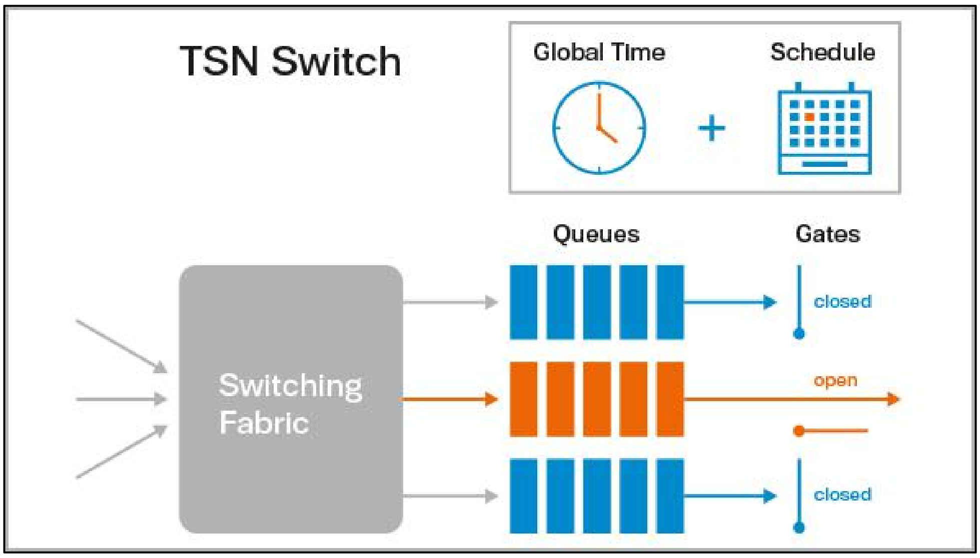
REPORT

Simulation and experimentation of TSN networks



SU Runbo

third year engineering student at the École Nationale Supérieure

d'Électricité et de Mécanique (ENSEM)

Abstract

This report presents a local analysis of TSN frames under hierarchical scheduling of TAS (Time-Aware Shaping) on TSN switches. The local analysis of TSN frames has been largely investigated in simulation models, but not yet with the presence of TSN switches. A realistic TSN traffic model was applied to several Hirschmann TSN switches. Moreover, time synchronization and TAS were utilized to implement a high accuracy scheduling. Its performance is illustrated to show how the frame priority works and also show the influence of IEEE802.1Qbv TAS on TSN network. This report proposes that a research should continue for the Software Defined Networking (SDN) supporting TSN, which has been recently derived.

Keywords : TSN(Time-Sensitive Networking); TAS(Time-aware shaping); switch; scheduling

Résumé

Ce rapport présente une analyse locale des trames TSN dans le cadre d'un scheduling hiérarchique de TAS (Time-Aware Shaping) sur les TSN switches. L'analyse locale des trames TSN a été largement étudiée dans les modèles de simulation, mais manque encore de la présence de TSN switch. Un modèle de trafic TSN réaliste a été appliqué à plusieurs Hirschmann TSN switches. De plus, la synchronisation temporelle et le TAS ont été utilisés pour mettre en place un scheduling de haute précision. Ses performances sont illustrées pour montrer comment la priorité de trame fonctionne et pour montrer également l'influence du TAS IEEE802.1Qbv sur le réseau TSN. Ce rapport propose de poursuivre les recherches sur une nouveauté récemment dérivée : le Software Defined Networking (SDN) qui supporte le TSN.

Mots clés : TSN(Time-Sensitive Networking); TAS(Time-aware shaping); switch; scheduling

Acknowledgements

I would like to acknowledge my indebtedness and render my warmest thanks to my supervisors, Prof. Ye-Qiong SONG and Mr. Theo Docquier for the following most important things. Firstly, I highly appreciate them for giving me the opportunity to do this project. And I am also thankful for their constructive feedback, their help and support allowed me to progress continuously during these 4 months of project.

And I highly appreciate the support provided by the administrative and technical staff of University of Lorraine and ENSEM engineering school.

Finally, my deep and sincere gratitude to my family for their continuous and unparalleled love, help and support.

Summary

1.Introduction

1.1. Project Motivation

The term industry 4.0 refers to the fourth industrial revolution, real-time industrial communication must be more robust and reliable between machines and industrial plants in this revolution. It is therefore necessary to produce methods and tools to master industrial real-time communication. One of the most renowned approaches is the Time-Sensitive Networking (TSN) technology. Today, this technology is a hot topic and opens many perspectives for all types of real-time applications: automotive, industrial control, smart grid, etc...

TSN is the latest important evolution of Ethernet, it is obviously classified in the data link (level 2) and physical (level 1) layers of the OSI model. It has bridged the real-time capacity gap of Ethernet as it provides deterministic real-time synchronized communication. TSN standards are defined by the IEEE 802.1Il working group. And the most important standards are IEEE 802.1AS (time synchronization) and IEEE 802.1Qbv (traffic planning and shaping).

My knowledge of this field at the start of the project was casual. I knew generally about networks without knowing anything specific about the real-time techniques. As this is a subject I am interested in making my career in, researching real-time Networking which I knew would be both challenging and interesting.

Initially, I wanted to find a project that would allow me to deepen my knowledge in computer science, in other words, a project that would serve me in my future research career, and fortunately, Professor Ye-Qiong SONG and Mr. Théo Docquier offered me a great opportunity to discover in detail laboratory life. So I joined the SIMBIOT team at Loria.

1.2.Report Structure

To begin with, this report presents the state of the art, particularly focus on the global market of TSN products which will be considerably useful and helpful in future works. It is followed by a succinct introduction of TAS Traffic model, in other words, how a pre-determined TAS scheduling can be correctly established. Then, my main missions will be addressed : the installation of industrial switches, the modeling of the TSN network, and the analysis of its performance in case of one-switch and three-switch in order to measure and test the TSN switches. This report will end with a conclusion of my project.

2.State of the art

2.1. The IEEE 802.1 Working Group

TSN is not only one communication protocol, but a collection of sub-standards from IEEE. Initially, IEEE 802.1Q-2005 introduced the prioritized BE traffic. In the same year, the IEEE AVB task group is created and has proposed IEEE 802.1Qav to shape the traffic ensuring thus bounded Worst-Case end-to-end Delay (WCD). After that, in 2012, the group was renamed to TSN whose purpose now is to extend the IEEE 802.1Q VLAN protocol to meet the requirements of time-sensitive applications.

2.2. TSN

IEEE 802.1 TSN technology is one of the most important evolution for industrial Internet of things. To address the TSN timing constraints, 'shapers' and 'scheduling' have been introduced and standardized, they are principally : IEEE 802.1Qav Forwarding and Queuing Enhancements for Time-Sensitive Streams, IEEE 802.1Qbv Enhancements to Traffic Scheduling: Time-Aware Shaper (TAS). As one new key feature of TSN, TAS is supposed to be capable of accommodating hard real-time streams with deterministic end-to-end delays. TAS uses a pre-determined scheduling, which guarantees timely transmission of trame CDT (Control Data Traffic). Besides, a guard-band (GB) is added to check that other traffic would not interfere with CDT traffic. TAS should be determined periodically by different time intervals. Some technical details of TAS will be presented in next section "a succinct introduction of TAS Traffic model". The table in Annex 16 gives more details about standards and amendments in TSN.

And another important technology of TSN is the time synchronization. Time plays an important role in the TSN network compared the Ethernet standards before such as IEEE 802.3 and IEEE 802.1Q. By clock synchronization, the network devices can run consistently and perform the required operations at the specified point in time. Time synchronization in TSN networks can be achieved with different technologies. Time in a TSN network is typically distributed from a central time source through the network itself. Time synchronization is one most important base in our experimentation, especially in the network topology study.

2.4. TSN product global market

More recently, TSN technology has achieved remarkable development. This technology is considered a non-negligible part of the 4rth Industrial Revolution because of the importance of the IoT. The TSN switch production is an indicator that the development of TSN technology that manufacturer have. Nowadays, more than dozen companies have produced their own TSN product or have launched the TSN product announcement. This part presents a survey of current products on the global market.

* <u>First category</u>

Company	Identification of product	TSN standards implemented	State	Annex
Cisco	Industrial Ethernet 4000	(1);(2)	(√)	2
Hirschmann	RS20-0800M2M2SDAEHH08.0.	(1);(2);(3)	(√)	3
NXP	SJA1105	(1);(2);(3);(8)	(√)	4
Kontron	KBox C-102-2 TSN STARTERKIT	(1);(2);(3);~~(4)~~	(√)	5
Renesas	R06PM0078EJ0100	(1);(2);(3);(4);(7);(8)	(√)	6
ANALOG DEVICES	TSN EVALUATION KIT	(1);(3);(4);(5);(7);(8)	(√)	7

*Second category

Company	Identification of product	TSN standards implemented	State	Annex
Winsystems	NET-429	(1);(2)	(√)	8
ORing	IGS-RX164GP+	(1);(4);(5)	(√)	9
TTTech	Nerve-MFN 100	(9)	(√)	10
Beckoff	EK1000	(9)	(√)	11
SoCe	MTSN IP Core	-	(√)	X
Siemens	To be determined	-	(*)	X
B&R	To be determined	-	(*)	X
Etherfly	To be determined	-	(*)	X

-The TSN standards' names would be numbered as follows :

(1) IEEE 802.1AS Timing and Synchronization for Time-Sensitive Applications

(2) IEEE 802.1Qav Forwarding and Queuing Enhancements for Time-Sensitive Streams

(3) IEEE 802.1Qbv Enhancements to Traffic Scheduling: Time-Aware Shaper (TAS)

(4) IEEE 802.1Qbu Frame Preemption

(5) IEEE 802.1Qcc Enhancements to SRP

(6) IEEE 802.1Qat Stream Reservation Protocol

(7) IEEE 802.3br Interspersed Express Traffic

(8) IEEE 802.1Qci Ingress Policing

(9) No accurate information for all standards

(n) The standards are confirmed

(n) Not yet tested

~~(n)~~ Surely in progress

-The different states of TSN switches would be represented as follows :

(√) Real physical TSN switches or cards produced

(*) Just announcement of TSN switches products

-The TSN product information of first group is publicly available online and some products are already used in our experiments such as Hirschmann and Cisco.

-For second group, either their product information is more or less hidden due to some unknown reasons, or some companies actually do not have real TSN products but the TSN product is ‘coming soon’ according to their announcement.

-All product information can be found from their data-sheet in numbered Annex.

Conclusion of TSN product global market :

The TSN product global market is divided into two groups by their states and the situation of ‘TSN standards implemented’. The first group has generally a high product quality and high product information transparency, this report proposes to choose this group in case of purchase. Considering the the lack of information transparency and some companies have not yet produced the real physical equipment, even some of them have highlights in “TSN standards implemented ”column, our conclusion has still some reservations about the second group.

3.Introduction of TAS Traffic model

3.1. The principle of Time-Sensitive Networks

In a more practical word, that frames of TSN should be transmitted depending on their VLAN tag and priority determined. These frames are filtered by the TSN switch following a schedule described by one TAS configuration cycle, this cycle then includes at least :

- A Guard-Band (GB) is a period during which all gates would be closed. This period allows a frame that has been started to finish sending. For this reason, the duration of the Guard-Band is based on the time it takes to transmit the maximum-size frame exiting in the network.

- The Control Data Traffic (CDT) slot, which is the period of time that the particular priority frame can be transmitted (priority 7 has been chosen as CDT priority). The CDT guarantees that periodical data has their own slot reserved on each cycle.

- The rest of the cycle is used to pass the rest of the data, which has lower priorities (priorities between 0 to 6). detailllling

The implementation of such a system makes it possible to pass priority 7 frames very quickly and predictably. The processing of this cycle can be expressed as following figure :

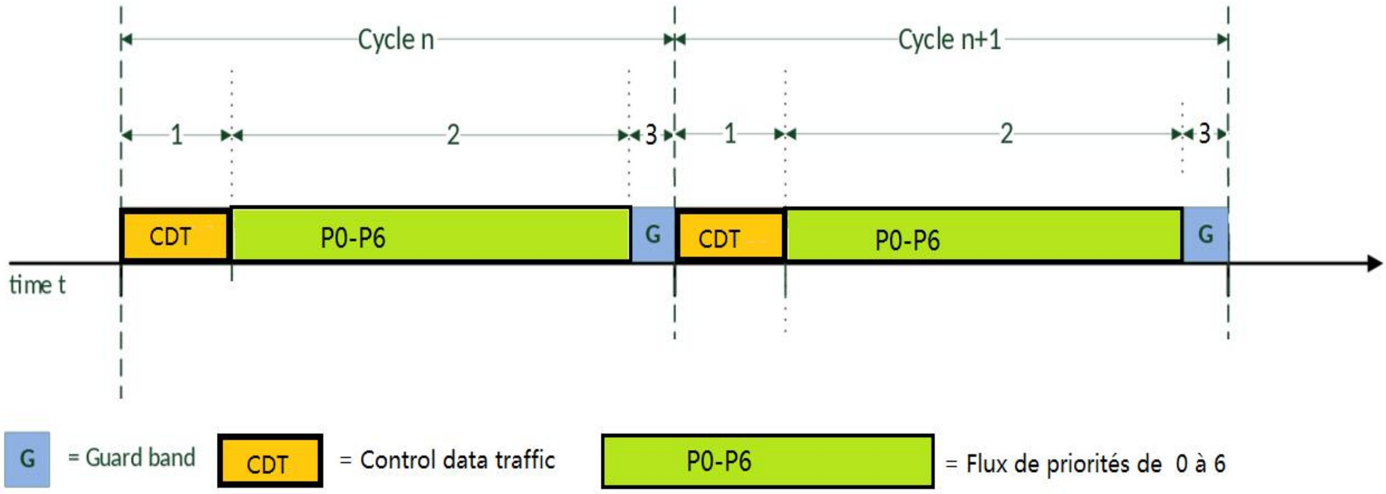


Figure 1 : Explanatory diagram of TAS configuration cycle

3.2. The TAS configuration cycle in table form

All TAS configuration cycle should be implemented in table form because the understandable way that TSN switch can read and load the configuration is uniquely the table form. For this purpose, the table form is supposed to be utilized in 'Gate Control list' configuration.

The Gate states are represented in a special way that the number indicates which port is open instead of '0' and '1' representing respectively the port is closed and the port is open.

TAS config.	Slot	Gate states	Interval
CDT	S_1	7	/
P0-P6	S_2	0,1,2,3,4,5,6	/
GB	S_3	-	/

Figure 2 : Sample of TAS configuration cycle in table form

The Figure 2 shows how the TAS configuration cycle can be expressed in table form. A TAS scheduling table is a list indicating which gate is open during each interval. In the above sample, the first column presents each different functional period, and the second column represents each slot that corresponds to the first column. Then, every number in third column indicates if the gate is open or not. For example, the '7' means the Gate 7 is open. Finally, the last column is the time interval of each slot.

3.3. Example of table form

TAS config.	Interval
Guard Band	26ms
CDT	150ms
Priorité 0 à 6	324ms

Figure 3 : Example of table form

According to the research 'Delay Analysis of AVB traffic in Time-Sensitive Networks (TSN)' did by Prof.Song and M.Maxim, the example of table form in Figure 3 has been proposed. Now, it's quite simple to understand the TAS configuration cycle through the table form, each slot reserved has their own TAS function and an interval corresponding to.

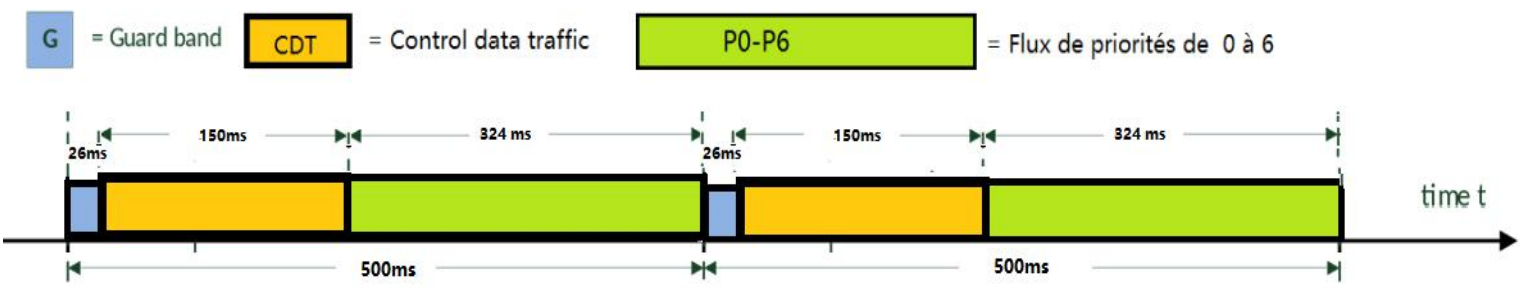


Figure 4 : TAS configuration cycle of this example

The Figure 4 shows the TAS configuration cycle of this example as defined in the Figure 3 example.

4.Missions

4.1. Installation of TSN switch

4.1.1. Installation of TSN on a Cisco IE 4000 switch

At the beginning of the project, one Cisco switch is proposed. Following the work of last internship trainee, the Cisco TSN switch can be configured by the control console of computer making the USB port be connected with the switch. Unfortunately, the some applications were not installed yet, the Cisco TSN should work under a CNC interface as a server. The first step of missions is the configuration and installation of TSN on a Cisco IE 4000 switch. After some works have been done, such as configuration of TSN on switch via USB port, then installation and discovering the CNC- server, one start-up guide was written and it's available in Annex 1, feel free if any one of you want more details.

4.1.2. Installation of TSN on a Hirschmann switch

Figure 5 : Hirschmann TSN swtich

After the configuration and installation of TSN on Cisco swith, the Hirschmann switch has arrived. *Hirschmann RS20-0800M2M2SDAEHH08.0. TSN switch is the first industrial switch to provide advanced security and real-time communication through time-sensitive networking (TSN) technology on some particular ports for standardized Ethernet usage in applications. (Site Hirschmann)*

Although the Hirschmann products are delivered without a default IP address, all configuration or modification should be implemented via a web interface at the local address of the switch. Therefore, an IP address has to be assigned to Hirschmann switch using a software 'HiDiscovery' developed by Hirschmann company.

The traditional method for configuring an IP address on a device is to use the serial port. But there will almost certainly be occasions when the correct serial cable is not available. This is where HiDiscovery comes into play. HiDiscovery will discover all Hirschmann devices on a LAN, even if they do not have an IP address. As it's shown in Figure 5, the "Signal" button will activate a device's LEDs, so it can be seen that which device that are communicating with. IP address can be assigned to the device, directly over the Ethernet connection. Besides, the HiDiscovery application is free of charge.

