Time Sensitive Networking - 802.1Qci

Abdalla Mahamid, Filip Rezabek and Kilian Holzinger*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: abedkh.mahamid@tum.de, rezabek@net.in.tum.de, holzinger@net.in.tum.de

***Abstract*—Time sensitive Networking (TSN) is a task of the IEEE 802.1 group. It provides a real-time communication and a large bandwidth to transfer big amount of data in time that fulfills the TSN communication requirement of end-devices. This technology plays an important role in several industries, for instance, IIoT (industrial internet of things), automotive and self-driving cars, that have to transfer a big amount of data to the targets that must be collaborating simultaneously [1] [2]. There are different standards which can be used to realize this. The main focus of this paper will be about the IEEE 802.1Qci Standard with its definition Per-stream filtering and policing (PSFP). PSFP consists of three main instance tables, first, Stream filter instance table, second, Stream gate instance table and finally, Flow meter instance table (see Figure 1). These tables have a relationship between each other that realize the functionality of PSFP. Furthermore, there are some other applications that their integration with PSFP give solutions for end-device TSN-requirements, for instance, IEEE 802.1Qbv, IEEE 802.1Qav (Section 3.3) in addition to centralized configuration module (Section 3.2).**

***Index Terms*—TSN, Time-sensitive Networking, IEEE 802.1, IEEE 802.1Qci, real-time communication, implementation of TSN.**

1. Introduction

Self-driving cars are a part of the future technology that requires high efficiency, secured systems with real-time communication [1]. This technology must avoid the high latency communication to transport multiple data flows in addition to the sensitivity of packet loss. Time-sensitive Networking with its definitions to the Standards IEEE 802.1Qci, IEEE 802.1Qva and IEEE 802.1Qbv are technologies that can provide the simultaneous, safe and secure transfer of data as well as combining time synchronization and transmission scheduling due to the end-points requirements [3]. It is also an important technology to the industry likewise Automotive, to enable optimization of communication in addition to reducing the cost in general [4].

The Standard IEEE 802.1Qci is supplying the per-stream filtering and policing (PSFP), which is a task of the Time-sensitive Networking group IEEE 802.1. Why is TSN important? TSN gives various of benefits for the industry, for instance, large bandwidth, security, interoperability and low latency and synchronization [2]. Technology like automotive, industrial internet of things or self-driving cars have a large amount of data has to be transferred from one point to another point with low latency and secure transfer because they are very sensitive data [2]. The current ethernet technology that the industry used to use is IEEE 802.3u that limited with 100 Mbit/s of bandwidth and half-duplex communication [2]. The TSN will give a solution that provides a high bandwidth transfer and communications [2]. Furthermore, TSN provides security technology that gives the framework higher level of defense, protection and performance. For the interoperability, it uses existing Standards and integrates them with new applications to satisfy the TSN-requirements, in other words, it is no need to develop everything from zero, TSN application can use existing technologies and improve them to fulfill the TSN-requirements [2].

Finally, TSN has various advantages against the current and common Ethernet Standards (802.3) to reduce the latency as well as enabling the synchronization between End-devices. TSN can transfer data taking into consideration the priority and time-requirements while the current Ethernet does not differentiate between critical data (data with TSN requirements) and normal data (without TSN requirements) . Later in this paper it is explained exactly how it can be secured, and which methods are used in order to fulfill the TSN requirements. To sum up, the essential goals of the TSN are the low latency communication (real-time), security and priority for critical flows.

Self-driving cars technology it is an important future technology that needs very secure, efficient, and low latency communication in addition to protection against Denial-of-Service (DoS) attacks [1]. IEEE Standards could not decide if the received data flows are urgent or not. Therefore, new technologies were developed to solve these problems. IEEE 802.1Qci is a TSN substandard of IEEE 802.1Q which is TSN substandard of IEEE 802.1 that provide the per-stream filtering and policing which is given solutions to particular problems. Per-stream filtering and policing filter and scheduled the ingress flows due to discarding the non-essential streams and scheduled high priority streams first. This paper points the workwise of per-stream filtering and policing and how it guarantees secured and low latency communication.

