early and proposing mitigation strategies. Simulations, Prototypes, Performance studies, use case demonstrations are highly required for the TSN solutions maturity.

Haris Suljic[5], does the performance analysis of TSN by considering IEEE 802.1Qbv, IEEE 802.1Qu, and Credit-based shaper. The simulation was carried out in OMNET++ 5.6.1 and INET 4.2.1 with the NeSTiNg framework. The author has considered a simulation duration of 3 minutes and a simulation cycle of 500 µseconds. Protected window of 17 µseconds and unprotected window of 360 µseconds and guard window of 123 µseconds. Parameters considered for simulation are end-to-end delay and link efficiency. 5 scenarios are considered for performance analysis. The first one is IEEE 802.1Qbv, here the BE traffic experience the packet delay and time-critical data has deterministic data delivery. The Second one is IEEE 802.1Qbu and it is used along with credit-based shapers. The conclusion of this simulation is Audio-video traffic is delayed by credit-based shaper, so frame pre-emption does not reduce the latency of frames with higher priority. The third scenario is using both frame preemption and time aware shaper. The Combination of these 2 reduces the delay of BE traffic whereas delay of deterministic data transmission is unchanged because of using proper gating mechanism. In scenario 4, Gating, Pre-emption, and Credit- based shapers are used. The combined result is frame pre-emption provides reduced end-to-end latency for event-triggered data and BE data experiences more delay compared to the previous scenarios. Scenario 5 is about an industrial in-car network with 14 end nodes and 2 switches. In this scenario frame pre-emption and gating mechanisms are
used. Gating results in an unaltered delay in the data delivery and pre-emption will result in increased delay for BE traffic. The conclusion of this experiment is to look at efficiently and dynamically designing the GCL, also look at various combinations of these standards with various configurations to check the performance of the network in a more efficient way.

In [7], the aspects of Pre-emption are discussed in detail. The performance of the TSN network when there are multiple express data (The data that pre-empts less-priority data) is explained. The author develops a new simulation tool known as TSNS. This tool is used for simulation of the TSN switch behavior according to the IEEE 802.1Q amendments. The environment used is Microsoft’s visual studio and code is developed using C language.

In [10], the author develops a tool to check how TSN configuration can be automated. The tool was developed to make use of OMNET’s NeSTiNg platform. Researchers can use this extension to automate the configurations, extract the flows and check the performance of the network by using the GUI which is provided by the author. This helps us save time in configuring the parameters through XML files.

Time-aware shaper schedules the frames according to the predefined GCL list. In [15] Ramon Serna Oliver, et al, proposes a method of synthesizing the GCL based on an SMT solver. Also discussion on Time aware shaper, Pre-emption, and Credit-based shaper is done [15]. But, still, there is a need to do more analysis by taking various case studies into account. We need to look at trade-offs between various performance parameters and efficiently design the system looking at particular applications. Another motivation behind this paper is to explore various options that are part of the OMNET++ simulation tool for Time-Sensitive Network simulation and look at the options available, appreciate and also look for various improvements in the existing NeSTiNg simulation framework.

Escola[22], The objective of this work is to implement and validate the designed system that allows automatically configuring the TAS’s scheduling parameters of the switches of a TSN network. For configuring TAS, IEEE 802.1Qcc proposes NETCONF, RESTCONF, and SNMP from which the author has chosen NETCONF. NETCONF enables us to install, delete and manipulate the configuration of network devices. NETCONF is a network management protocol, but it does not specify how to model the configuration. To tackle this issue, the TSN standards suggest using YANG. YANG is a data modeling language which is used by the NETCONF protocol. YANG is described in the RFC 6020.

Anna arestova[29], In this paper the analysis of Frame preemption(FP) and Time aware shaper(TAS) is carried out and it is compared with Strict priority scheduling(ST) by NeSTiNg framework of OMNET++ simulator. For simulation, a multi-hop, tree topology with multiple star configurations are considered which is comparable to an industrial automation network. The result shows that the TAS produces a large configuration overhead but performs the best and it is suitable for ultra-low latency requirements. FP can be combined with TAS for the efficient band usage. ST shows the least configuration overhead and is suitable for real-time applications with fewer millisecond cycles. The transmission offsets can be effectively scheduled to reduce the interference in the egress queue of a switch when ST and FP standards are used.