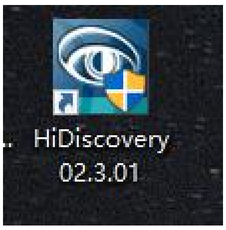


Figure 6 : HiDscovery

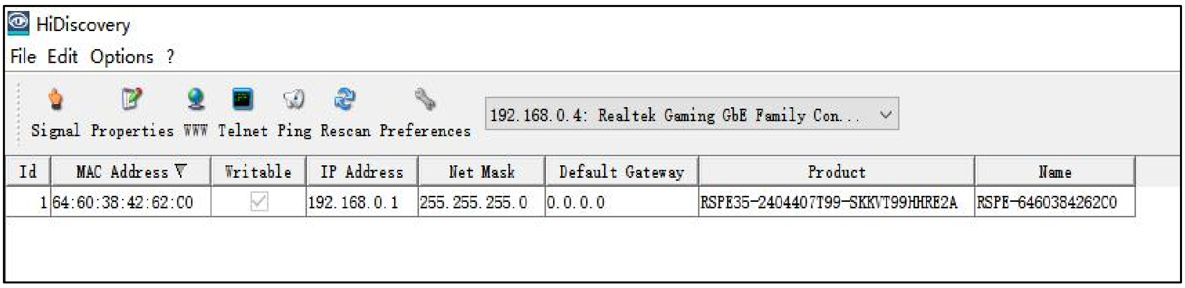


Figure 7 : Operation interface of HiDscovery

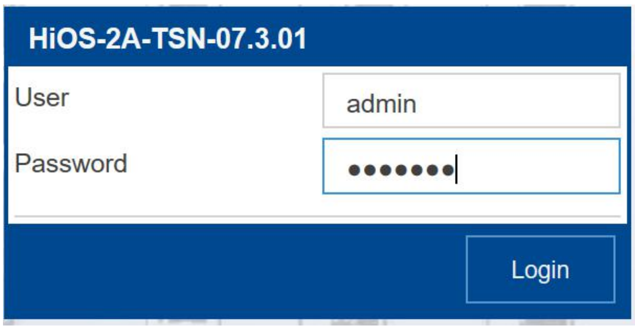


Figure 8 : Login of web interface

Once the IP address has been assigned, the web interface will be accessible with the user name and password. The default user name and password are respectively : ‘admin’ and ‘private’. To avoid any cases of confusion, the default user name and password should be all along kept.

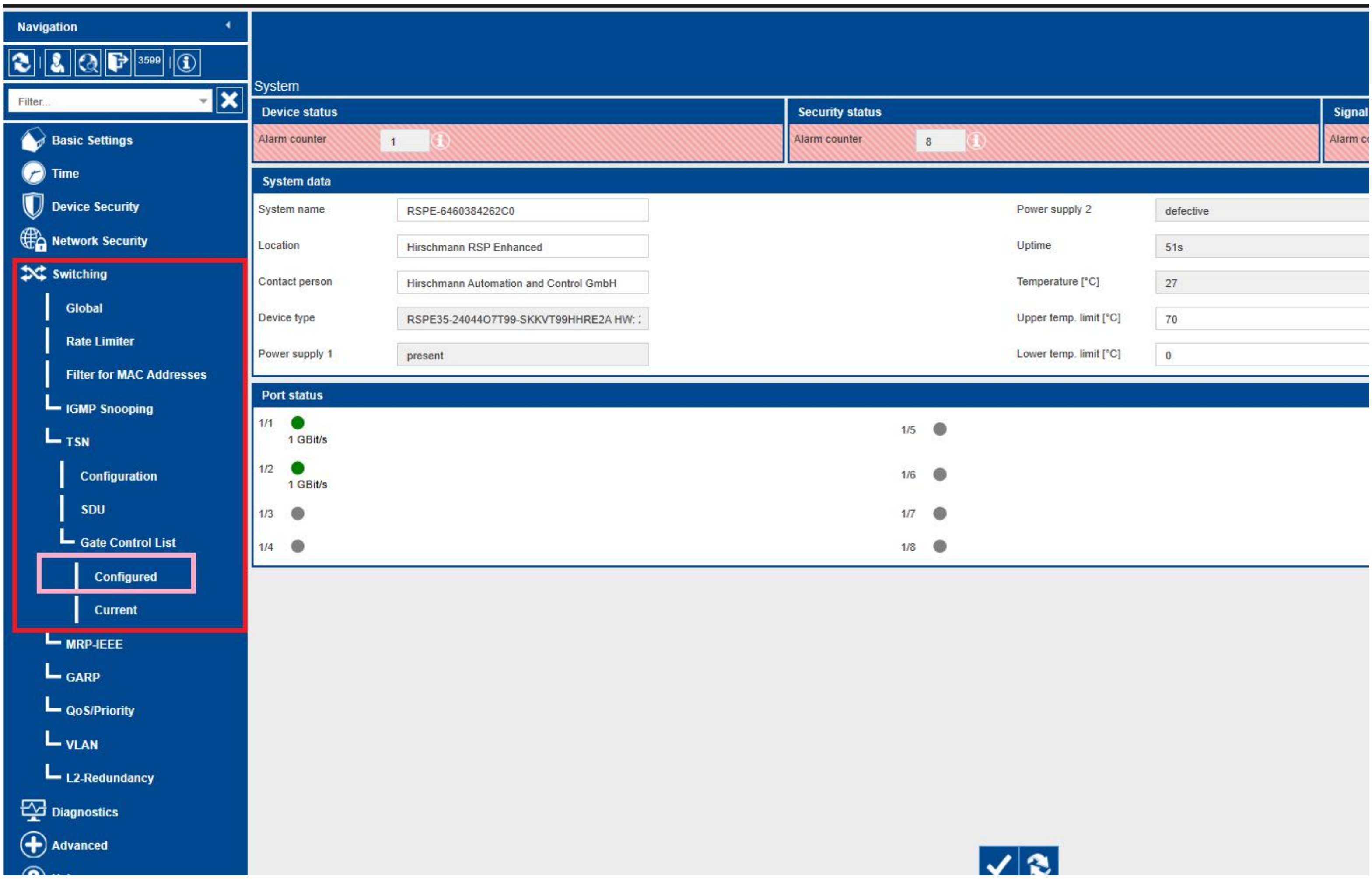


Figure 9 : Main page of Hirschmann web interface

Index	Gate states	Interval [ns]
1	-	26,000
2	7	150,000
3	0,1,2,3,4,5,6	324,000

Figure 10 : Gate Control list

The Figure 9 shows the looking of web interface main page and the way highlighted in red how to select the 'switching' branch, go to 'TSN' then 'Gate Control List' to configure the TAS scheduling table. The Figure 10 illustrated above shows how to implement the TAS scheduling table in web interface of Hirschmann switch.

Attention :

(1) Each Hirschmann TSN switch has only three TSN ports(ports1,2,3), which means only these three ports have TSN functions such as TAS, others are normal Ethernet ports.

(2) Each time that TAS scheduling table has been modified via web interface of Hirschmann switch, make sure that 'cycle time' is also correctly modified in 'TSN'-'Configuration' branch at the same time, otherwise the web interface will be out of service.

(3) The only thing that should be modified is 'IP address', other contents in each column should not be touched, especially 'MAC address'.

(4) Make sure the IP address assigned is one of network addresses available. There

are basically two methods to check. One classical way is to do 'Ping' between the Host and the switch. And the more practical way is to type the IP address of switch in any Web browser, the IP address is assigned correctly if the Web interface of Hirschmann switch is successfully accessed in.

Conclusion of installations of TSN on Cisco and Hirschmann switches :

The installation and configuration of Cisco IE 400 switch are more or less unamiable for inexperienced beginners who never use such switch. First of all, the installation requires a Linux operating system, or some necessary files can not be downloaded. Then, the Power unit has to be wired, the CNC interface has to be installed, and the Cisco switch has to be connected to USB port using 'PUTTY'. Even in the end, the configuration of Cisco switch brings a extremely time-consuming work of assembly language. Nothing is done 'automatically' but 'manually' during the installation and configuration of Cisco switch. Comparing with the Cisco switch, all works on Hirschmann switch become much simpler, just few steps should be followed such as address-assignment, the entrance of web interface, etc. Another highlight of Hirschmann switch is that all support documentation related is accessible through a '?' button at the bottom right corner. However, the web interface is frequently out of service while the 'Gate Control List' is being modified. Immediately after that, the web interface will be not accessible anymore unless the switch has been restarted. Besides this, it is totally unacceptable that the system saving occasionally takes too long, for example, sometimes nearly 40 seconds.

4.2. Experimentation of TSN networking on Hirschmann switch

4.2.1. Ostinato

Ostinato is a packet generator and network traffic generator with an intuitive GUI and support for network automation using a powerful Python API. Craft and send packets of several streams with different protocols at different rates.

Ostinato aims to provide a traffic generator and network testing tool for every network engineer and developer - something not possible today with existing commercial network test equipment. With the right tool, network developers and engineers can do their jobs better and improve the quality of networking products. (Source : site Ostinato)

In fact, there exists amount of packet generator software around the world. However, Ostinato is impressively developed and it has a well established GUI which is easy to intervene and manage the packet generation, for these reasons, it has been chosen as the packet generator in our experimentation.

<u>How to install Ostinato on Ubuntu ?</u>

Ostinato is available with default Ubuntu repositories and can be simply installed using the following command.

Step 1	`sudo apt-get update -y`	Copy
Step 2	`sudo apt-get install -y ostinato`	Copy

Figure 11 : Quick install steps of Ostinato

Execute the commands above step by step.

Note: -y flag means to assume yes and silently install, without asking any questions in most cases.

Figure 12: Ostinato

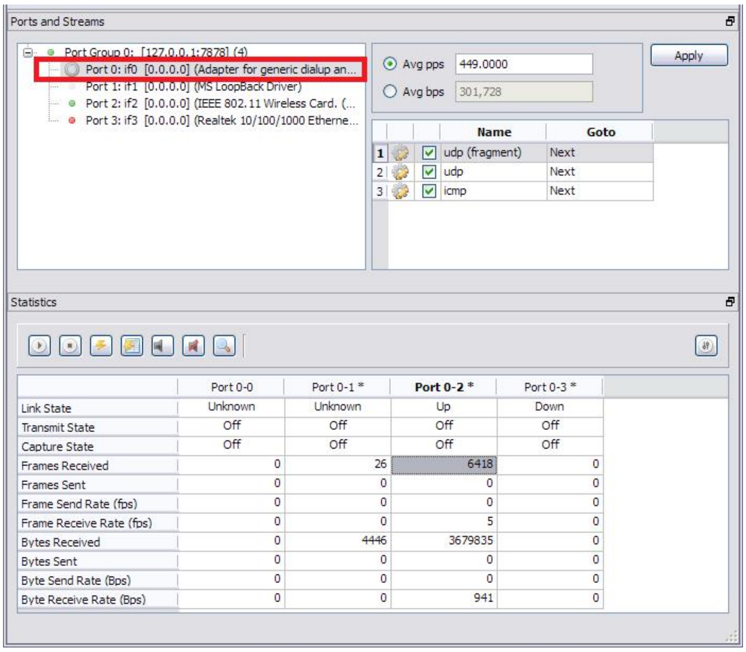


Figure 13 : Main window of Ostinato

The Figure 13 shows the main window of Ostinato. The Ethernet Ports are displayed at the upper left corner, the port connected to Hirschmann switch is normally the Port0. Once the Port0 is selected, the parameterized stream can be created beside the Port Part.

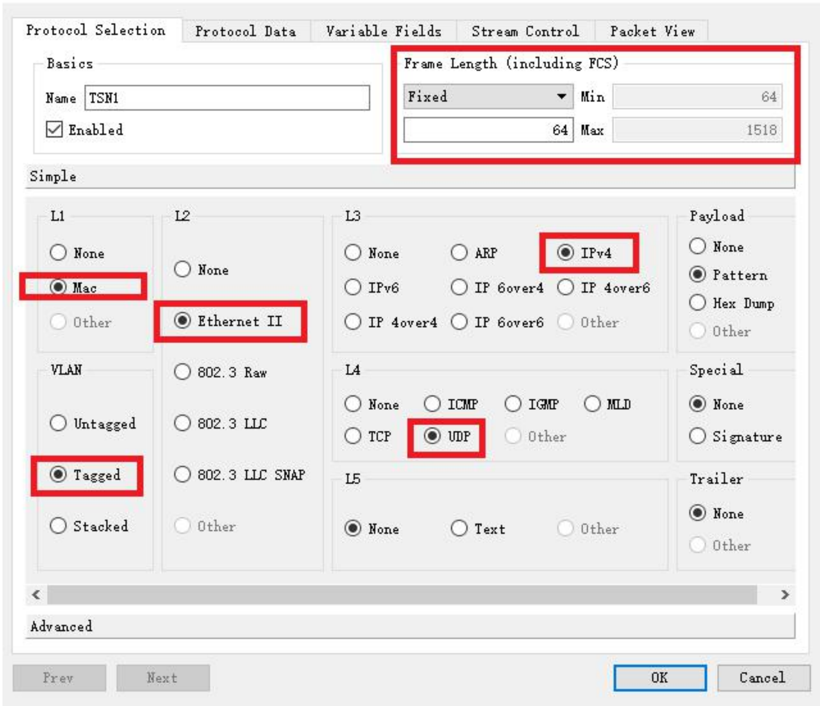


Figure 14 : Edit Stream

The Figure 14 shows the window of 'Edit Stream'. With the aid of this window, the Frame length and the protocols utilized of each layer can be configured. In the case of TSN networking, what should be selected are : the 'Tagged' in VLAN, 'Ethernet II' in L2, 'IPv4' in L3 and 'UDP' in L4. All options mentioned here are highlighted in red.

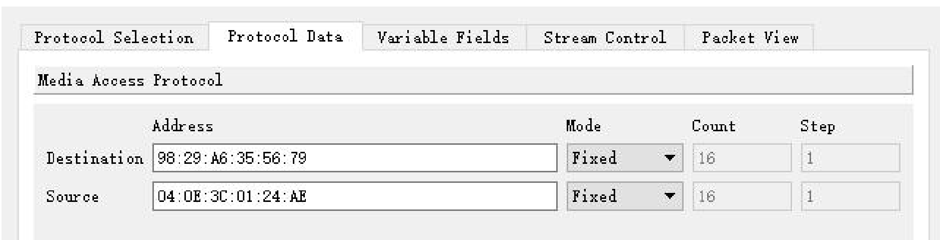


Figure 15 : MAC address

The MAC addresses of both destination and source should be initialed, as what is shown in Figure 15. Attention, any capture function of network traffic will not work if MAC address initialization is not completed or incorrectly done.

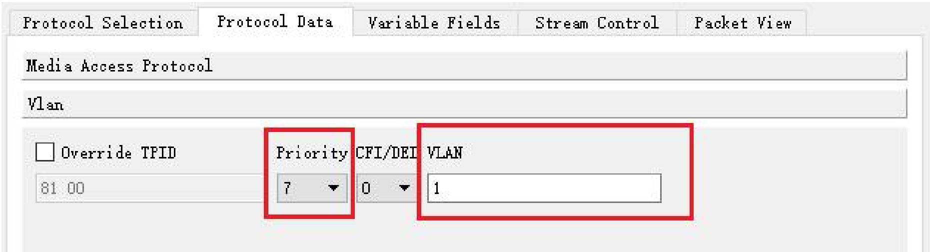


Figure 16 : Vlan configuration

Figure 16 shows the 'Vlan' configuration. The 'VLAN' field is the VLAN ID or VID. Each VLAN can be uniquely identified by VLAN ID, which is transmitted and received as IEEE 802.1Q tag in an Ethernet frame. By default all VLAN ID admitted on Hirschmann switch is VLAN IDs 1.With this in mind, the 'VLAN' field should be filled in '1'. And according to the TAS scheduling table determined, the 7 has to be selected as CDT priority.

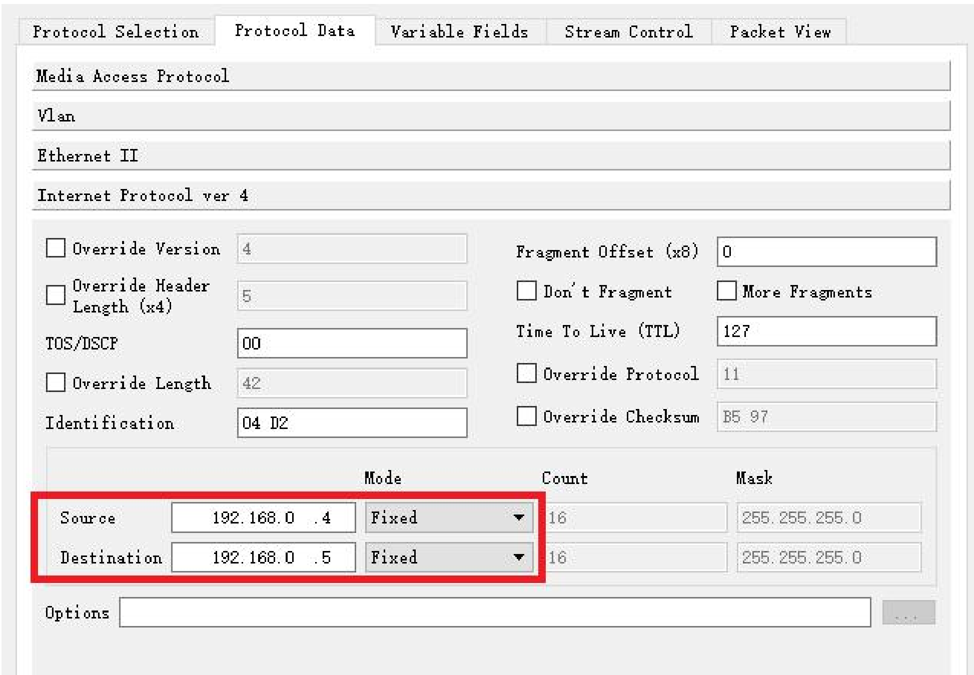


Figure 17 : IPv4 configuration

Figure 17 shows the 'IPv4' configuration. This one is simple to achieve by typing the IP addresses of the source and the destination.