There are 4 sections in this paper. Section 2 explaines and gives a related work, how the Per-stream filtering and policing works. The 3ed Section shows several implementations of 802.1Qci, in addition to definitions for important concepts that will be used in the rest of the paper. Finally, in the last Section, Section 4, the author's conclusion will be presented.

doi: 10.2313/NET-2021-05-1_03

2. Background and Related Work

IEEE 802.1Q Standard technology of layer 2 that make decisions using Ethernet-Headers and not IP-Headers. The current used IEEE Ethernet Standards do not have layer 2 deterministic capability, thus, the intention of IEEE 802.1Q was to supply deterministic flows on standard Ethernet. The traditional IEEE Ethernet Standards give no attention for the sensitive information and their priorities, in addition to no ensure for safety, security or protection in a network [3]. IEEE 802.1Q managed and delivered flows to minimize the transmission time for real-time applications using scheduling and policing to fulfil deferent requirements of deferent applications. A Sub-Standard of IEEE 802.1Q, IEEE 802.1Qci with their definition for "per-stream filtering and policing" is a Sub-Standard that with other Standards/Sub-Standards meets the critical requirements that can realize transmission of frames in a particular and predictable time due to filtering the sensitive information and queuing them taking into consideration their priorities [4].

2.1. Per-Stream Filtering and Policing (PSFP)

IEEE 802.1 Qci defines Per-stream filtering and policing (PSFP) that consist of three instance tables, Stream Filters, Stream Gates and Flow Meters. The relationships between the tables can be seen in Figure 1.

To start with, stream filter instance table. It consists of several components that help determine which frame should be processed, for example a Stream Filter Instance Identifier is an integer value that works as an ID for this stream and its index in this table [5]. Then, Stream Gate Instance Table. It is an instance that contains parameters for each flow. For instance, a stream gate instance identifier, that nearly the same functionally of above example without the "index as position". Stream gate state has two states, OPEN and CLOSE, that determine which flow is permitted to pass through the gate [5]. Lastly, internal priority value specification, that have also two options, the null value and an internal priority value. Each have different functionality with the same goal to determine frame's traffic class [5].

Finally, Flow Meter Instance Table. It is as the Stream Gate Instance Table contains parameters for each flow, that gives them a specification. An important specification called Bandwidth Profile Parameter and Algorithm. (For more information see [5]).

To sum up how it exactly works, there is an example. The Figure 1 shows, there is a flow ingress (input) that passes to the *Stream Filter Instance Table* where it have several flows. Each flow has multiple attributes, Stream (flow) ID, Priority, Gate ID and Meter ID. The stream filter table with its application can identify each flow from its unique ID, knows its priority level and which Gate it should be passed to. After taking into consideration the priority, a flow will be forwarded to *Stream Gates Table* which have different gates. Each Gate is specified with a unique Gate ID. With a specific application that provides the ability to match each flow with its correct Gate ID the flow is passing to the matching Gate. Each gate in the

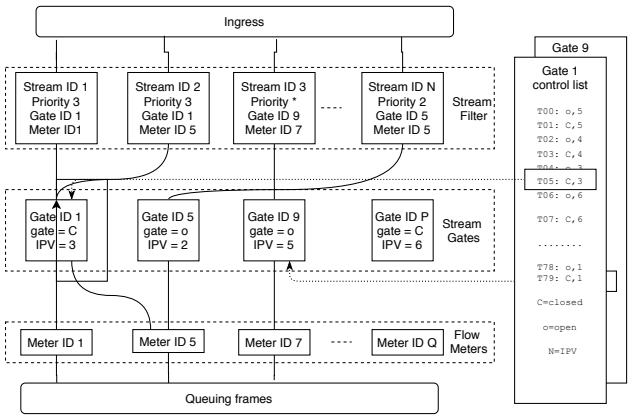

Figure 1: Per-stream filtering and policing [5]

Stream Gate Table has attributes, for instance, Gate ID, gate, internal priority value (IPV) etc. Gate ID is a unique ID that the Stream Filter Table use to match its flow with the correct Gate. The attribute gate is like a status for a gate, with two status, OPEN (o) and CLOSE (c). There is also a *Gate Control List* that is important to control the status of a gate and to update it.