3. Proposed Work

3.1. What is the objective of the proposed work?
The objective of this work is to simulate a complex multi-hop TSN network and implement TSN domains using VLAN concepts. Also check the performance of inter-TSN and intra-TSN communication for IEEE 802.1Qbu and IEEE 802.1Qbv when they are applied together.

3.2. What is the need of TSN domains?
Many upcoming industrial automation use cases require TSN domains. A TSN domain is a collection of devices, their ports, and their LANs which use TSN standards to transmit time-
sensitive data [1]. In the proposed work, TSN domains (VLANs) are created based on the functionality of the nodes. Here TSN domain and VLAN are interchangeably used.

3.3. Specification of software tools used for simulation.
Details of the software versions used
i. Operating System - Ubuntu 18.0
ii. OMNET++ - 5.5.1
iii. INET - 4.1.2
iv. NeSTiNg - First version available from omnetpp.org

3.4. What is the need for choosing topology as in figure 5 and figure 6?
We have assumed a real-time scenario where a factory in which the following floors are present.
i. Ground floor: Contains switch "A" and 4 nodes.
ii. First floor: Contains switch "B" and 4 nodes.
iii. Second floor: Contains switch "C" and 4 nodes.
iv. Third floor: Contains switch "D" and 4 nodes.

If on each floor there is a device that belongs to a specific TSN domain then we should go for topology mentioned in figure 5 for intra-TSN and figure 6 inter-TSN communication.
The proposed inter-TSN and intra-TSN communication network are as depicted in figure 3 and figure 4 respectively. In the block diagram the color codes are used for identifying the domains. The devices with same color belong to one domain. We see such '4' domains in figure 3 and '5' domains in figure 4.

As mentioned in the literature survey, according to [5], there is a need to have further look into the configuration aspects of TSN protocols for more efficient data transmission. According to [4] lot of simulations, prototypes verification are required in TSN because many applications require improved versions of existing standards. Hence, we have checked how the combination of both these protocols will increase the efficiency of the communication instead of using them separately.

![Figure 3. Block diagram of Intra-TSN domain.](image-url)

3.5. Intra-TSN domain/Intra-VLAN communication
Figure 3 shows the block diagram of the intra-TSN domain network which has 4 TSN-enabled switches and each switch has 4 devices physically connected to them. Each device generates a specific type of traffic and we have scheduled the switch in such a way that each traffic class is handled effectively in the presence of TSN protocols such as Pre-emption and Gating. The configuration of switches, their ports, type of schedule, types of the algorithms used is mentioned in tables 2 and 3.
3.5.1. Details of TSN domains.

i. TSN Domain 1 is implemented by VLAN 20 which has a robotController that generates time-critical periodic data and broadcast the data to only robot1, robot2 and robot3.

ii. TSN domain 2 is implemented by VLAN 30 and includes a camera which generates a non-periodic data with priority 5 and broadcasts the data to CPD1(Camera Processing Device 1), CPD2, and CPD3.

iii. TSN domain 3 is implemented by VLAN 40 which includes the sensor as the source node that produces nonperiodic data with priority 6. The sensor data is broadcast to only the nodes in that specific VLAN, i.e. to actuator1, actuator2, and actuator3.

iv. TSN domain 4 is implemented by VLAN 60 and BETG as the BE Traffic Generator and broadcasts the data only to BETR1(BE Traffic Receiver), BETR2 and BETR3.

3.5.2. Details on traffic generated by all the nodes and frame sizes.

The robotController transmits minimum sized, network control and periodic traffic hence a frame size of 102 Bytes is chosen. This is the minimum size requirement for frame which is preemptable[7]. Similarly a maximum frame size of 1500 bytes is chosen for BETG and Camera which are similar to Best effort traffic. As the sensors used in real time applications generate the traffic lesser than BE traffic and greater than control traffic hence the sensor traffic chosen is 500 Bytes. Number of robotController frames considered during the simulation period is 138 frames. The number of frames can be increased by increasing the simulation duration. The number of frames sent and received by all the nodes are depicted in Figure 9 and Figure 10.