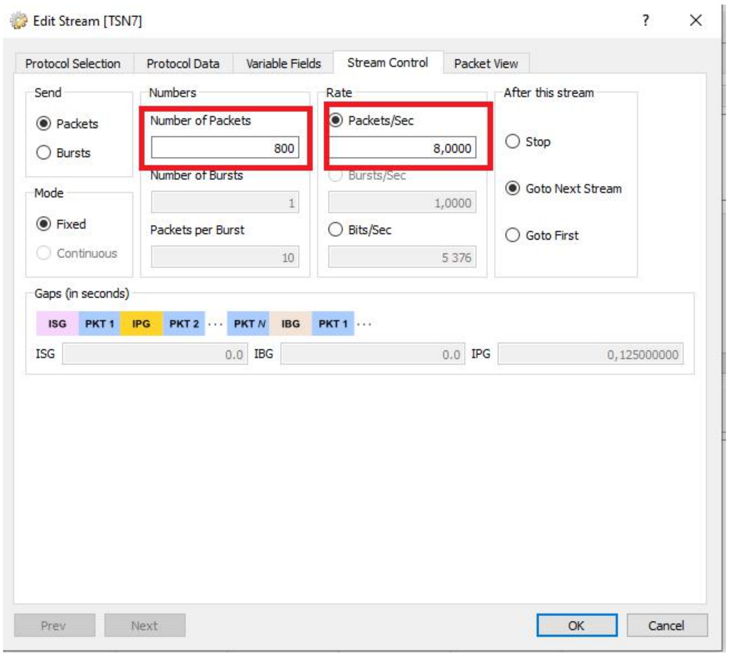


Figure 18 : Stream Control configuration

Figure 18 shows the 'Stream Control' configuration. The most important fields are respectively 'Number of Packets' and 'Packets/Sec'. 'Number of Packets' field is used for determine how many packets will be totally sent . The 'Packets/Sec' asks a value that how many packets should be periodically sent per second. For example, the 8 of 'Packets/Sec' means the packet generator Ostinato will send 8 packets per second, the fixed difference between each packet if 1/8=0.125(s). And 800 packets should be totally sent as what is shown in Figure 14, then the duration of packet generation will be 800/8=100s because of 8 packets/Sec.

Attention :

(1) The content of each configuration field can vary according to TSN scheduling determined.

(2) In case of 'Broadcast' transmission, the destination MAC address should be 'ff:ff:ff:ff:ff:ff' and IP address should be ' x, x, x,255'.

(3) To validate all TSN functions, three TSN ports(ports1,2,3) are supposed to be utilized rather than other normal ports.

4.2.2. Wireshark

*Wireshark is the world's foremost and widely-used network protocol analyzer and the network traffic capture at the same time. This software helps users to see what's happening on the network at a microscopic level. Originally named **Ethereal**, the project was renamed 'Wireshark' in May 2006 due to trademark issues. In fact, Wireshark is very similar to tcpdump, but has a graphical front-end, plus some integrated sorting and filtering options.* (Source : site Wireshark & Wikipédia)

We have learned a lot of using Wireshark thanks to lessons at Faculty of Science and Technology, and Wireshark is extremely strong as a networking analyse tool. There is no doubt that Wireshark is our first and only choice.

How to install Wireshark on Ubuntu ?

Wireshark is available with default Ubuntu repositories and can be simply installed using the following command.

Step 1	`sudo apt-get update -y`
Step 2	`sudo apt-get install -y wireshark`

Figure 19 : Quick install steps of Ostinato

Execute the commands above step by step.

Note: -y flag means to assume yes and silently install, without asking any questions in most cases.

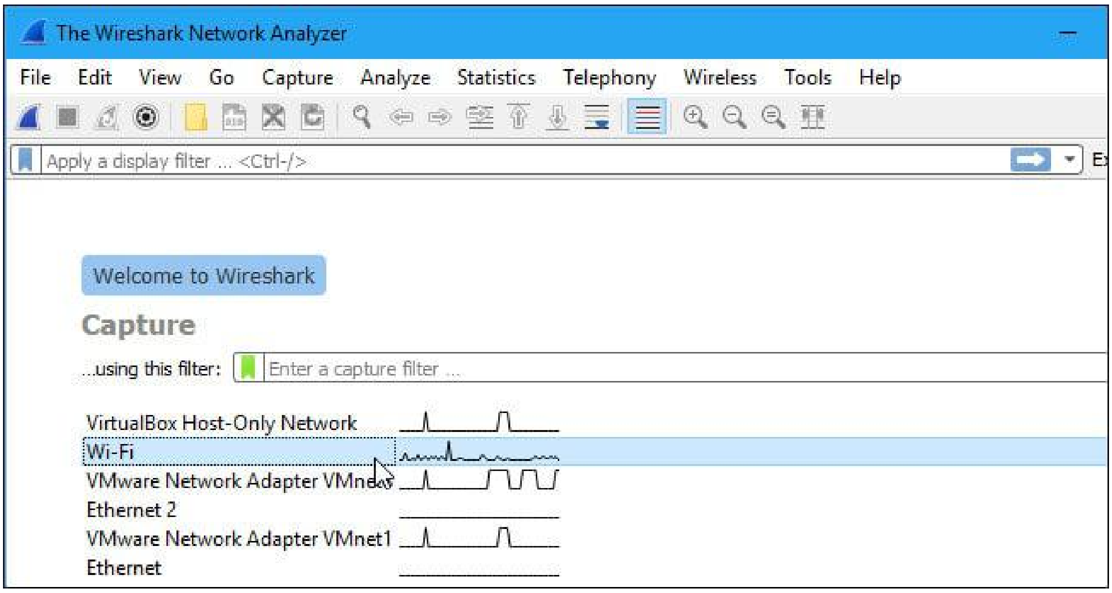


Figure 20 : Main window of Wireshark

The Figure 20 shows the main window of wireshark. The capture source should be chosen in this window. Considering the communication between the source and the switch is based on Ethernet ports, then the 'Ethernet' capture option need to be selected. In most of cases during the experimentation, the name of option should be selected starts with '**enp**'.

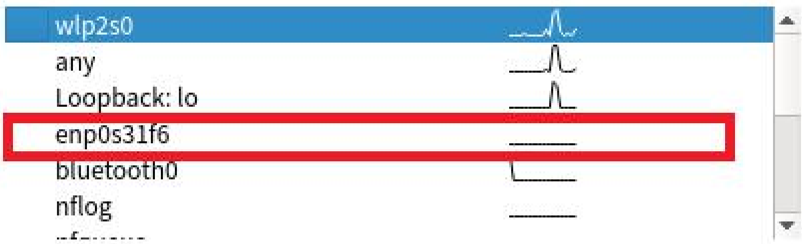


Figure 21 : Capture options

Such as what is shown in Figure 21, the capture option which should be selected is '**enp**0s316'.

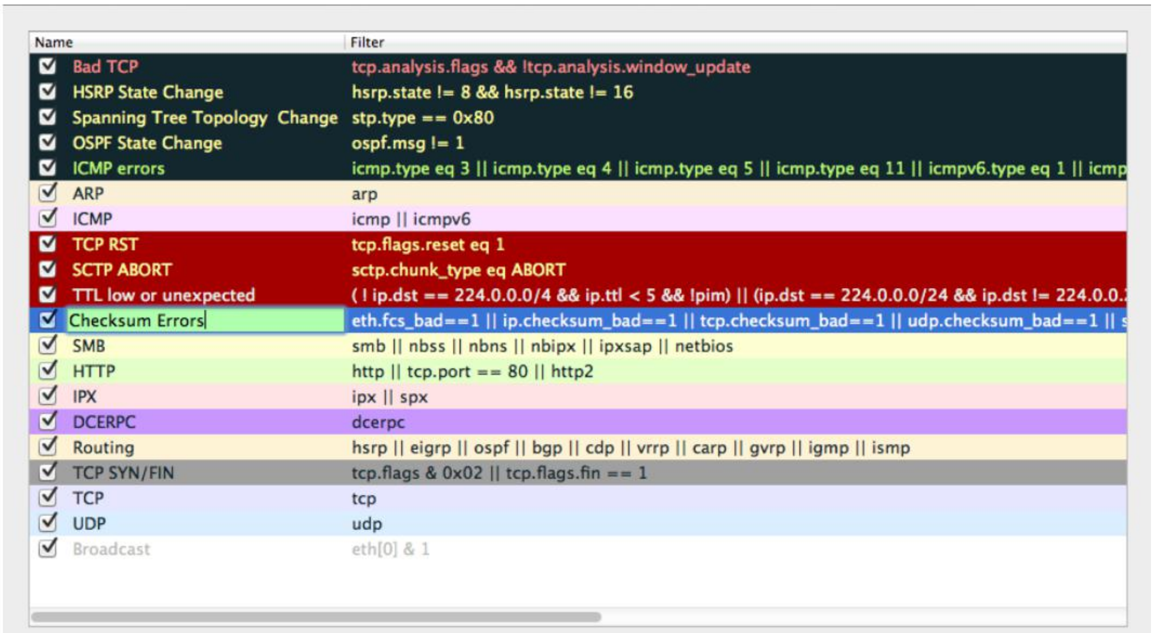


Figure 22 : Wireshark coloring rules

Wireshark can be set up so that it will colorize packets according to a display filter. This allows users to emphasize the packets they might be interested in. Coloring Rules of Wireshark is shown in Figure 22. All packets generated from Ostinato are configured to UDP, this is to say that the protocol filter is 'UDP' or the color which should be observed is blue.

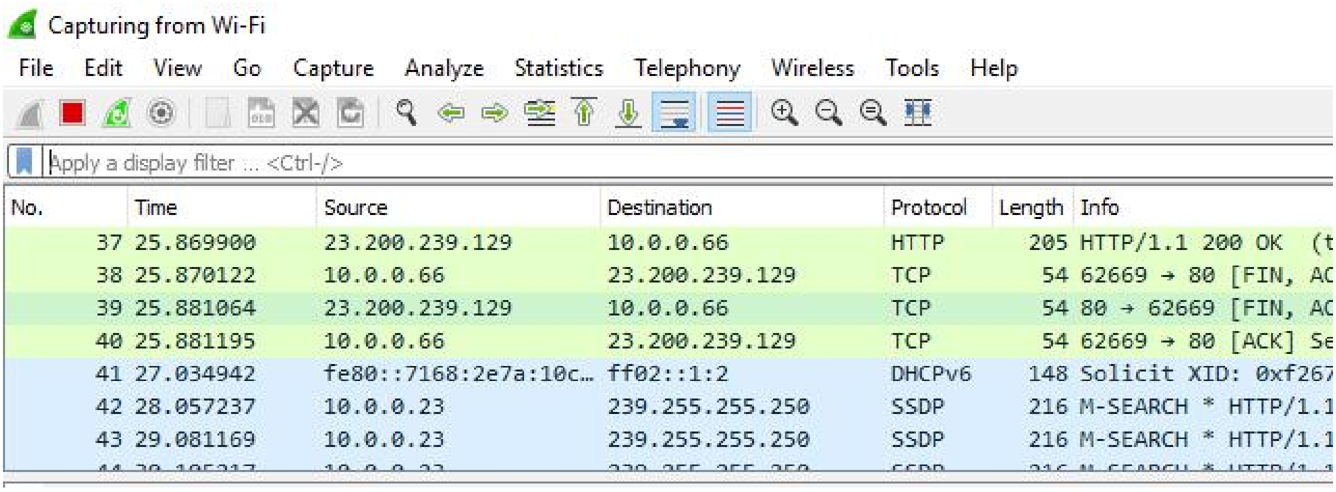


Figure 23 : example of Wireshark capturing

To take a closer look at how Wireshark works in the experimentation, the example of Wireshark capturing is illustrated in Figure 23. The first thing that should be done is to understand what they represent to. The fist two columns are the number of packets detected and the time (moment) of each. The third and fourth columns represent the addresses of Source and Destination, then the protocol utilized of each packet, followed by their length and some particular information.

4.2.3. The network topology of case study

The purpose of this section is to implement the case presented in the publication 'Delay Analysis of AVB traffic in Time-Sensitive Networks (TSN)' of Prof.Song and M.Maxim. The case is about an internal network containing 3 TSN switches. The TAS scheduling table can be found in section **3.3**. The first step is focused on one switch, test ones' capacities. Then the three Hirschmann TSN switches will be utilized to establish the The complete network topology is illustrated in Figure 24.

The TSN cycle has been reproduced in the Hirschmann switch whose configuration is accessible via a web interface at the local address of the switch. Due to the capacity limited of software Ostinato, the time unity has been set up to mini-second (ms) instead of micro-second (μ s). This case study will be achieved and analyzed in next section using Hirschmann switches.

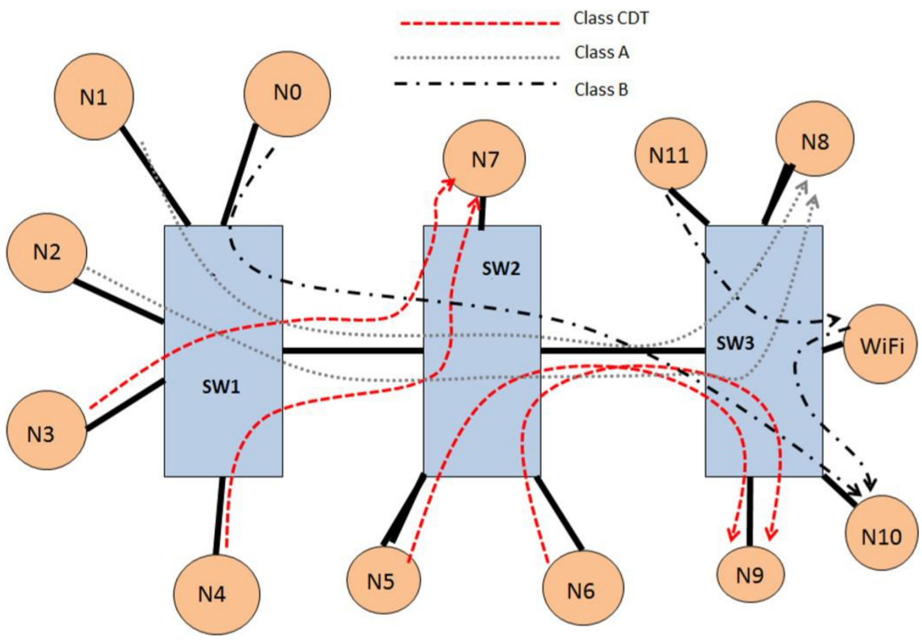


Figure 24 : Network topology (source : paper 'Delay Analysis of AVB tra ic in Time-Sensitive Networks (TSN)' of Prof.Song and M.Maxim)

4.2.4. Working environment

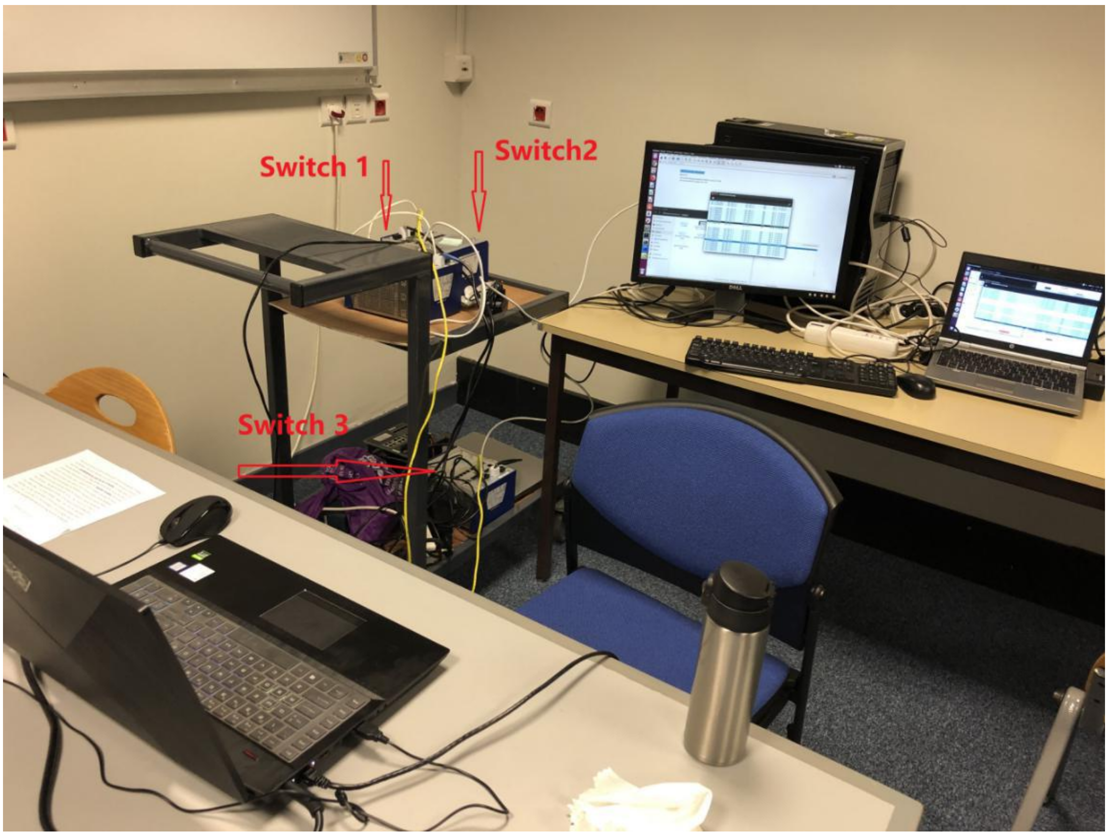


Figure 25 : Working environment

Then the complete network topology will be established by 3 identical Hirschmann TSN switches, 1 PC plays the source role and 3 PC will be used for capturing the output of each switch via the tool Wireshark, the photo of this studio is illustrated in Figure 25.

4.2.5. The one-switch work

With the configuration established in Figure 16 and Figure 18, 8 packets per second with VID 1 and priority 7 will be sent periodically, as the result, the time difference between each packet is fixed to 0.125s -> 125 ms. The priority 7 has been the 7 has to be selected as CDT priority, so the CDT interval will allow the priority 7 packets pass, and the rest packets which are not into CDT interval will be accumulated until the next cycle. From these facts, two different cases will appear in this one -switch work.

Based on a 150 ms CDT interval, the maximum packets that can be sent is two (125ms). On the other words, two packets can be transmitted in a normal way and two other packets which is out of CDT interval will be accumulated until the next cycle. Then the visualization of switch output will be : two packets have been detected almost simultaneously at the start of CDT interval, and two packets can be detected in normal transmission way. This case is shown in Figure 26, and it can be called ‘case 2-1-1’ or ‘case 2-2’ .