Further, IPV gives the priority of a flow that in this gate. The flow is in the Stream Gate Table and to forward it, the gate controls the status to know if is allowed to let the flow pass or not. If the gate status is OPEN, then the flow allowed to pass to the next table, but if the status is CLOSE then it is not allowed to pass. When the flow is passed, each flow in the Flow Meter Table has also attributes (parameters) that are as specified in *Bandwidth Profile Parameters and Algorithm*. Then the flow will be passed to queue it according to its attributes and priorities. The algorithm that used to schedule the frames is similar to the schedule method of IEEE 802.1Qbv. (more info [5]).

3. Implementations

Before we dive into the details, here some important terms that will make the rest of the explanation more understandable.

3.1. Definitions of Important Components

- **TSN flow**: describes the time-sensitive communication between end devices. Each stream (flow) has a different time requirement that it gives no concessions for its right in strict transmission time [4].
- **End devices**: End devices are the hosts or the source and the destination nodes in our network that the TSN flows weillcbe transmitted between them. Each of these devices has to run an application that requires deterministic communication [4].
- **Bridges**: or "Ethernet Switches" are special switches that capable to transfer or receive frames a TSN flow taking into consideration their schedule and priorities. In other words, the TSN switches should have the ability to forward the

doi: 10.2313/NET-2021-05-1_03

frames on a schedule and receive frames according to a schedule [4].

- **Central Network Controller (CNC)**: It can be defined as a proxy for the Network and the Control Application, where they are needing deterministic communication. The TSN frames are simply transmitted on the schedule defined by the CNC. In other words, it is an application, which provides configuration frames for TSN bridges. These frames are response to TNS stream requirements that received from the Centralized User Configuration (CUC). This application gives it the vendor of the TSN bridges [4].
- **Centralized User Configuration (CUC)**: CUC is an application that communicates with the Central Network Controller (CNC) and End-devices. CUC makes requests to CNC for TNS flows, where each flow has deference requirements. In other words, CUC is an application that receives requests from End-devices and then CUC will transfer the configuration flows to the CNC in the Network for processing [4].

3.2. Security Policies

IEEE 802.1Qci avoids traffic overload condition, that impact the bridges and the end-devices on a network, that is mean is improving the robustness of a network, for instance, daniel-of-Service (DoS) attack, error through streams transmission or likewise if we receive a flow that is not in the schedule time period then it is dropped [6].

After a while had IEEE 802.1Qci a progress. The source states that little progress has been made to connect the standard with existing industrial security systems and architectures [7].
IEEE 802.1Qci Standard is relatively new addition to IEEE 802.1Q Standard, The source states that 802.1Qci does not define how specific policies are created and deployed. Furthermore, there are not a lot of contribution to give explicit ways how the security systems can be deployed and employed, or how the filtering rules are dynamically updated as the Internet and Networks scaling up [7]. TSN networks apply the centralized configuration module, Central User Configuration (CUC) and Central Network Controller (CNC) [7].

How these Components work together? The CNC is a software program works on a costumer own network or premises that communicate with the bridges (a network's component) and controls it. CNC has two principle responsibilities, First, it resolves routes and scheduling TSN flows, second, it configures the bridges for TSN operations. The CNC communicates with the CUC to receive the communications requirements that a network must provide, then the CNC processes all the communications requirements to determine the routes and to schedule the end-to-end transmission for each TNS flow. CNC provides a unique identifier for each flow (include MAC-Address) to help the bridges without doubt identify each flow. Finally, CNC transfers all these processed data flows to bridges the premise [4].