In the simulation process, the scheduling configuration for switches is written in an XML file and all the VLAN configurations are done in the .ini file of the NeSTiNg framework. The types of data generated by the nodes in the network, size of each data, periodicity, and their priorities are mentioned in table 2.

3.6. Inter-TSN domain/Inter VLAN communication

A new node named as newdevice is now added to Switch D. This belongs to a new VLAN named as VLAN 50.

In the inter VLAN communication we have assumed that there is a communication from robotController that belongs to VLAN20 to newnode that belongs to VLAN 50. Generally when communication between 2 VLANs is implemented by using an L3 router. But, here we
use the native VLAN concept with the trunk port. This helps us avoid a dedicated router to have communication between 2 VLANS.

Figure 7 and figure 8 explain the flowchart of data flow when the native VLAN concept is implemented for inter-TSN domain communication.

We have explicitly tagged the native VLAN with a specific ID in which our desired TSN node is present.

Example: Inter TSN domain traffic L2 routing from VLAN 20 to VLAN 50

Assume that a device is connected to switchD.eth[6] and it is mapped to VLAN 50. The egress traffic of switchC.eth[6] port with VLAN tag 20 will be mapped to VLAN tag 50. This is a native VLAN for this switch-D. Hence the Node connected to switchD.eth[6] gets the TSN traffic which demonstrates inter-TSN domain communication.

| TSN Domain | Corresponding Nodes                        |
|------------|--------------------------------------------|
| TSN domain1| robotController, robot1, robot2, robot3    |
| TSN domain2| sensor, actuator11, actuator2, actuator3   |
| TSN domain3| camera, CPD1, CPD2, CPD3                   |
| TSN domain4| BETG, BETR1, BETR2, BETR3                 |
| TSN domain5| newdevice                                  |

Table 2. Traffic generated by all classes of nodes.

| Node          | Frame size | Periodic or Not? | PCP value |
|---------------|------------|------------------|-----------|
| robotController| 102 Bytes  | Periodic         | 7         |
| sensor        | 500 Bytes  | Nonperiodic      | 6         |
| camera        | 1500 Bytes | Nonperiodic      | 5         |
| BETG          | 1500 Bytes | Nonperiodic      | 4         |
Figure 7. Flowchart demonstrating native VLAN routing for inter-TSN domain.

Figure 8. Reference for ports of Switch C and Switch D.

Table 3. Switch parameters used while simulation.

| Parameter                  | Value                       |
|---------------------------|-----------------------------|
| Switch processing delay   | 5 \(\mu\)second             |
| Queue buffer capacity     | 363,360b                    |
| Ethernet cable parameters| delay = 0.1 \(\mu\)second, datarate = 1Gbps |
| TSA algorithm             | Strict priority             |
| Queue 0 - 6 of all switches| Express Queue = False       |
| Queue 7 of all switches   | Express Queue = True        |

4. Results

According to figure 3 where the communication is within the VLAN, i.e. robotController to robot1, robot2, robot3. Or Sensor to Actuator1, Actuator2, Actuator3, and so on pre-emption along with Gating is applied. 17 \(\mu\)seconds protected window, 360 \(\mu\)seconds unprotected window and, 123 \(\mu\)seconds guard window is considered in one switch cycle. The latency in transmitting time-critical data is constant throughout the simulation time as shown in figure 11. This is possible because, in an entire switching cycle, the robotController transmits the data at a fixed time, and also it is periodic. By knowing this factor GCL vector can be efficiently designed to maintain constant delay in the case of robotController. Looking at the latency charts the
deterministic data delivery is guaranteed for the destinations present after multiple hops also. The latency in transmitting sensor data to all its nodes varying with simulation time as shown in figure 12. This variation in the latency is due to the buffer capacity of queues in switch as well as other high-priority data transmission. Latency for transmitting camera data and BETG data to all the corresponding nodes is periodic as shown in figures 13 and 14 respectively.

The graph of frames sent Vs received by the source nodes to destination nodes of single-hop is shown in figures 9 and 10 for both cases. The robot1, camera1, and sensor1 nodes have 0 frame loss whereas for BETR1 loss is only 2/1385 frames.