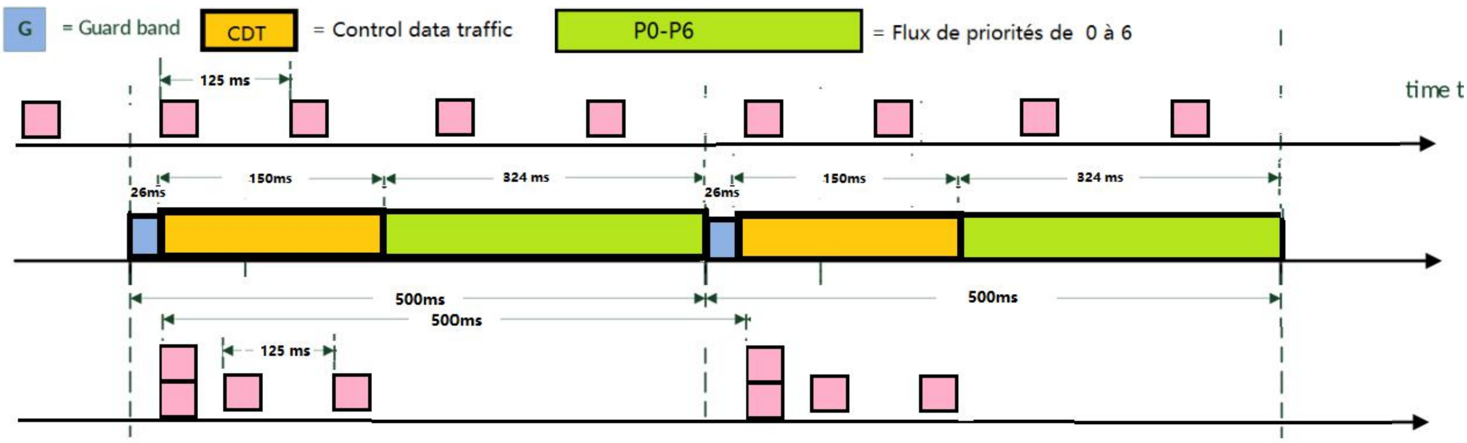


Figure 26 : Graphic description of case 2-1-1

Another case is, only one packet strictly into CDT interval, then other three packets will be accumulated until the next cycle. Therefore, the visualization of switch output will be : three packets have been detected almost simultaneously at the start of CDT interval, and only one packet can be detected in normal transmission way. This case is shown in Figure 27, and it can be called ‘case 3-1’ .

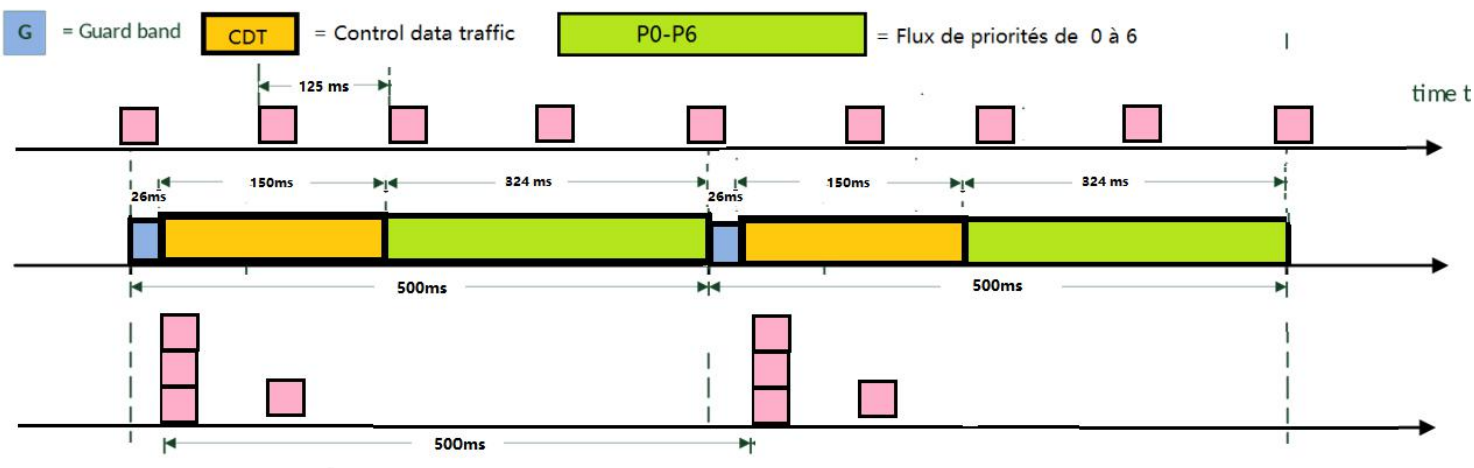


Figure 27 : Graphic description of case 3-1

49	15:36:58,2557421…	192.168.0.4	192.168.0.255	UDP	60	0 → 0 Len=14
50	15:36:58,2557575…	192.168.0.4	192.168.0.255	UDP	60	0 → 0 Len=14
51	15:36:58,2616974…	192.168.0.4	192.168.0.255	UDP	60	0 → 0 Len=14
52	15:36:58,3866893…	192.168.0.4	192.168.0.255	UDP	60	0 → 0 Len=14
53	15:36:58,7557334…	192.168.0.4	192.168.0.255	UDP	60	0 → 0 Len=14
54	15:36:58,7557493…	192.168.0.4	192.168.0.255	UDP	60	0 → 0 Len=14
55	15:36:58,7616896…	192.168.0.4	192.168.0.255	UDP	60	0 → 0 Len=14
56	15:36:58,8866877…	192.168.0.4	192.168.0.255	UDP	60	0 → 0 Len=14

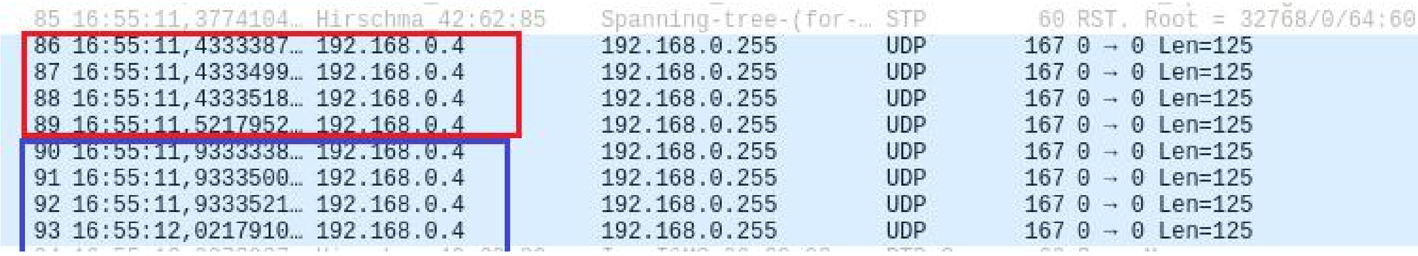
Figure 28 : Visualization of case 2-1-1 via Wireshark

According to the visualization illustrated in Figure 28, from packets 49 to 56 captured by Wireshark : 4 packets received can be seen between 58.255s to 58.386s, and other 4 packets received can be seen between 58.755s to 58.885s. The first 4 packets received which are highlighted in red correspond to the first cycle in Figure 26, and the second 4 packets highlighted in blue correspond to the second cycle in Figure 26.

The time differences between packets 49-50 and packets 53-54 are so small that the way they are transmitted can be described as 'these two packets are transmitted at the same time'. And the time difference between the third and forth packets in each cycle is strictly 0.125s : 58.386-58.261=0.125s, 58.886-58.761=0.125s. Besides, each time difference between 49-53, 50-54, 51-55 and 52-56 packets is perfectly the cycle time 0.5s.

Look back to the Figure 26 which shows the case of 2-1-1, every propriety is checked. The first two packets detected are transmitted almost at the same time, the time difference between the third and forth packets is strictly 0.125s, and the cycle time is always 0.5s.

From these evidences calculated and checked, the Hirschmann switch works correctly in case 2-1-1.



85	16:55:11,3774104…	Hirschma_42:62:85	Spanning-tree-(for-…	STP	60 RST. Root = 32768/0/64:60
86	16:55:11,4333387…	192.168.0.4	192.168.0.255	UDP	167 0 → 0 Len=125
87	16:55:11,4333499…	192.168.0.4	192.168.0.255	UDP	167 0 → 0 Len=125
88	16:55:11,4333518…	192.168.0.4	192.168.0.255	UDP	167 0 → 0 Len=125
89	16:55:11,5217952…	192.168.0.4	192.168.0.255	UDP	167 0 → 0 Len=125
90	16:55:11,9333338…	192.168.0.4	192.168.0.255	UDP	167 0 → 0 Len=125
91	16:55:11,9333500…	192.168.0.4	192.168.0.255	UDP	167 0 → 0 Len=125
92	16:55:11,9333521…	192.168.0.4	192.168.0.255	UDP	167 0 → 0 Len=125
93	16:55:12,0217910…	192.168.0.4	192.168.0.255	UDP	167 0 → 0 Len=125

Figure 29 : Visualization of case 3-1 via Wireshark

According to the visualization illustrated in Figure 29, from packets 86 to 93 captured by Wireshark. 4 packets received can be seen between 11.433s to 11.521s, and other 4 packets received can be seen between 11.933s to 12.021s. The first 4 packets received which are highlighted in red correspond to the first cycle in Figure 27, and the second 4 packets highlighted in blue correspond to the second cycle in Figure 27.

The time differences between packets 86-87-88 and packets 90-91-92 are so small that the way they are transmitted can be described as ‘these three packets are transmitted at the same time’. And the time difference between the third and forth packets in each cycle is strictly less than 0.125s. Besides, each time difference between 86-90, 87-91, 88-92 and 89-93 packets is perfectly the cycle time 0.5s.

If we look back to the Figure 27 which shows the case of 3-1, every propriety is checked, the first three packets detected are transmitted almost at the same time, the time difference between the third and forth packets is strictly less than 0.125s, and the cycle time is always 0.5s.

From these evidences calculated and checked, the Hirschmann switch works correctly in case 3-1.

Conclusion of the one-switch work :

As what is discussed before, the Hirschmann TSN switch works correctly in case of one-switch. Every propriety shown in Figure 26 and Figure 27 has been checked by some particular time differences calculated. The Hirschmann switch accumulated the packets out of the CDT interval, and the packets into CDT have been correctly transmitted, then the rule of cycle time 0.5s is always well respected. The test results of the case 2-1-1 and the case 3-1 are just two cases chosen in order to test Hirschmann TSN switch, more work can be done considering some sort of network topology can be more complicated.

4.2.6. The three-switch work

In this section, the first thing which should be known is the delay of TAS scheduling table. In case of three-switch work, the configuration of each TSN switch should be identical. However, the some TSN functions will not be successfully implemented because of ignoring of the data transmission time.

To simplify the explication, the TAS scheduling table to use is the same in section 3.3. For example, if the size of one packet is 64 bytes by default of Ostinato, and the port transmission rate is set up to Gbit/s, so the transmission time will be $64 \times 8/ 10^9 = 5.12\text{x}10^{-7}$(s) applying the formula (Size of data x 8)/transmission rate. $5.12\text{x}10^{-7}$(s)=$512\text{x}10^{-9}$(s)=512(ns). To make sure the delay is not 'just enough', the transmission time or the delay can be doubled : 512 x 2=1024 (ns) ≈ 1000 (ns)

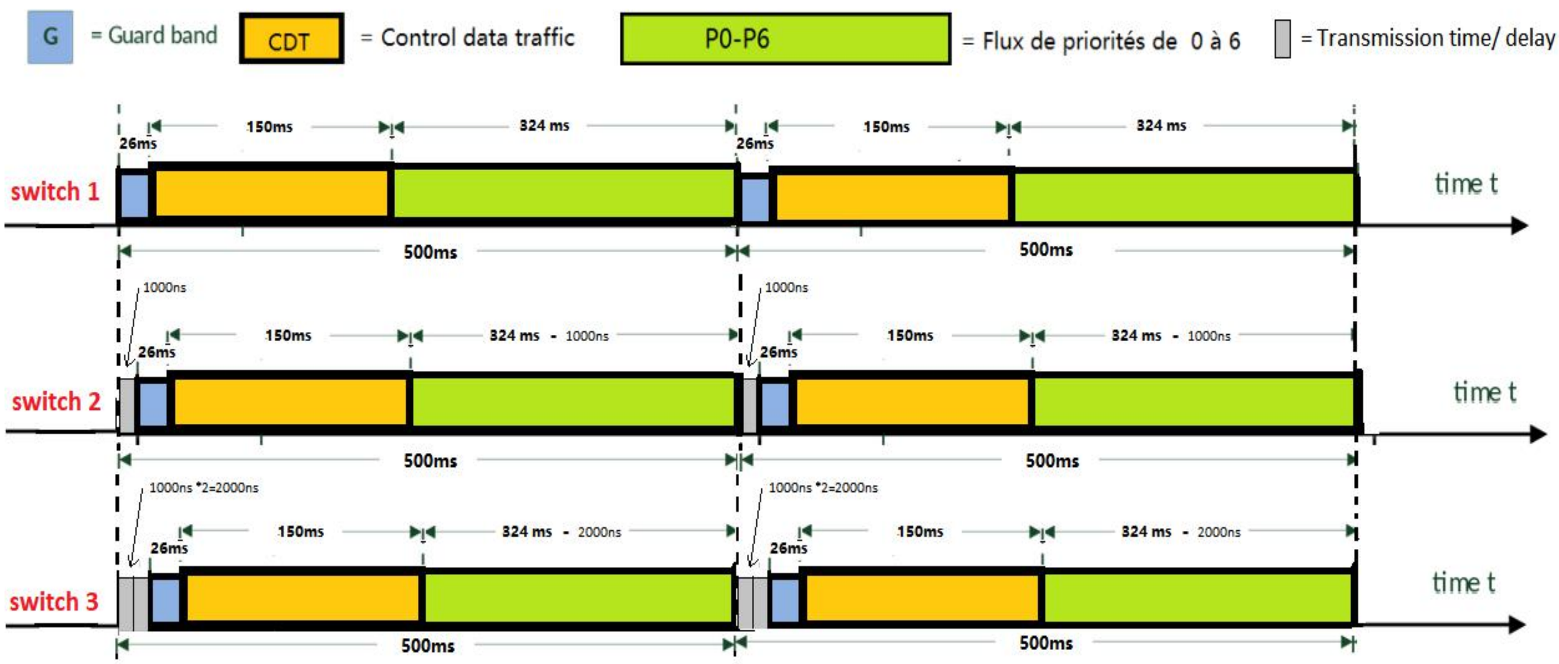


Figure 30 : Example of delayed TAS configuration cycle in three-switch work

The Figure 30 shows an example of delay TAS configuration in case of three-switch. The first TAS configuration cycle of switch 1 should not be modified, then one delay/ transmission time should be added at the start of the second TAS configuration cycle of switch 2, and the third TAS configuration cycle of switch 3 will be added similarly to switch 2 but double again the time because another transmission time is supposed to be taken into account.

The way that the TAS configuration delays can also be expressed in table form.

TAS config.	Gate states	Interval
GB	-	26 ms
CDT	7	150 ms
P0-P6	0,1,2,3,4,5,6	324 ms

Figure 31 : TAS configuration table of switch 1

TAS config.	Gate states	Interval
P0-P6	0,1,2,3,4,5,6	1000 ns
GB	-	26 ms
CDT	7	150 ms
P0-P6	0,1,2,3,4,5,6	324ms - 1000ns

Figure 32 : TAS configuration table of switch 2

TAS config.	Gate states	Interval
P0-P6	0,1,2,3,4,5,6	2000 ns
GB	-	26 ms
CDT	7	150 ms
P0-P6	0,1,2,3,4,5,6	324ms - 2000ns

Figure 33 : TAS configuration table of switch 3

The Figures from 31 to 33 show the TAS scheduling table of each switch. The delay/ transmission time is replaced by P0-P6 interval because this duration can be reserved for any no-CDT packets. As what is discussed before, the TAS configuration cycle of switch 1 should not be modified, so the table in Figure 31 is exactly the same as in section 3.3. Then one delay/ transmission time should be added at the start of the second TAS scheduling cycle of switch 2, this can be seen in Figure 32 that the delay of transmission has been added in the first line. And the third TAS configuration cycle of switch 3 will be added similarly to switch 2 but with doubled transmission time because another transmission time has to be taken into account, this is shown in Figure 33 that the first line doubled the value of first line of switch 2 table. Finally, do not forget that the cycle time is always fixed to 0.5s for all our experimentation, that means the time added before should be subtracted from last lines of switch 2 and switch 3, which are 324ms-1000ns and 324ms-2000ns.

According to the theory of TAS scheduling delay, once the TAS scheduling tables of Hirschmann switches have been correctly built, the outputs of all switches are supposed to be the same. On other words, except that the switch 1 has chance to accumulate the packets, the switch 2 and switch 3 can only receive the packets transmitted into the CDT intervals. If the switch 1 output is case 2-1-1, then the switch 2 and switch 3 outputs should be case 2-1-1 as well. If the switch 1 output is case 3-1, then switch 2 and switch 3 outputs should be case 3-1 as well.

4.3. Analysis of three-switch performance

Following the three-switch work discussed before, if the switch 2 and switch 3 have different cases from switch 1, there will be 4 possibilities :

(1) The TAS scheduling tables are not built correctly, either the calculation mistake has been made somewhere or the interval values are incorrectly entered. This possibility is the easiest to deal with. Once the possibility (1) has been checked, that means the TSN switch have problem somewhere because next possibilities all point out the problem comes from TSN switch.

(2) The time synchronization dose not work, and this point should be treated very carefully because any TSN functions will not make sense without the time synchronization. The importance of time synchronization has been introduced in section 2.2. For this reason, the time synchronization should be checked by Hirschmann company, and another thing which should be payed attention to is the difference between clock synchronization and time synchronization.

(3) The function of TAS scheduling works correctly in case of one-switch, but ‘no’ in three-switch work even the time synchronization is implemented. The three-switch work is too complicated to explain especially how each switch accumulate internally the packets out of CDT interval although each switch output can be visualized. And another difficulty of locating the problem of Hirschmann TSN switch is that it has only 3 TSN ports.

(4) Both the (2) and the (3) at the same time, the worst possibility.

<u>How to do the experimentation of three-switch work</u> :

To test the TSN switches is case of three-switch and to understand better these four possibilities, two specific examples have been implemented and studied.

The first example is the ‘wrong’ one : the TAS scheduling tables of 3 switches are totally correct, but the results obtained in the end is not what should be released from the correct configuration. That’s why the first example is called ‘wrong’ one, the method is correct but the result is not what it should be.

The second example is the ‘wired’ one : the TAS scheduling tables of 3 switches are totally incorrect, but the results obtained in the end is what should be released from the correct configuration. That’s why the second example is called ‘wired’ one, the method is wrong but the result is ‘welcome’.

4.3.1.Experimentation of ‘wrong’ one example :

All TAS scheduling tables of first example are illustrated in Figures 31, 32 and 33. The graphic description of these is also shown in Figure 30.

The way that source packets are generated is illustrated in Figure 18, it’s always 8 packets/ second. The switch 1 output is illustrated in Figures 28 and 29, they are either case 2-1-1 or case 3-1, and the graphic description of first example can be found in Figures 26 and 27.