The progress that happens is to integrate the 802.1Qci Standard security system with the Centralized Configurations module (CCM). The idea is to centralize the policies

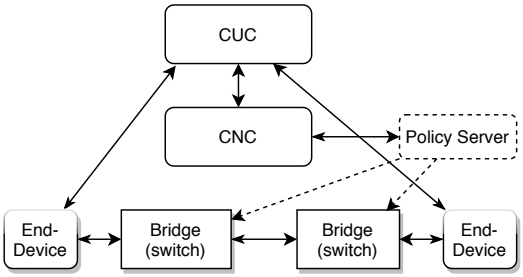

Figure 2: Centralized Configurations module (CCM) [7]

in a policy server to collect the global security policies for the network to dynamically and automatically update them due to the requirements. In the following points it will be explained how the 802.1Qci Standard is integrated with the CCM.

The Policy Server communicates with the CNC.Iin addition, the identifications of TSN endpoints should be imported as objects, besides, the route of the flow should be in a secured mode. Then the Policy Server should be provided on configurated global security policies that are required to be integrable to integrate with the configured IEEE 802.1Qci policies and to enable the end-devices to deal with them. The bridges should be provided with the security system and the policies. For this reason, the Policy Server should enable to update the security system beside to the policies when the requirement for a route change from CNC that received new information and requirements from the CUC. At the same time the control mechanism Quality of Service (QoS) is also should be updated when the bridges receive new flows. The CNC is allowed to deal with per-flow shaper but is not allowed to handle with QoS. Here come the Policy Server to deal with QoS and to adjust the QoS configuration taking into consideration its buffer size and the current number of on-board TSN flows without any meddling from human resource. In general, the Policy Server communicate with the bridges on a network as well as with a CNC on the same premise that configurate the communication requirements for each flow. [7]

3.3. Queuing Frames

After a flow passes all the levels from the ingress till the flow meter, it gets queued taking into consideration all the communication requirements that the End-devices required. To be queued there is an important component that manage this process called Credit Based Meter (CBM). To explain CBM the Credit Based Shaper (CBS) should be referred.

CBS is defined in IEEE 802.1Qav that have two important concepts, $idleslope$ and $sendslope$. $idleslope$ is a definition of the reserved Bandwidth and $sendslope$ is a definition of the Bandwidth subtracted from reserved Bandwidth. CBS has an important role in the TSN network to keep the maintain of a reserved bandwidth [1]. If two End-devices want to communicate, for security reasons the route should be protected and reserved for the sent flow. This reservation takes part from the bandwidth (or the whole, depend on the communication requirements). To control this reservation, it should be controller

doi: 10.2313/NET-2021-05-1_03

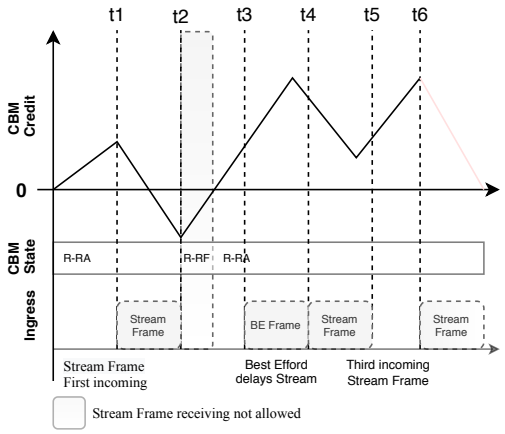

Figure 3: Credit Based Metering example [1]

that manages all the traffic flows. CBS is the method that has been defined to solve this problem. It shapes the traffic flows depending on the reservation Bandwidth that gives information about the maximum interval and size of flow frames to that allow to enter the network medium [1].