In the Inter-TSN network, the high priority robotController data has the constant latency of 13.42 µseconds for the first hop(to robot1), 19.56 µseconds for the second hop(to robot2) and 25.7 µseconds for the third hop(to robot3). This is similar to the latency of the Intra-TSN network. The latency chart for all the remaining source nodes to their corresponding destination nodes in the Inter-TSN network follows the same periodic pattern as intra-TSN latency charts with small variations in the values.

Figure 9. Frames sent Vs received in single hop intra-TSN communication.

Figure 10. Frames sent Vs received in single hop inter-TSN communication.

Figure 11. Simulation time vs latency from robotController to all its destinations.

Figure 12. Simulation time vs latency from sensor to all its destinations.
5. Conclusion and future work

In this work, the hybrid IEEE standards 802.1Qbv and 802.1Qbu are used for scheduling the traffic in the switch. By this configuration, we can guarantee deterministic data delivery of time-critical applications and make sure that non-periodic BE traffic will have very little frame loss of 2-5 frames out of 1385 frames. This paper implements a new configuration of TSN domains and also a concept of native VLAN whenever Inter-TSN domain communication is required. This avoids the usage of a dedicated router interconnecting multiple TSN domains. Hence the approach we followed here is a cost-effective solution. The advantage of making multiple broadcast domains in a single network significantly reduces CPU cycles of switches. This avoids unnecessary broadcast of the traffic which is not destined to the nodes. As future work, the hardware validation of complex networks is necessary along with IEEE 802.1AS(Time synchronization) standard. Also proposing new scheduling algorithms to have efficient communication between time-sensitive nodes is required.

References

[1] https://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf
[2] https://1.ieee802.org/tsn/iec-ieee-60802
[3] N. Finn, "Introduction to Time-Sensitive Networking," 2018 in IEEE Communications Standards Magazine, vol. 2, no. 2, pp. 22-28, JUNE 2018, doi: 10.1109/MCOMSTD.2018.1700076.3
[4] L. Lo Bello and W. Steiner, "A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems,2019 " in Proc. of the IEEE, vol. 107, no. 6, pp. 1094-1120, June 2019, doi: 10.1109/JPROC.2019.2905334.
[5] Haris Suljic, Mia Muminovi "Performance study and analysis of time sensitive networking" Thesis for the Degree of Master of Science in Computer Science -Embedded Systems, Malardalen University School of Innovation Design and Engineering Vasteras, Sweden, June 14, 2019.
[6] Ieee standard for local and metropolitan area networks–bridges and bridged networks,"IEEEStd 802.1Q-2014 (Revision of IEEE Std 802.1Q-2011), pp. 1–1832, 2014
[7] Lejla Murselovi "Performance analysis of the preemption mechanism in TSN", Thesis for the Degree of Master of Science in Computer Science, Malardalen University School of Innovation Design and Engineering Vasteras, Sweden, June 9, 2020.
[8] https://omnetpp.org/download-items/Core4INET.html
[9] J. Jiang, Y. Li, S. H. Hong, A. Xu and K. Wang, “A Time-sensitive Networking (TSN) Simulation Model Based on OMNET++,”2018 IEEE Int. Conf. on Mechatronics and Automation (ICMA), 2018, pp. 643-648, doi: 10.1109/ICMA.2018.8484302.
[10] M. Pahlevan, J. Schmeck and R. Obermaisser, "Evaluation of TSN Dynamic Configuration Model for

Figure 13. Simulation time vs latency from camera to all its destinations.