407	17:37:57,0358969…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
408	17:37:57,0359084…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
409	17:37:57,0359103…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
410	17:37:57,0359122…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
411	17:37:57,0758878…	Hirschma_42:62:85	Spanning-tree-(for-…	STP	60	RST. Root = 32768/0/64
412	17:37:57,5358925…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
413	17:37:57,5359079…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
414	17:37:57,5359101…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
415	17:37:57,5359119…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125

Figure 34 : Visualization of output of switch 2 in ‘wrong’ one case

As the visualization illustrated in Figure 34, from packets 407 to 415 captured by Wireshark. 4 packets have been received since 57.035s, and other 4 packets have been received since 57.535s. The time differences between packets 407-410 and packets 411-415 are so small that the way they are transmitted can be described as ‘these four packets are transmitted at the same time’. The theoretical result should be either case 2-1-1 or case 3-1, but the visualization dose not match either case. By the way, this result can be called ‘case all-4’. Therefore, it is quite clear that there is something wrong because the case all-4 appears surprisingly instead of case 2-1-1 and case 3-1. Only one propriety checked is the duration of cycle 0.5s.

612	54.432735337	192.168.0.4	192.168.0.255	UDP	167	0x04d2 (1234)
613	54.432742292	192.168.0.4	192.168.0.255	UDP	167	0x04d2 (1234)
614	54.432743682	192.168.0.4	192.168.0.255	UDP	167	0x04d2 (1234)
615	54.432744909	192.168.0.4	192.168.0.255	UDP	167	0x04d2 (1234)
616	54.549345633	Hirschma_42:63:05	Spanning-tree-(for-b…	STP	60	
617	54.932736954	192.168.0.4	192.168.0.255	UDP	167	0x04d2 (1234)
618	54.932756883	192.168.0.4	192.168.0.255	UDP	167	0x04d2 (1234)
619	54.932760287	192.168.0.4	192.168.0.255	UDP	167	0x04d2 (1234)
620	54.932762693	192.168.0.4	192.168.0.255	UDP	167	0x04d2 (1234)

Figure 35 : Visualization of output of switch 3 in ‘wrong’ one case

As the visualization illustrated in Figure 35, from packets 612 to 620 captured by Wireshark. 4 packets have been received since 54.432s, and other 4 packets have been received since 54.932s. The time differences between packets 612-615 and packets 617-620 are so small that the way they are transmitted can be described as ‘these four packets are transmitted at the same time’. Once the switch 2 output represents case all-4, the switch 3 output will be the same because these each 4 packets are grouped after the accumulation, this point can not explain the ‘something wrong’ in switch 2. Again, only one propriety checked is the duration of cycle 0.5s.

Conclusion of the experimentation of ‘wrong’ one example

From the results obtained of switch 2 and switch 3, the switch 3 has the same case as switch 2 does, they both have case all-4. Initially, the switch 2 and switch 3 outputs are supposed to be the case 3-1 or case 2-1-1, however, these outputs are totally different as what is expected. The 4 possibilities that are discussed before should be checked one by one. The possibility (1) has been be easily checked, the possibility (2) and possibility (3) are difficult to check, but it is quite obvious that there is something 'wrong'. Therefore, one conclusion can be made : the first example of three-switch work is really the 'wrong' one without checking the possibility (2) and possibility (3) one by one. There is a technical problem in Hirschmann TSN switch.

4.3.2.Experimentation of 'wired' one example :

TAS config.	Gate states	Interval
P0-P6	0,1,2,3,4,5,6	124ms
GB	-	26 ms
CDT	7	150 ms
P0-P6	0,1,2,3,4,5,6	324ms - 124ms

Figure 36 : TAS configuration table of switch 2

TAS config.	Gate states	Interval
P0-P6	0,1,2,3,4,5,6	300ms
GB	-	26 ms
CDT	7	150 ms
P0-P6	0,1,2,3,4,5,6	324ms - 300ms

Figure 37 : TAS configuration table of switch 3

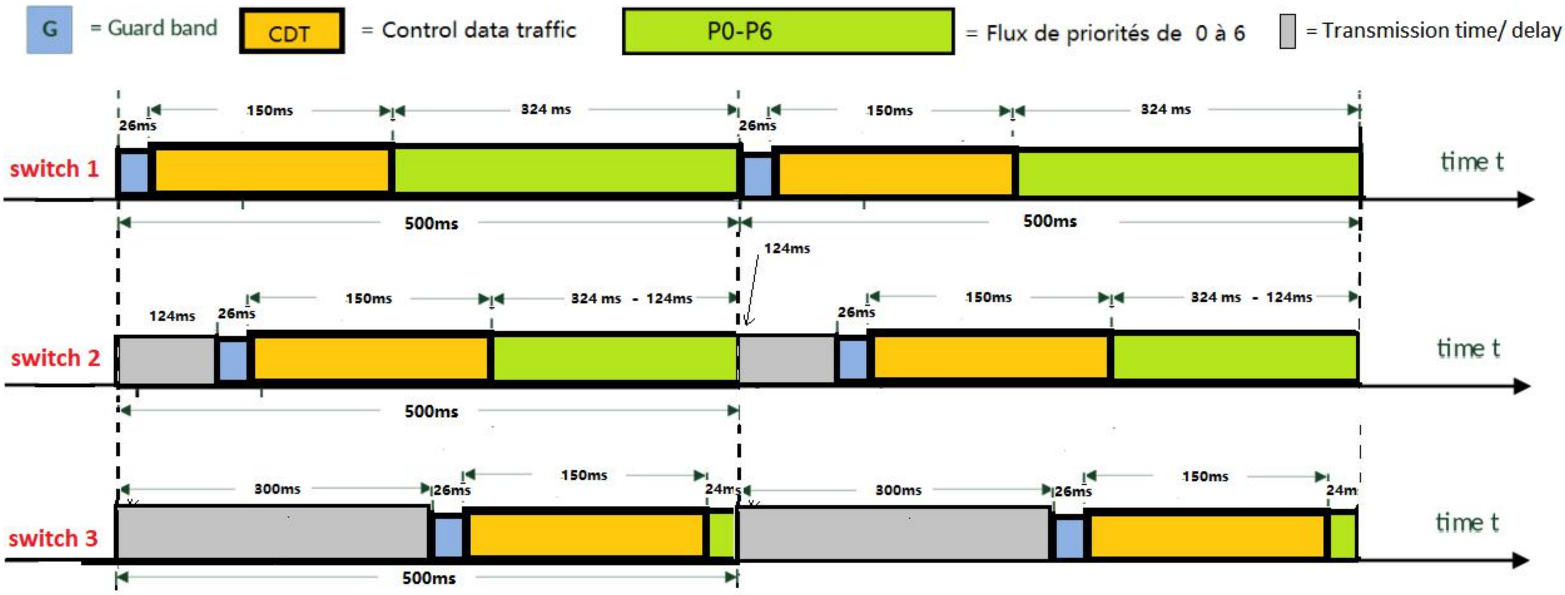


Figure 38 : Graphic description of second example - the 'wired' one

For second example, also the 'wired' one, the TAS scheduling table of first switch is shown in Figure 31 which is not touched. And the TAS scheduling tables of switch 2 and switch 3 are respectively illustrated in Figures 36 and 37. The graphic description of these configurations is also shown in Figure 38.

The way that source packets are generated is illustrated in Figure 18, it's always 8 packets/ second. The switch 1 output is illustrated in Figure 29, it represents the case 3-1.

From the graphic description of switch 2 and switch 3, it is clearer that the delay/transmission time has been raised to 124ms for switch 2 and 300ms for switch 3. According to the switch 1 output in figure 29, the time difference between the first three packets and the last packet is 11.521-11.433=0.088s =88ms, it is less than 120ms as delay/transmission time of switch 2. So the last one packet would wait for only 88ms after the three first packets are instantly transmitted. As the result, all packets will be accumulated because they are all out of CDT interval of switch 2. From these facts, the switch 2 output should be the case all-4.

Once the switch 2 output is the case all-4, the switch 3 output should be the same case.

90	16:55:11,9333338…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
91	16:55:11,9333500…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
92	16:55:11,9333521…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
93	16:55:12,0217910…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
94	16:55:12,2072837…	Hirschma_42:62:80	IeeeI&MS_00:00:00	PTPv2	60	Sync Message
95	16:55:12,2072904…	Hirschma_42:62:80	IeeeI&MS_00:00:00	PTPv2	60	Follow_Up Message
96	16:55:12,2072915…	Hirschma_42:62:80	IeeeI&MS_00:00:00	PTPv2	78	Announce Message
97	16:55:12,4333243…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
98	16:55:12,4333403…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
99	16:55:12,4333424…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125
100	16:55:12,5217830…	192.168.0.4	192.168.0.255	UDP	167	0 → 0 Len=125

Figure 39 : Visualization of output of switch 2 in 'wired' one case

As the visualization illustrated in Figure 39, from packets 90 to 100 captured by Wireshark, 4 packets have been received from 11.933s to 12.021s, and other 4 packets have been received from 12.433s to 12.521s. The time differences between packets 90-91-92 and packets 97-98-99 are so small that the way they are transmitted can be described as 'these three packets are transmitted at the same time'. The theoretical result should not be case 3-1 but all-4, and therefore it is quite clear that there is something wrong because the case 3-1 appears surprisingly instead of case all-4. Only one propriety checked is the duration of cycle 0.5s.

188	23.723992918	Hirschma_42:63:00	IeeeI&MS_00:00:00	PTPv2
189	23.998410778	Hirschma_42:63:05	Spanning-tree-(for-b…	STP
190	24.000680772	192.168.0.4	192.168.0.255	UDP
191	24.000705548	192.168.0.4	192.168.0.255	UDP
192	24.000709969	192.168.0.4	192.168.0.255	UDP
193	24.089230825	192.168.0.4	192.168.0.255	UDP
194	24.500767776	192.168.0.4	192.168.0.255	UDP
195	24.500789342	192.168.0.4	192.168.0.255	UDP
196	24.500793776	192.168.0.4	192.168.0.255	UDP
197	24.589147374	192.168.0.4	192.168.0.255	UDP

Figure 40 : Visualization of output of switch 3 in 'wired' one

As the visualization illustrated in Figure 40, from packets 190 to 197 captured by Wireshark, 4 packets have been received can be seen from 24.000s to 24.089s, and other 4 have been packets received can be seen from 24.500s to 24.589s. The time differences between packets 190-191-192 and packets 194-195-196 are so small that the way they are transmitted can be described as 'these three packets are transmitted at the same time'. The theoretical result should not be case 3-1 but all-4, especially the delay/transmission time is set up to 300ms, the probability that switch 3 has the case 3-1 result is much smaller than the switch 2 dose. The probability is close to '0' because there is basically no opportunity that the last packet can be transmitted just at the end of the CDT interval of switch 2. After all, it is quite clear that there is something wrong because the case 3-1 appears surprisingly instead of case all-4. Only one propriety checked is the duration of cycle 0.5s.

Conclusion of the experimentation of 'wired' one example

From the results obtained of switch 2 and switch 3, they have the same case as switch 1 does, they all have case 3-1. Theoretically, the switch 2 and switch 3 outputs are supposed to be the case all-4, however, these are totally different as what is expected.

Then, the 4 possibilities that are discussed before should be checked one by one. The appearance of case 3-1 for all 3 switches is a ‘wonderful gift’ because the aim of adding delay is to make the appearance of case 3-1 for all 3 switches and this result is actually supposed to be found in ‘wrong’ one example. The possibility (1) has been be easily checked, the possibility (2) and possibility (3) are difficult to check but it is quite obvious that there is some unknown problems. Therefore, one conclusion can be made : the first example of three-switch work is really the ‘wired’ because that is a ‘wonderful gift’ out of plan.

4.3.3.Conclusion of analysis of one-switch work and three-switch work

In case of one-switch work, the Hirschmann TSN switch works correctly. Every propriety has been checked. The Hirschmann switch accumulated the packets out of the CDT interval, and the packets into CDT are transmitted, and cycle time is always well respected. However, the three-switch work brings a lot fun and shock. From the results obtained of both ‘wrong’ one example and ‘wired’ one example, when the method is correct but the result is not what it should be, and when the method is wrong but the result is the expected one for correct configuration. So to check the 4 possibilities is very important. For the reason that the possibility (1) is so easy to check, the possibilities left should be checked one by one and very carefully, because this not only represents if the TSN functions well work but aloso relates to the quality of the product. More work has to be done in order to find out the source of problem in case of three-switch.

5.Conclusion

In conclusion, this end-of-study project allowed me to deepen my technical and professional knowledge in computer networking, especially the TSN technology which has been focused on during the whole project. Some specific points have been studied from reading the paper related to TSN or from the introduction and explication made by Prof. Ye-Qiong SONG and Mr. Theo Docquier. Moreover, by doing the installation and configuration of Cisco and Hirschmann TSN switch, the theoretical knowledge has been finally transformed practical experience, and the experimentation of one-switch work and three-switch work which is supposed to establish the topology in Figure 24 helped me to understand better how the TSN works in real case. Admittedly, some of difficulties have been met during the project, for example, the configuration of Cisco switch brings a extremely time-consuming work of assembly language. But from the process of overcoming difficulties, I gained more experience, learned more knowledge, and even enjoyed the feeling of solving difficulties. All in all, under the prospect that I decided to pursue a doctor's degree, this project experience will definitely help my future career, especially in the near future I will conduct my internship at Loria.

list of the Figures

Bibliographie

Web sites :
https://hirschmann.com/en/Hirschmann_Produkte/Industrial_Ethernet/Software/Software_Tools/HiDiscovery/index.phtml
https://ostinato.org/
https://www.cisco.com/c/en/us/td/docs/switches/lan/cisco_ie4000/tsn/b_tsn_ios_support.html
https://www.cisco.com/c/en/us/td/docs/switches/lan/cisco_ie4000/tsn/b_tsn_ios_support.pdf
http://www-igm.univ-mlv.fr/~dr/XPOSE2007/vlanparlegrandquinapascomprislesconsignes/8021QTrame.html
Beckoff : https://www.beckhoff.com/EK1000/
https://vimeo.com/channels/buildingautomation/246928455
Kontron:https://www.kontron.com/products/iot/iot-industry-4.0/ethernet-tsn-switching
B&R:https://www.br-automation.com/en-gb/about-us/press-room/opc-ua-over-tsn-implementation-28-10-2019/
Renesas : https://www.renesas.com/eu/en/solutions/event/scf2017/tsn.html
SoCe : https://soc-e.com/mtsn-multiport-tsn-switch-ip-core/
ORing : https://oringnet.com/en-global/news/detail/420
Cisco :https://www.cisco.com/c/en/us/products/switches/industrial-ethernet-4000-series-switches/index.html
Hirshmann :https://hirschmann.com/en/Hirschmann_Produkte/Industrial_Ethernet/OpenRail_family_/BOBCAT-Switches/index.phtml
TTTech : https://www.tttech.com/product-filter/time-sensitive-networking-tsn/
Winsystems : https://www.winsystems.com/product/net-429/
Siemens :https://new.siemens.com/global/en/products/automation/industrial-communication/industrial-ethernet/tsn.html
NXP:https://www.nxp.com/products/interfaces/ethernet-/automotive-ethernet-switches/five-ports-avb-tsn-automotive-ethernet-switch:SJA1105TEL
Etherfly ： https://www.safe-connect-systems.com/en/etherfly-key-benefits.html
ANALOG DEVICES : https://www.analog.com/en/index.html

Publications :
[1] Jonathan Falk, David Hellmanns, Ben Carabelli, Naresh Ganesh Nayak, Frank Dürr, Stephan Kehrer, and Kurt Rothermel, 2019, Nesting : Simulating ieee time-sensitive networking (tsn) in OMNet++.
[2] Ye-Qiong Song, Dorin Maxim, 2017, Delay analysis of avb traffic in time-sensitive networks (tsn). Proceedings of RTNS '17, Grenoble, France.

Annexes

Annex 1

Startup guide

Installation of TSN on a Cisco IE 4000 switch

SU Runbo

Version 1.0

UNIVERSITÉ DE LORRAINE

Loria
Laboratoire lorrain de recherche en informatique et ses applications

cisco

TABLE OF CONTENTS

About this guide

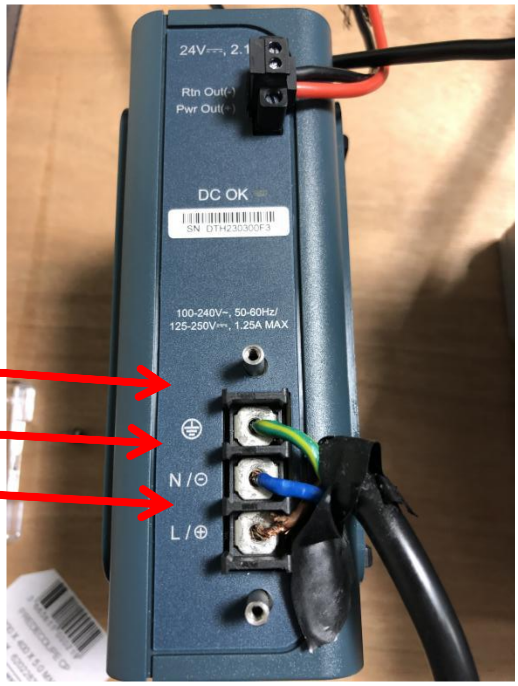
The term industry 4.0 refers to the fourth industrial revolution, real-time industrial communication must be more robust and reliable between machines and industrial plants in this revolution. One of the most widely used approaches is Time-Sensitive Networking (TSN) technology.

The purpose of this Quick Start Guide is to help you prepare and perform the initial configuration of the Cisco IE 4000 TSN switch. Some of the steps and instructions in this guide may vary based on previous configurations. For more information about installing and configuring this type of switch if you encounter difficulties, refer to the IE 4000 documentation on the Cisco.com website.

In this guide, 10.1.1.4 has been chosen as the IP address of the TSN switch and 10.1.1.2 has been chosen as the IP address of the CNC-server.

Power unit wiring

Since the TSN switch has to be alimented by DC power, first thing have to be done is the wiring of the power unit that transforms AC into DC. Just follow the next steps. Make sure that the cable plug is not connected to power in while wiring.



Step 1 :

The yellow-green (ground) is the earth.

N is neutral (blue)

L (Line) is the phase (red or brown)

Step 2 : install the protection piece and tighten the screws.

<u>**Step 3**</u> : connect the equipment to power and check DC OK lights up well.