CBM is based on the credit value of the CBS and others. It takes the two Slopes, $idleslope$ and $sendslope$. It also contains further parameters, maximum burst size parameter ($Burst_{max}$), stream frame sending duration ($T_{duration}$), frame size (FS_{stream}) port bandwidth (B), Ethernet inter frame gap (T_{ifg}) and the maximum credit value ($Credit_{max}$). $Burst_{max}$ is a parameter that gives the number of the allowed streams frame to income burst. $T_{duration}$ defined as following [1]:

$$T_{duration} = \frac{FS_{stream}}{B} + T_{ifg}$$

$T_{duration}$ is important to calculate the $Credit_{max}$ that defined as following [1]:

$$Credit_{max} = sendslop \cdot T_{duration} \cdot (Burst_{max} - 1)$$

In addition to all the definitions, CBM can be in two different statuses R-RA and R-RF.
R-RA = RUNNING RECEIVING ALLOWED
R-RF = RUNNING RECEIVING FORBIDDEN

As expected, the status at the beginning is R-RA with credit 0 that will be changed depending on $idleslope$. The credit still increasing till a flow frame is incoming or the credit hit the maximum then stopped. If it stopped because of $credit_{max}$ reached, then the credit value stays as it is till a frame is incoming . As the credit in R-RA can increase, also it can decrease. $sendslope$ is responsible for decreasing the credit for receiving duration of a flow. Also, if it decreased and still has positive value, still acting normal like before, but if it crosses the zero to the negative value, then immediately the status will be changed to R-RF. In the status R-RF no frames from now allowed to be queued and will be dropped. This will be changed if the credit again increases to positive value to change it to R-RA. This will happen by $idleslope$ if an incoming frame was dropped [1].

4. Future Work

TSN is the future to realize synchronization and simultaneous communication to enable future ideas as self-driving cars the possibility to become true. TSN will be improving and the standards will be implementing in applications. Credit Based Meter (CBM) also will be implementing and testing, where the environment is simulated to analyze and collect more information about the efficiency, performance and the maximum burst configuration [1]. Furthermore, it will be analyzing how this technology integrated with other TSN traffic shaper concepts and how it deals with them in this simulated network [1].

5. Conclusion

Today's industrial requirements are above the current standards Ethernet and there ability to fulfill the requirements. They are not enabling the communication between two end-devices simultaneously, low latency or to determent the critical data and their priority to handle it. Therefore, a new technology was developed as a solution for this problem to satisfy the communication requirements. Time sensitive Networking (TSN) provides low latency transfer, simultaneous communication between two end-devices, security and priority for critical data. TSN has several Standards that provide different functions and applications.

In this paper 802.1Qci and its definition for Per-stream filtering and policing was discussed. Because the technology relatively new, there are some gaps in the implementation and it still developing. A problem that 802.1Qci have it, when the internet daily scaling up and a lot of changes happen, how can be the policy of a network updated to still in full swing with these changes. A solution was to centralize the policies in one server that called Policy Server, to controls all the policies and update them according to the requirements that will take them from the CNC (look Section/Subsection 3.2).

In addition to these implementations, Credit Based Meter (CBM) will tested and analyzed in a simulated environment networks, where also will show how it can deal with other implementations.

References

[1] P. Meyer, "Preventing DoS Attacks inTime Sensitive Networking In-Car Networks through Credit Based ingress Metering."

[2] D. Greenfield, "4 reasons why time sensitive networking matters," 2016, [Online; accessed 28-September-2020].

[3] J. L. Messenger, "Time-Sensitive Networking: An Introduction."

[4] Cisco, "Time-sensitive networking: A technical introduction," 2017.

[5] "Ieee standard for local and metropolitan area networks–bridges and bridged networks–amendment 28: Per-stream filtering and policing," *IEEE Std 802.1Qci-2017 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, IEEE Std 802.1Q-2014/Cor 1-2015, IEEE Std 802.1Qbv-2015, IEEE Std 802.1Qbu-2016, and IEEE Std 802.1Qbz-2016)*, pp. 1–65, 2017.

[6] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-low latency (ull) networks: The ieee tsn and ietf detnet standards and related 5g ull research," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 88–145, 2019.

[7] J. H. Robert Barton, Maik Seewald, "Management of IEEE 802.1Qci Security Policies for Time Sensitive Networks (TSN)," 2018.