Figure 14. Simulation time vs latency from BETG to all its destinations.
Safety-Critical Applications," 2019 IEEE Intl. Conf. on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom), Xiamen, China, 2019, pp. 566-571, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00086
[11] Falk, Jonathan, et al. "NeSTiNg: Simulating IEEE time-sensitive networking (TSN) in OMNeT++." 2019 Int. Conf. on Networked Systems (NetSys). IEEE.
[12] T. Wan and P. Ashwood-Smith, "A Performance Study of CPRI over Ethernet with IEEE 802.1Qbu and 802.1Qbv Enhancements," 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, 2015, pp. 1-6, doi: 10.1109/GLOCOM.2015.741759
[13] Addanki, Vamsi, and Luigi Iannone. "Moving a step forward in the quest for Deterministic Networks (DetNet)." 2020 IFIP Networking Conf. (Networking). IEEE.
[14] P. Pop, M. L. Raagaard, S. S. Craciunas and W. Steiner, "Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks," in IET Cyber-Physical Systems: Theory Applications, vol. 1, no. 1, pp. 86-94, 12 2016, doi: 10.1049/iet-cps.2016.0021.
[15] R. Serna Oliver, S. S. Craciunas and W. Steiner, "IEEE 802.1Qbv Gate Control List Synthesis Using Array Theory Encoding, 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Porto, 2018, pp. 13-24, doi: 10.1109/RTAS.2018.00008.
[16] A. M. Kentis, M. S. Berger and J. Soler, "Effects of port congestion in the gate control list scheduling of time sensitive networks," 2017 8th Int. Conf. on the Network of the Future (NOF), London, pp. 138-140, doi: 10.1109/NOF.2017.8251236.
[17] G. M. Garner, A. Gelter and M. J. Teener, "New simulation and test results for IEEE 802.1AS timing performance," 2009 Int. Symp. on Precision Clock Synchronization for Measurement, Control and Communication, Brescia, pp. 1-7, doi: 10.1109/ISPCS.2009.5340214.
[18] D. Shrestha, Z. Pang and D. Dzung, "Precise Clock Synchronization in High Performance Wireless Communication for Time Sensitive Networking," 2018 in IEEE Access, vol. 6, pp. 8944-8953, doi: 10.1109/ACCESS.2018.2805378.
[19] M. D. Johas Teener and G. M. Garner, "Overview and timing performance of IEEE 802.1AS," 2008 IEEE International Symp. on Precision Clock Synchronization for Measurement, Control and Communication, Ann Arbor, MI, pp. 49-53, doi: 10.1109/ISPCS.2008.4659212.
[20] M. D. Johas Teener and G. M. Garner, "Overview and timing performance of IEEE 802.1AS," 2008 IEEE Int. Symp. on Precision Clock Synchronization for Measurement, Control and Communication, Ann Arbor, MI, pp. 49-53, doi: 10.1109/ISPCS.2008.4659212.
[21] M. Pahlevan, B. Balakrishna and R. Obermaisser, "Simulation Framework for Clock Synchronization in Time Sensitive Networking," 2019 25th IEEE Int. Symp. on Real-Time Distributed Computing (ISORC), Valencia, Spain, 2019, pp. 213-220, doi: 10.1109/ISORC.2019.00046.
[22] Albert Bergstrom "Automatic generation of network configuration in simulated time sensitive networking(TSN) applications", Thesis for the Degree of Master of Science in Computer Science with Specialization in Embedded Systems, Malardalen University School of Innovation Design and Engineering Vasteras, Sweden. June 9, 2020.
[23] https://techhub.hpe.com/eginfolib/networking/docs/switches/K-KA-KB/15-18/atmg/content/ch01s27.html
[24] C. von Arnim, M. Dragan, F. Frick, A. Lechler, O. Riedel and A. Verl, "TSN-based Converged Industrial Networks: Evolutionary Steps and Migration Paths," 2020 25th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA), pp. 294-301, doi: 10.1109/ETFA46521.2020.9212057.
[25] https://www.ni.com/en-in/innovations/white-papers/18/time-sensitive-networking-tns-frequently-asked-questions.html
[26] Zhao, Lin, et al. "Comparison of Time Sensitive Networking (TSN) and TTEthernet." 2018 IEEE/IAIA 37th Digital Avionics Systems Conference (DASC). IEEE.
[27] A. C. T. d. Santos, B. Schneider and V. Nigam, "TSNSCHED: Automated Schedule Generation for Time Sensitive Networking," 2019 Formal Methods in Computer Aided Design (FMCAD), pp. 69-77, doi: 10.23919/FMCAD.2019.8894249.
[28] Jiang, Junhui, et al. "A Simulation Model for Time-sensitive Networking (TSN) with Experimental Validation." 2019 24th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA). IEEE, 2019.
[29] A. Arestova, K. -S. Jens Hielscher and R. German, "Simulative Evaluation of the TSN Mechanisms Time-Aware Shaper and Frame Preemption and Their Suitability for Industrial Use Cases," 2021 IFIP Networking Conference (IFIP Networking), 2021, pp. 1-6, doi: 10.23919/IFIPNetworking52078.2021.9472830.