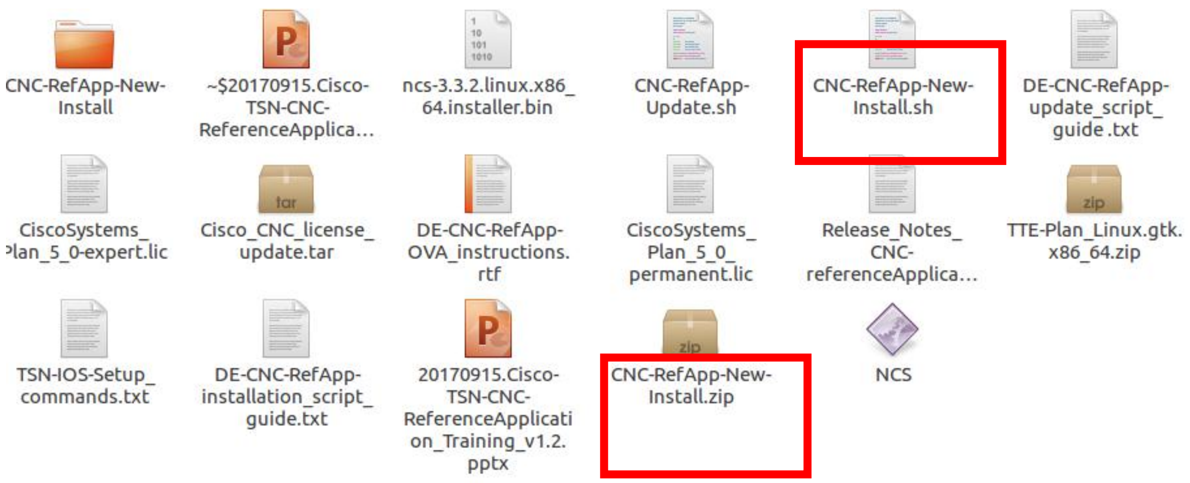
CNC interface installation (server)

To connect the TSN switch and the PC, please install the CNC interface such as a server. Be careful, this CNC interface installation requires that the Ubuntu host has Internet access to download the required files. Please follow the next steps.

To find out all the details,
please read the help file : “Cisco-TSN-CNC-ReferenceApplication_Training_v1.2”

Step 1:

Download the files "CNC-RefApp-New-Install.zip" and "CNC-RefApp-New-Install.sh", make sure that both files "CNC-RefApp-New-Install.zip" and "CNC-RefApp-New-Install.sh" are **in the same folder, and in the download folder**!



Step2:

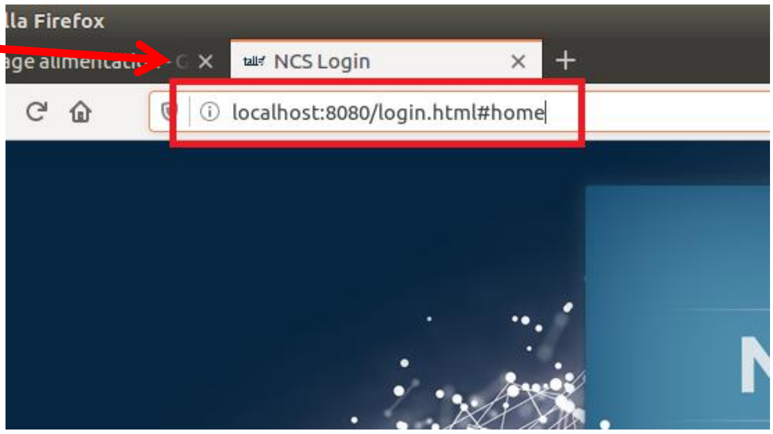
Type the following commands in Linux terminal (the terminal must be opened in the same folder):

$ chmod + x CNC-RefApp-New-Install.sh

$. / CNC-RefApp-New-Install.sh

Step 3:

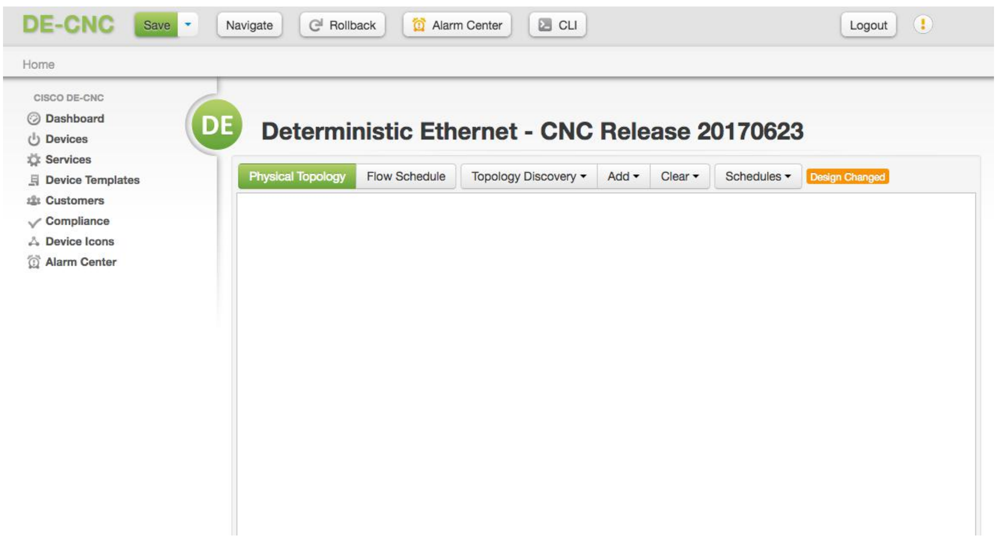
Open the CNC reference application in your own browser.
http: // localhost: 8080 / index.html.



Step 4:
Use your own Ubuntu account **username and password** to access the CNC.

Step 5:

If all goes correctly, following interface should be shown :



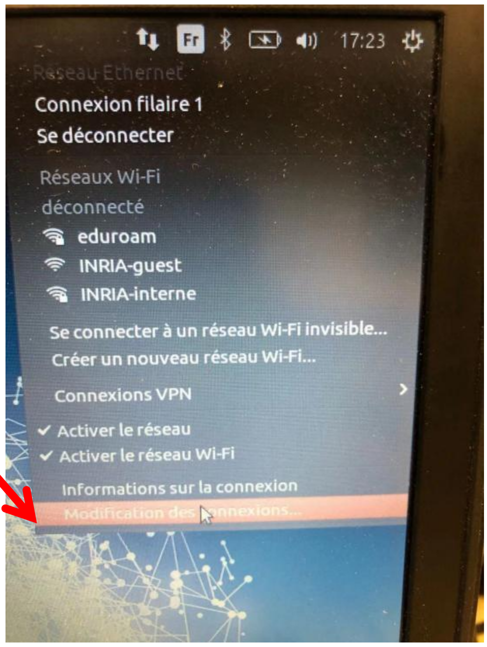
Connection to the switch ports between PC and Switch

The connection between PC and Switch can be tested after plugging the PC into an Ethernet port of the switch.

Step 1:

Go into the internet connection, a wired connection 1 is supposed to be shown in Ethernet Network.

Before click Wired Connection 1,
configuration of connections has to be changed.

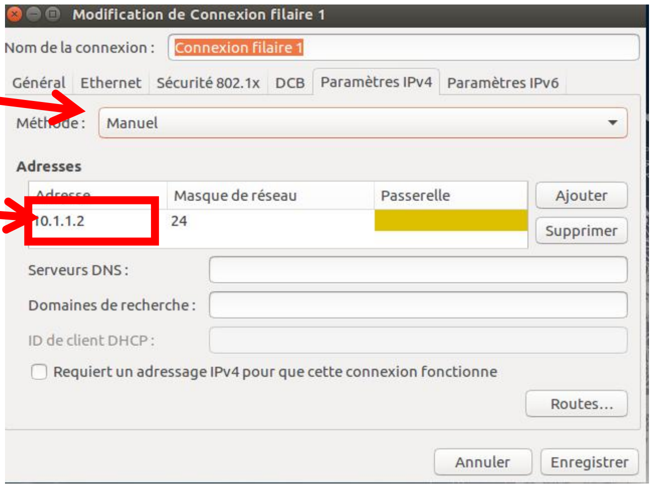


Step 2:

Change the method to 'Manual'

Enter the address chosen.

Then save this configuration.



Reminder: if this method can not configure the connection, try to change the IP address by using the 'ifconfig' command.

Step 3:

After the connection has been configured, the configuration has to be checked if it is already set up.

The available Ethernet interfaces can be viewed by using the 'ifconfig' command.

```
runbo@poney:~$ ifconfig
enp0s25   Link encap:Ethernet  HWaddr 9c:b6:54:a2:d4:bd
          inet adr:10.1.1.2  Bcast:10.1.1.255  Masque:255.255.255.0
          adr inet6: fe80::4a7c:5134:820d:1acd/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:303 erreurs:0 :0 overruns:0 frame:0
          TX packets:249 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:24569 (24.5 KB) Octets transmis:38573 (38.5 KB)
          Interruption:17 Mémoire:d4700000-d4720000
```

Here, the address is changed correctly.

Step 4:

Use 'Ping' to test the communication between PC and Switch if the IP address of the switch is 10.1.1.4 (to be checked in next section).

```
runbo@poney:~$ ping 10.1.1.4
PING 10.1.1.4 (10.1.1.4) 56(84) bytes of data.
64 bytes from 10.1.1.4: icmp_seq=2 ttl=255 time=2.70 ms
64 bytes from 10.1.1.4: icmp_seq=3 ttl=255 time=2.29 ms
64 bytes from 10.1.1.4: icmp_seq=4 ttl=255 time=0.720 ms
64 bytes from 10.1.1.4: icmp_seq=5 ttl=255 time=1.36 ms
64 bytes from 10.1.1.4: icmp_seq=6 ttl=255 time=1.23 ms
64 bytes from 10.1.1.4: icmp_seq=7 ttl=255 time=1.23 ms
64 bytes from 10.1.1.4: icmp_seq=8 ttl=255 time=5.76 ms
```

Configuration express TSN

This section will guide you through the Express TSN configuration on the Cisco IE 4000 TSN switch. Simply follow the next steps.

For complete details, please visit the site :
https://www.cisco.com/c/en/us/td/docs/switches/lan/cisco_ie4000/tsn/b_tsn_ios_support.html

Step 1:
Type the command: dmesg | grep tty to search for making the USB port be connected with the switch

```
runbo@poney:~$ dmesg | grep tty
[    0.000000] console [tty0] enabled
[  217.043225] cdc_acm 3-1.3:1.0: ttyACM0: USB ACM device
[ 7790.234453] cdc_acm 3-1.3:1.0: ttyACM0: USB ACM device
[ 8245.805729] cdc_acm 3-1.3:1.0: ttyACM0: USB ACM device
[ 8536.982237] cdc_acm 3-1.3:1.0: ttyACM0: USB ACM device
```

Here the port ttyACM0 which is activated. Normally the port is the same for any computer of yours.

Step 2:

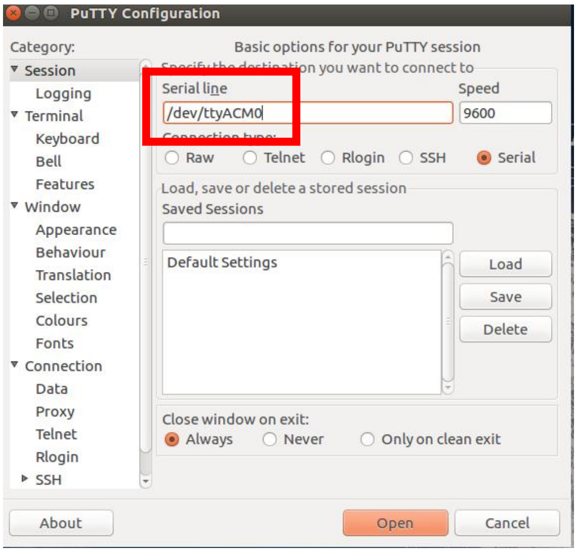
Once the enabled port is found, configuration TSN will be started via Putty.
Type the command: sudo putty (PuTTY is a terminal emulator and client for SSH, Telnet, rlogin, and raw TCP protocols, Wikipedia definition).

```
runbo@poney:~$ sudo putty
[sudo] Mot de passe de runbo :
```

the following window will be shown :

Choose 'Serial' for the type Connection and type ACM0 instead of S0 for 'Serial line'.

Then click 'Open'.



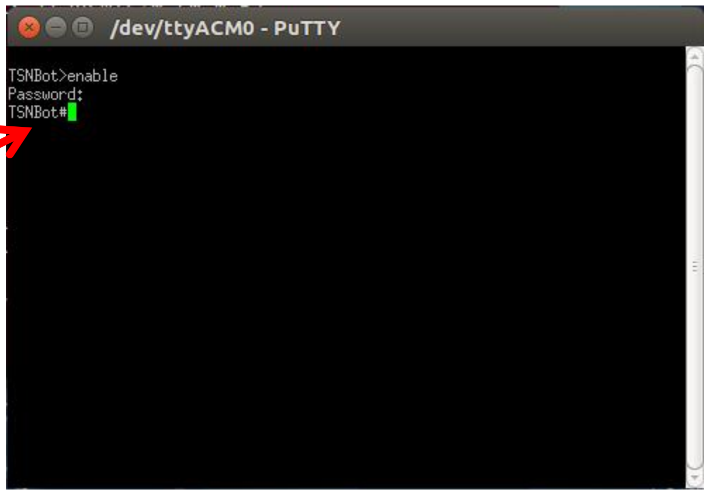
Step 3:

If everything goes correctly, the following interface should be shown(sometimes it takes a little time):

Type enable,

Then the password is : **admin**,

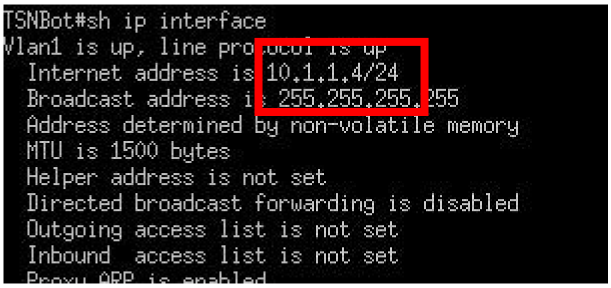
The '>' will change to '#'



Step 4:

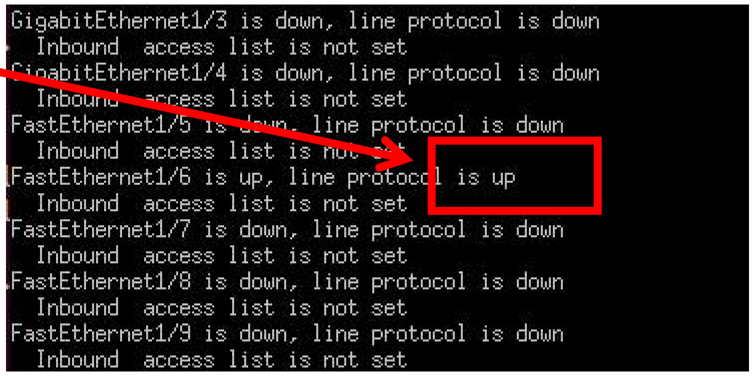
To view switch IP information, type: sh ip interface

In this example, the IP address of the switch is : 10.1.1.4



```
TSNBot#sh ip interface
Vlan1 is up, line protocol is up
  Internet address is 10.1.1.4/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
```

The enabled Ethernet port is 6



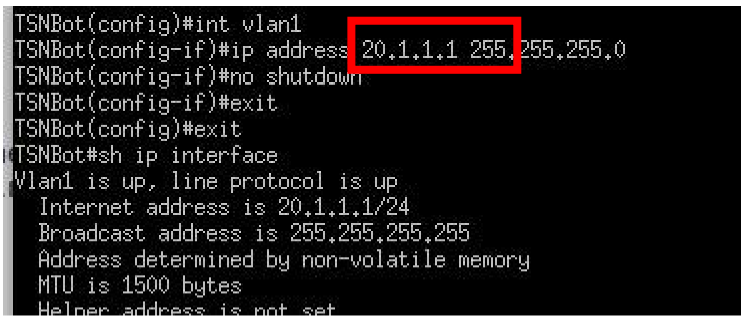
```
GigabitEthernet1/3 is down, line protocol is down
  Inbound  access list is not set
GigabitEthernet1/4 is down, line protocol is down
  Inbound  access list is not set
FastEthernet1/5 is down, line protocol is down
  Inbound  access list is not set
FastEthernet1/6 is up, line protocol is up
  Inbound  access list is not set
FastEthernet1/7 is down, line protocol is down
  Inbound  access list is not set
FastEthernet1/8 is down, line protocol is down
  Inbound  access list is not set
FastEthernet1/9 is down, line protocol is down
  Inbound  access list is not set
```

If you want to change the IP address of switch, use the following commands :

```
int vlan1
   ip address <ip address> <mask>
   no shutdown
exit
```

Example :

(Don't forget to check the implementation of the modification using : sh ip interface)



```
TSNBot(config)#int vlan1
TSNBot(config-if)#ip address 20.1.1.1 255.255.255.0
TSNBot(config-if)#no shutdown
TSNBot(config-if)#exit
TSNBot(config)#exit
TSNBot#sh ip interface
Vlan1 is up, line protocol is up
  Internet address is 20.1.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
```

Step 5:

To view the TSN CNC connection, type : sh tsn cnc or sh tsn schedule

```
TSNBot#sh tsn cnc
CNC Connection Status
----------------------------------------------------------------------
Device name:                TSNBot
CNC server:                 196.168.0.1:4569
Last CNC Contact:           16:57:12  Fri Feb 22 2019
Last Good CNC Connection:   N/A
Current Schedule ID:        0 - 0x0
----------------------------------------------------------------------

TSNBot#sh tsn schedule
Device   ID: TSNBot
Schedule ID: 0 - 0x0
Status      : Fail to program TSN schedule
   Fail reason     : Fail to communicate with CNC
   Mitigate action: Check CNC IP connectivity and re-download TSN schedule
```

In the first section, 'Last Good CNC Connection : **N/A**' means that there is no TSN CNC connection at the moment.

In the second section, 'Status : **Fail to program TSN Schedule**' and 'Fail reason : Fail to communicate with CNC', it is more obvious that the TSN CNC connection is not properly configured, this is normal because the IP address of the CNC server has not been given for the switch.

Step 6:

A CNC-server ip address should be given, so use the following command : tsn cnc-server <ip address>

Example :

```
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
TSNBot(config)#tsn cnc-server 20.1.1.2
TSNBot(config)#
```

Be careful, the ip address is not correct in the example, you have to use **10.1.1.2** chosen in the section 'Connection to the switch ports between PC and Switch'.

In the previous section, **10.1.1.2** Ethernet IP address is used in PC side, now Ping since the switch side:

```
TSNBot#ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
TSNBot#
```

Success rate is 100 percent ! Bravo !

If everything goes correctly, the communication information between switch and PC can be viewed now:

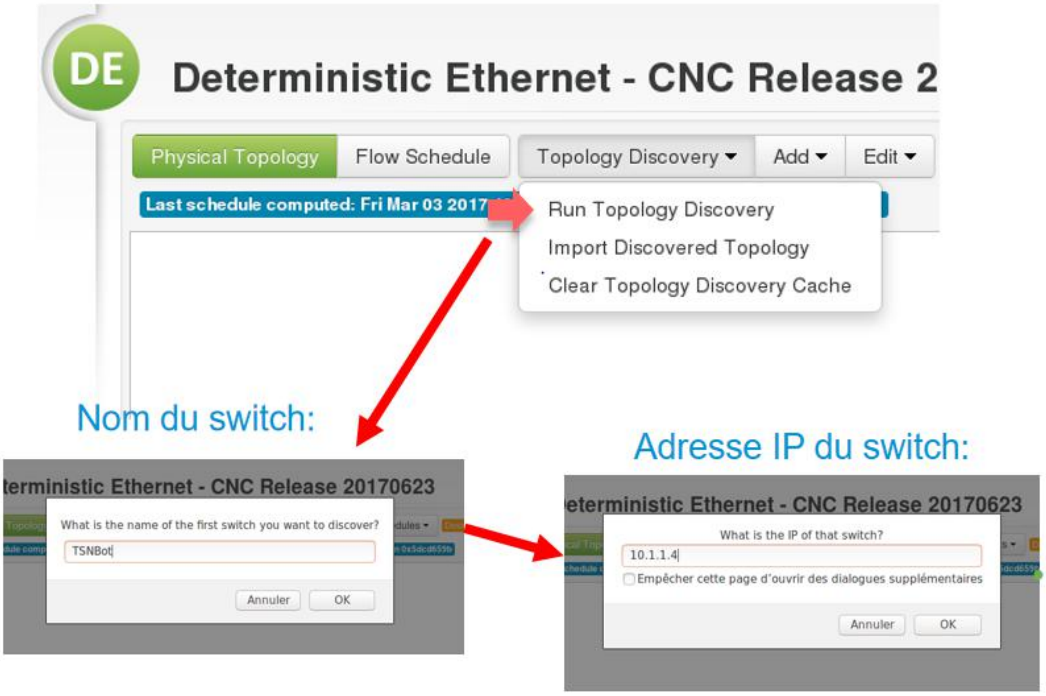
```
TSNBot#sh tsn cnc
CNC Connection Status
---------------------------------------------------------------------
Device name:                TSNBot
CNC server:                 20.1.1.2:4569
Last CNC Contact:           08:56:12  Fri Feb 22 2019
Last Good CNC Connection:   08:56:22  Fri Feb 22 2019
Current Schedule ID:        1574093485 - 0x5DD2C2AD
---------------------------------------------------------------------

TSNBot#sh tsn schedule
Device   ID: TSNBot
Schedule ID: 1574093485 - 0x5DD2C2AD
Status     : TSN schedule has been programmed successfully
```

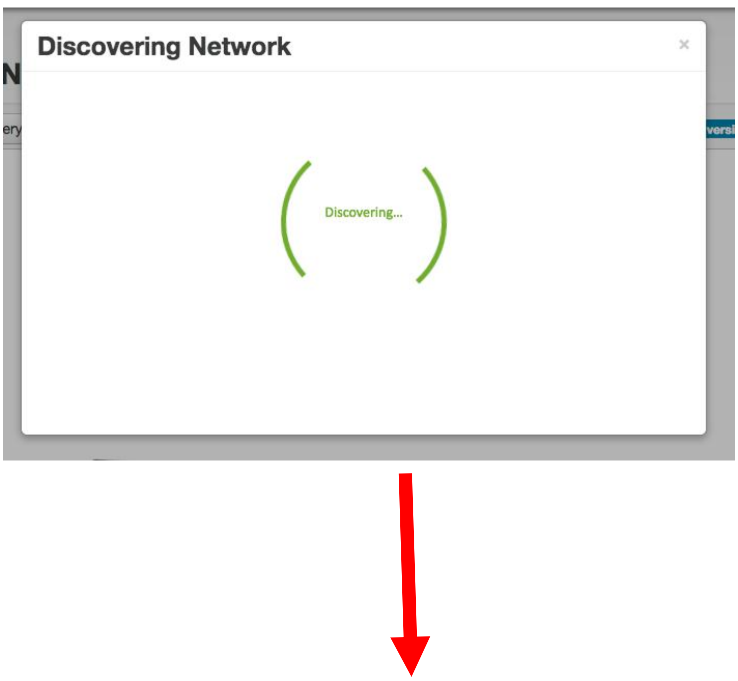
Finally, the connection between PC and Switch was successfully done.

Functions in CNC-Server

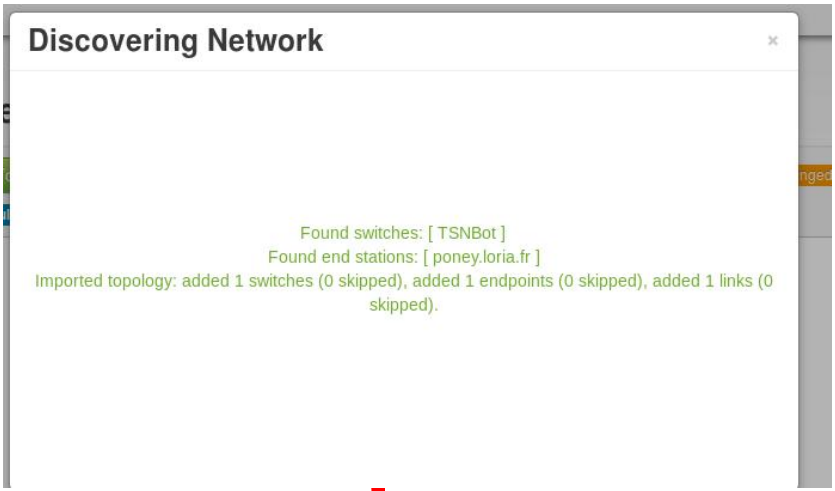
The CNC-server interface will find the switch automatically using 'Run Topology Disc'. overy'. It’s enough to enter the name of the switch and its IP address.



You are going to receive :

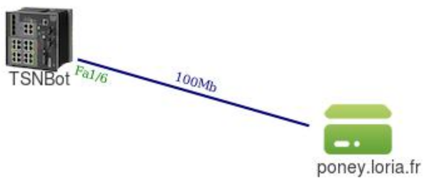


Discovering

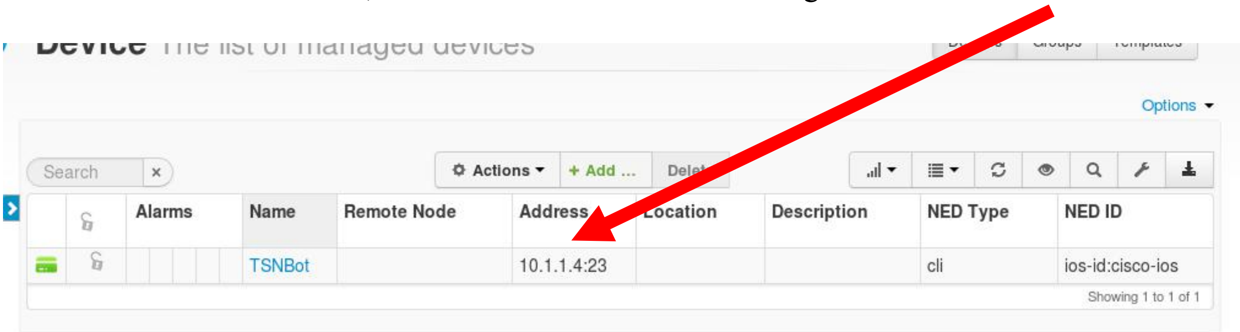


Result

Switch found



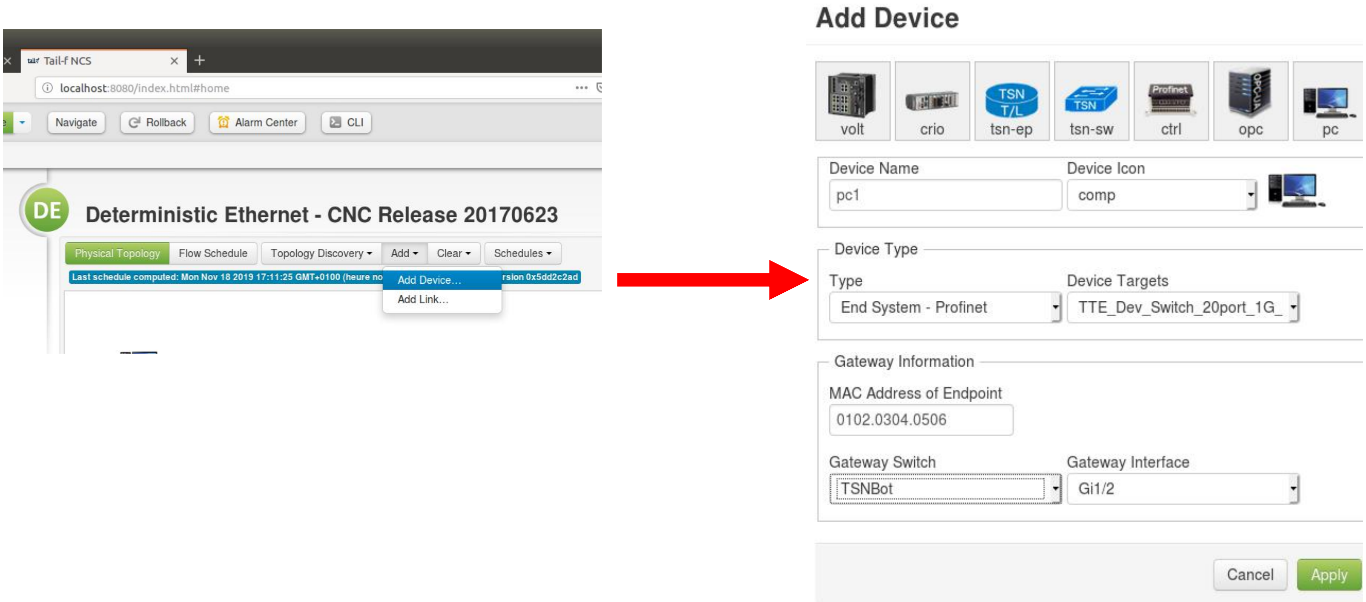
Check the section 'devices', a switch with its IP address configured will be found :



		Alarms	Name	Remote Node	Address	Location	Description	NED Type	NED ID
			TSNBot		10.1.1.4:23			cli	ios-id:cisco-ios

If you are going to add another PC, do the same thing of the section 'Connecting to the switch ports between PC and Switch' on the new PC. For example, if the CNC-server PC is 10.1.1.2, and the TSN switch is 10.1.1.4, then you can give 10.1.1.1 for the new PC.

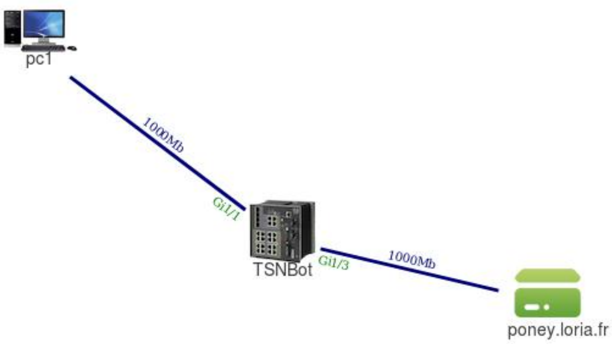
But in the CNC-Server, the PC must be added manually :



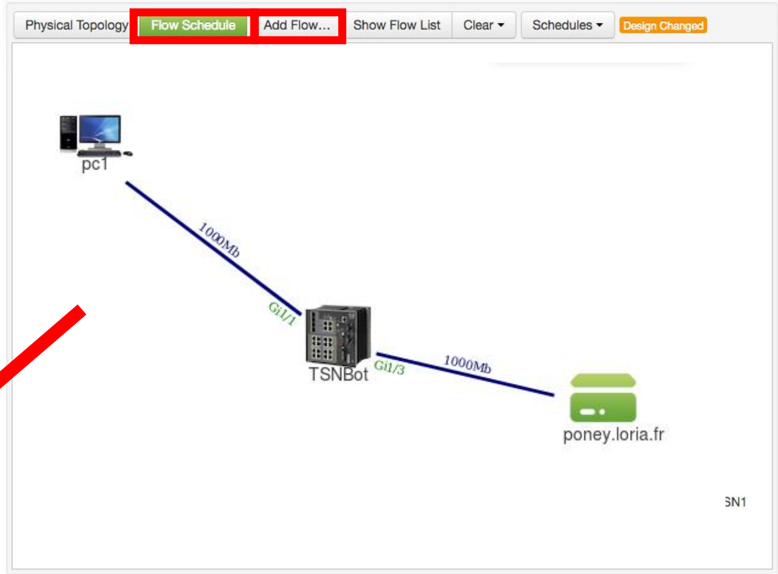
Don't forget to enter 'TSNBot' in Gateway Switch instead of nothing.

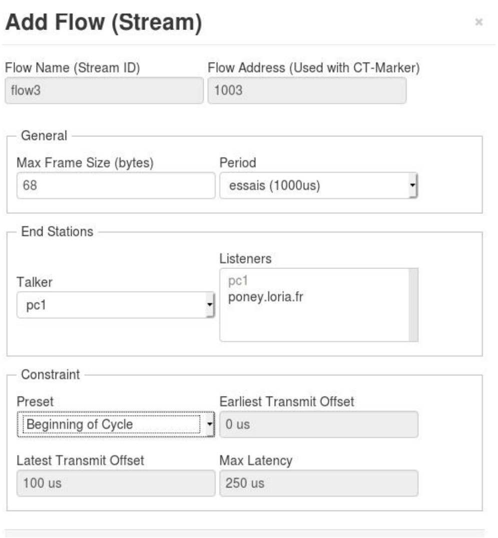
.

And then you're going to receive:

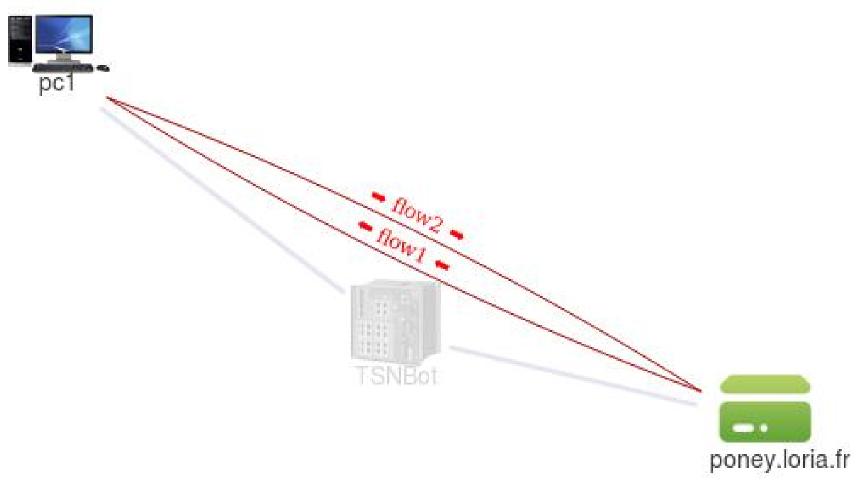


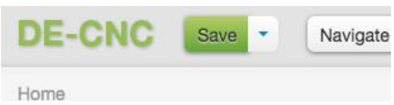
Next, flows can be created :
Click the 'Flow Schedule' button, and then ‘Add Flow’ button :





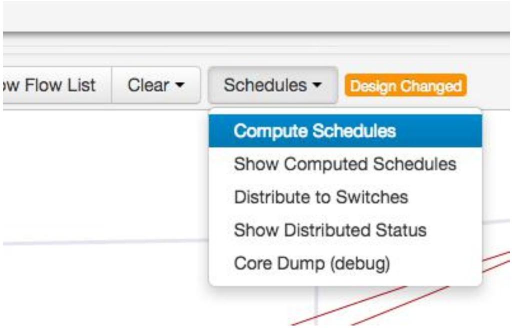
To create the streams, for example, you should firstly declare the talker and listener. In our case, streams will be sent in both directions, from PC1 to CNC-Server and from CNC-Server to PC1.





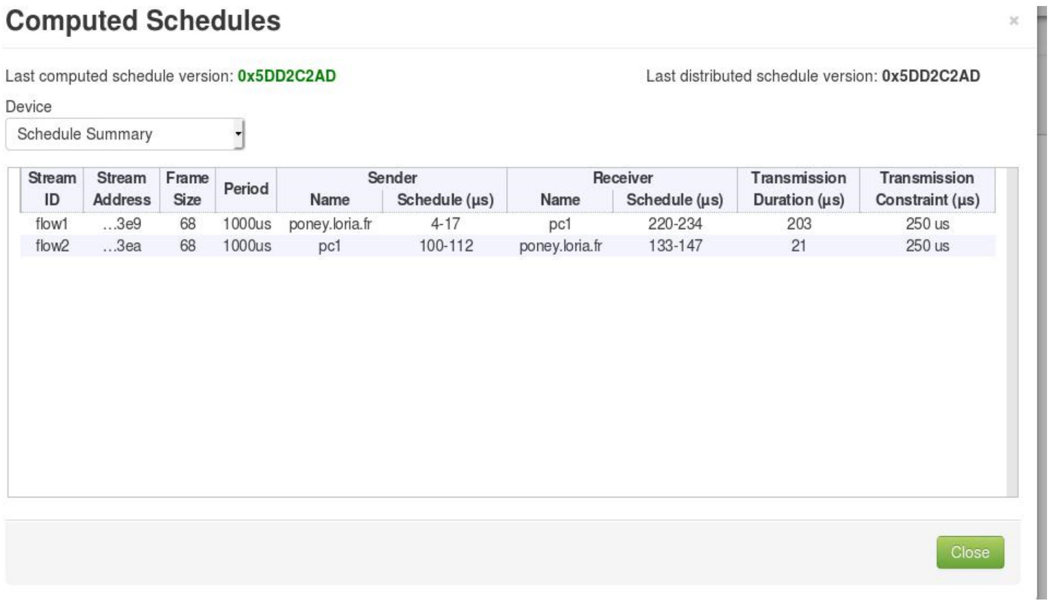
, don't forget to register your work regularly..

the Schedule can be generated : click 'Compute Schedules' button.

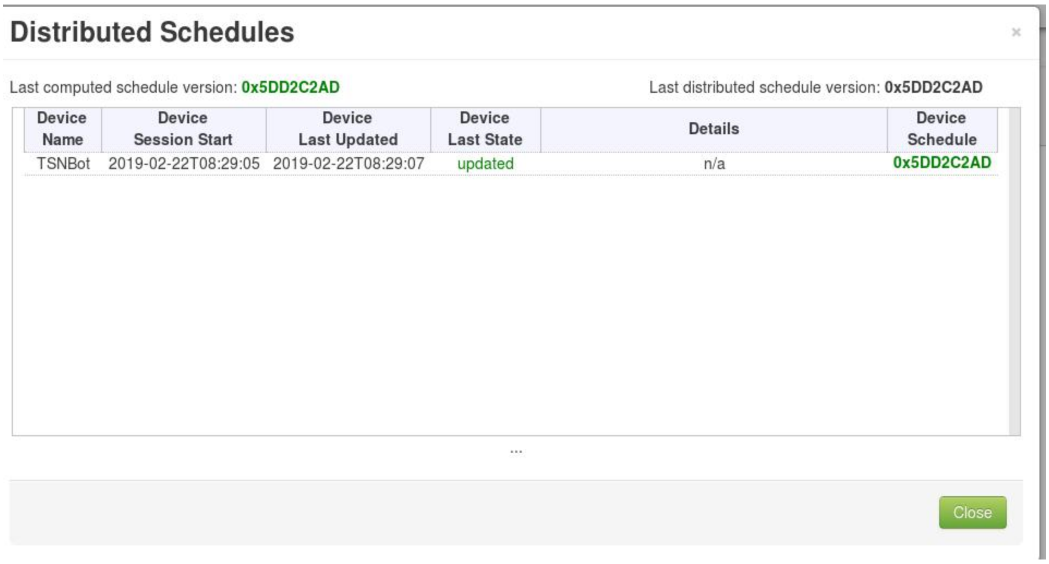


Wait a while, then you will have :

Now, you can check the schedule information :



And also some Switch information :



For all the details, ,
please read the help file “Cisco-TSN-CNC-ReferenceApplication_Training_v1.2”

Related documentation

TSN installation documentation, available at at following web pages :
https://www.cisco.com/c/en/us/td/docs/switches/lan/cisco_ie4000/tsn/b_tsn_ios_support.html
Ou
https://drive.google.com/drive/folders/1EcC6jr6mQjJTlkahG2N2iQ6ql1diowX-
Ou
https://www.cisco.com/c/en/us/td/docs/switches/lan/cisco_ie4000/tsn/b_tsn_ios_support.pdf

CNC installation documentation, available at following web pages :
https://drive.google.com/open?id=1PVbgK_lg2C3RaKbTl9ohBzDxuj0eeYVo

Documentation d’utilisation CNC-Serveur, disponible à l’adresse :
https://drive.google.com/open?id=1UmgIpa-8RV2PrURvNry05bCNZLCWaBr5

Type	Standards
Warranty	Five-year limited HW warranty on all IE-4000 PIDs and all IE Power Supplies (see table 3 above). See link below for more details on warranty
Mean time between failures (MTBF)	IE-4000-4TC4G-E: 578, 730 Hours IE-4000-8T4G-E: 591, 070 Hours IE-4000-8S4G-E: 583, 700 Hours IE-4000-4T4P4G-E: 562, 300 Hours IE-4000-16T4G-E: 558, 310 Hours IE-4000-4S8P4G-E: 535, 880 Hours IE-4000-8GT4G-E: 591, 240 Hours IE-4000-8GS4G-E: 583, 700 Hours IE-4000-4GC4GP4G-E: 550, 940 Hours IE-4000-16GT4G-E: 558, 630 Hours IE-4000-8GT8GP4G-E: 519, 190 Hours IE-4000-4GS8GP4G-E: 536, 220 Hours

Table 14. Management and standards

Description	Specification	
IEEE standards	• IEEE 802.1D MAC Bridges, STP • IEEE 802.1p Layer2 COS prioritization • IEEE 802.1q VLAN • IEEE 802.1s Multiple Spanning-Trees • IEEE 802.1w Rapid Spanning-Tree • IEEE 802.1x Port Access Authentication • IEEE 802.1AB LLDP • IEEE 802.3ad Link Aggregation (LACP) • IEEE 802.3af Power over Ethernet provides up to 15.4W DC power to each end device • IEEE 802.3at Power over Ethernet provides up to 25.5W DC power to each end device	• IEEE 802.3af Power over Ethernet • IEEE 802.3at Power over Ethernet Plus • IEEE 802.3ah 100BASE-X SMF/MMF only • IEEE 802.3x full duplex on 10BASE-T • IEEE 802.3 10BASE-T specification • IEEE 802.3u 100BASE-TX specification • IEEE 802.3ab 1000BASE-T specification • IEEE 802.3z 1000BASE-X specification • IEEE 1588v2 PTP Precision Time Protocol • IEEE 802.1AS PTP • IEEE 802.1Qbv TSN
RFC compliance	• RFC 768: UDP • RFC 783: TFTP • RFC 791: IPv4 protocol • RFC 792: ICMP • RFC 793: TCP • RFC 826: ARP • RFC 854: Telnet • RFC 951: BOOTP • RFC 959: FTP • RFC 1157: SNMPv1 • RFC 1901,1902-1907 SNMPv2 • RFC 2273-2275: SNMPv3 • RFC 2571: SNMP Management • RFC 1166: IP Addresses • RFC 1256: ICMP Router Discovery	• RFC 1305: NTP • RFC 1492: TACACS+ • RFC 1493: Bridge MIB Objects • RFC 1534: DHCP and BOOTP interoperation • RFC 1542: Bootstrap Protocol • RFC 1643: Ethernet Interface MIB • RFC 1757: RMON • RFC 2068: HTTP • RFC 2131, 2132: DHCP • RFC 2236: IGMP v2 • RFC 3376: IGMP v3 • RFC 2474: DiffServ Precedence • RFC 3046: DHCP Relay Agent Information Option • RFC 3580: 802.1x RADIUS • RFC 4250-4252 SSH Protocol
SNMP MIB objects	• BRIDGE-MIB • CALISTA-DPA-MIB • CISCO-ACCESS-ENVMON-MIB • CISCO-ADMISSION-POLICY-MIB • CISCO-AUTH-FRAMEWORK-MIB • CISCO-BRIDGE-EXT-MIB • CISCO-BULK-FILE-MIB • CISCO-CABLE-DIAG-MIB • CISCO-CALLHOME-MIB • CISCO-CAR-MIB • CISCO-CDP-MIB	• CISCO RTTMON-RTP-MIB • CISCO-SNMP-TARGET-EXT-MIB • CISCO-STACK-MIB • CISCO-STACKMAKER-MIB • CISCO-STP-EXTENSIONS-MIB • CISCO-SYSLOG-MIB • CISCO-TCP-MIB • CISCO-UDLDP-MIB • CISCO-VLAN-IFTABLE-RELATIONSHIP-MIB • CISCO-VLAN-MEMBERSHIP-MIB • CISCO-VTP-MIB

Annex 3

Product information
Compact OpenRail Fast Ethernet Switch 4-25 ports - RS20-0800M2M2SDAEHH08.0.

Industrial Ethernet:Compact configurable managed switches:with Firmware Release 8:Compact OpenRail Fast Ethernet Switch 4-25 ports http://www.e-catalog.beldensolutions.com/link/57078-24455-49854-374156-373456/en/RS20-0800M2M2SDAEHH08.0./uistate	
Name	**Compact OpenRail Fast Ethernet Switch 4-25 ports**
	
	8 port Fast-Ethernet-Switch, managed, software Layer 2 Enhanced, for DIN rail store-and-forward-switching, fanless design
Delivery informations	
Availability	not yet available
Product description	
Description	8 port Fast-Ethernet-Switch, managed, software Layer 2 Enhanced, for DIN rail store-and-forward-switching, fanless design
Port type and quantity	8 ports in total; 1. uplink: 100BASE-FX, MM-SC; 2. uplink: 100BASE-FX, MM-SC; 6 x standard 10/100 BASE TX, RJ45
Type	RS20-0800M2M2SDAEHH08.0.
Order No.	943 434-003
More Interfaces	
Power supply/signaling contact	1 x plug-in terminal block, 6-pin
V.24 interface	1 x RJ11 socket
USB interface	1 x USB to connect the AutoConfiguration Adapter ACA21-USB
Network size - length of cable	
Multimode fiber (MM) 50/125 µm	0 - 5000 m, 8 dB link budget at 1300 nm, A = 1 d/km, 3 dB reserve, B = 800 MHz x km
Multimode fiber (MM) 62.5/125 µm	0 - 4000 m, 11 dB link budget at 1300 nm, A = 1 dB/km, 3 dB reserve, B = 500 MHz x km
Network size - cascadibility	
Line - / star topology	any
Ring structure (HIPER-Ring) quantity switches	50 (reconfiguration time < 0.3 sec.)
Power requirements	
Operating voltage	12/24/48 V DC (9,6-60) V and 24 V AC (18-30) V (redundant)
Current consumption at 24 V DC	321 mA
Current consumption at 48 V DC	161 mA
Power output in Btu (IT) h	26.3
Software	
Management	Serial interface, web interface, SNMP V1/V2, HiVision file transfer SW HTTP/TFTP
Diagnostics	LEDs, log-file, relay contact, RMON, port mirroring 1:1, topology discovery 802.1AB, address conflict detection, network error detection, SFP diagnostic [temperature, optical input and output power (µW and dBm)], Trap for configuration saving and changing, duplex mismatch detection, disable learning.
Configuration	Comand line interface (CLI), TELNET, BootP, DHCP, DHCP option 82, HIDiscovery, easy device exchange with auto-configuration adapter ACA21-USB (automatic software and/or configuration upload), automatic invalid configuration undo, configuration signature (water marking)
Security	Port Security (IP und MAC) with multiple addresses, SNMP V3 (no encryption)
Redundancy functions	HIPER-ring (ring structure), MRP (IEC-ring functionality), RSTP 802.1D-2004, redundant network/ring coupling, MRP and RSTP in parallel, redundant 24 V power supply
Filter	QoS 4 classes, port priorisation (IEEE 802.1D/p), VLAN (IEEE 802.1Q), shared VLAN learning, multicast (IGMP Snooping/Querier), multicast detection unknown multicast, broadcastlimiter, fast aging
Industrial Profiles	EtherNet/IP and PROFINET (2.2 PDEV, GSDML stand-alone generator, automatic device exchange) profiles included, configuration and diagnostic via automation software tools like e.g. STEP7, or Control Logix
Flow control	Flow control 802.3x, port priority 802.1D/p, priority (TOS/DIFFSERV)
Presettings	Standard
Ambient conditions	
Operating temperature	0 °C ... 60 °C
Storage/transport temperature	-40 °C ... 70 °C
Relative humidity (non-condensing)	10 % ... 95 %
MTBF	53.5 years (MIL-HDBK-217F)

HIRSCHMANN
A BELDEN BRAND

Industrial Ethernet:Compact configurable managed switches:with Firmware Release 8:Compact OpenRail Fast Ethernet Switch 4-25 ports http://www.e-catalog.beldensolutions.com/link/57078-24455-49854-374156-373456/en/RS20-0800M2M2SDAEHH08.0./uistate	
Protective paint on PCB	No
Mechanical construction	
Dimensions (W x H x D)	74 mm x 131 mm x 111 mm
Mounting	DIN Rail
Weight	410 g
Protection class	IP20
Mechanical stability	
IEC 60068-2-27 shock	15 g, 11 ms duration, 18 shocks
IEC 60068-2-6 vibration	1 mm, 2 Hz-13.2 Hz, 90 min.; 0.7 g, 13.2 Hz-100 Hz, 90 min.; 3.5 mm, 3 Hz-9 Hz, 10 cycles, 1 octave/min.; 1 g, 9 Hz-150 Hz, 10 cycles, 1 octave/min
EMC interference immunity	
EN 61000-4-2 electrostatic discharge (ESD)	6 kV contact discharge, 8 kV air discharge
EN 61000-4-3 electromagnetic field	10 V/m (80-1000 MHz)
EN 61000-4-4 fast transients (burst)	2 kV power line, 1 kV data line
EN 61000-4-5 surge voltage	power line: 2 kV (line/earth), 1 kV (line/line), 1 kV data line
EN 61000-4-6 conducted immunity	3 V (10 kHz-150 kHz), 10 V (150 kHz-80 MHz)
EMC emitted immunity	
FCC CFR47 Part 15	FCC 47 CFR Part 15 Class A
EN 55022	EN 55022 Class A
Approvals	
Safety of industrial control equipment	cUL 508
Hazardous locations	ISA 12.12.01 Class 1 Div. 2
Shipbuilding	n/a
Railway norm	n/a
Substation	n/a
Scope of delivery and accessories	
Scope of delivery	Device, terminal block, operating manual

For more information please contact:
Hirschmann Automation and Control GmbH
Stuttgarter Strasse 45-51
72654 Neckartenzlingen
Germany
Phone: +49 7127/14-1809
E-Mail: inet-sales@belden.com

- Ingress rate-limiting on a per-port and per-priority basis for Unicast/Multicast and Broadcast traffic
- Frame replication and retagging of traffic
- Frame mirroring for enhanced diagnostics
- Hardware support for IEEE 802.1AS and IEEE 802.1Qav for AVB traffic support
- Ingress and egress timestamping per port
- Ten IEEE 802.1Qav credit-based shapers available; shapers can be freely allocated to any priority queue on a per port basis
- Support for AVB SR Class A, Class B and Class C traffic
- IEEE 1588v2 one-step sync forwarding in hardware
- IEEE 802.1X support for setting port reachability and disabling address learning
- Broadcast storm protection
- Statistics for dropped frames and buffer load

2.3 TT and TSN features (SJA1105TEL only)

- IEEE 802.1Qbv time-aware traffic
- IEEE 802.1Qci per-stream policing (pre-standard)
- Support for ring-based redundancy (for time-triggered traffic only)
- 1024 deterministic Ethernet flows with per-flow based:
  - Time-triggered traffic transmission
  - Ingress policing and reception window check
  - Active and redundant routes
  - Statistics

2.4 Interface features

- MII/RMII interfaces supporting all standard Ethernet PHY technologies such as (but not limited to) Fast Ethernet (IEEE 100BASE-TX), IEEE 100BASE-T1 and optical PHYs
- RGMII for interfacing with Gigabit Ethernet (1000BASE-T) PHYs (Gigabit Ethernet; Ref. 4)
- MAC and PHY modes for interfacing (MII/RMII/RGMII) directly with another switch or host processor
- Programmable drive strength for all interfaces
- SPI at up to 25 MHz for host processor access

2.5 Other features

- 25 MHz system clock input from crystal oscillator or AC-coupled single-ended clock
- 25 MHz reference clock output
- Device reset input from host processor
- IEEE 1149.1 compliant JTAG interface for TAP controller access and boundary scan

Annex 5

TECHNICAL INFORMATION

SYSTEM		KBox C-102-2 industrial computer with PCIe-0400-TSN network card Realtime Linux + Tools
PROCESSOR		Intel® Core™ i5-6442EQ processor: Quad-Core 1.9 GHz, 4 GByte RAM, 128 GByte SSD
TSN NETWORK CARD		PCIe, 4 Ethernet 10/100/1000Mbps ports
SWITCH OPERATION		Cut through, StoreAndForward
SUPPORTED TSN STANDARDS IEEE-802.1 TSN		- 802.1 as(rev) timing and synchronization - 802.1 Qbv traffic scheduling - 802.1 Qbu frame preemption - 802.1 Qcc Stream Reservation Protocol enhancements (in progress) - 802.1 CB, Qci, ... N/A – will be provided per update
SOFTWARE SUPPORT	OPERATING SYSTEM APPLICATIONS	- RT Linux Operating System - BIOS with RT patches - TSN NIC driver - TSN switch driver - PTP4 Linux stack (IEEE802.1AS) - Startup scripts - TSN TX sample application - TSN RX monitoring application - Ostinato Network traffic generator - TSN schedule config. tool
POWER		24 VDC direct input, 230 VAC / 24 VDC adaptor included
DIMENSIONS		290 x 155 x 210 mm (11.42" x 6.1" x 8.27")
ENVIRONMENTAL	OPERATING TEMPERATURE STORAGE TEMPERATURE CLIMATIC HUMIDITY	0 °C to +60 °C -40 °C to +85 °C (-40 °F to 185 °F) 93 % RH at 40 °C, non-condensing (acc. to IEC 60068-2-78)
RoHS COMPLIANT		yes

ORDERING INFORMATION

ARTICLE	DESCRIPTION
KBOX C-102-2 TSN STARTER KIT (DUO)	Dual KBox C-102-2 including PCIe-0400-TSN IEEE-802.1 TSN network interface card and cables. Ready-to-go with preinstalled Realtime Linux and tools

Global Headquarters

Kontron S&T AG

Lise-Meitner-Str. 3-5
86156 Augsburg, Germany
Tel.: + 49 821 4086 0
Fax: + 49 821 4086 111
info@kontron.com
www.kontron.com

Ethernet TSN Layer 2 Switch IP Datasheet

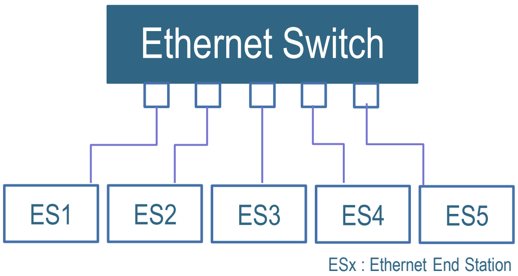
Ethernet TSN Layer 2 Switch

Overview

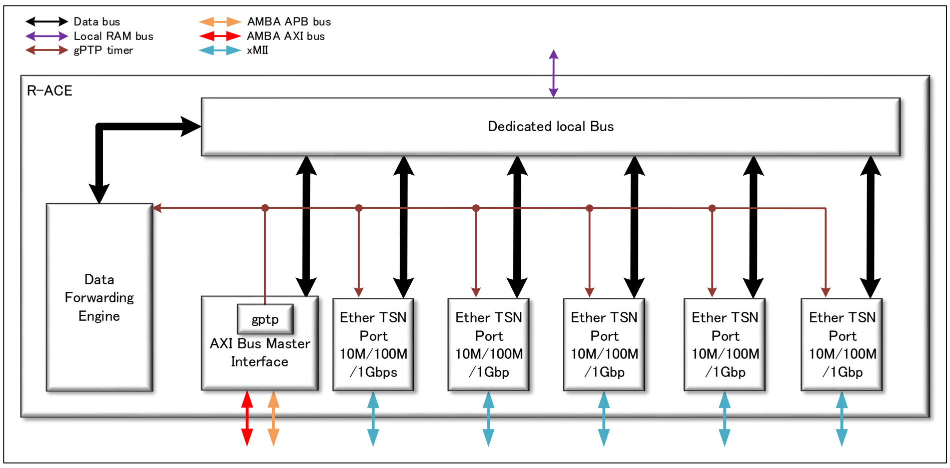
- The Renesas Ethernet TSN Layer 2 Switch IP provides interface features for Layer 2 switch(5ports) in Ethernet Time Sensitive Networking.
- The Ethernet TSN switch using dedicated low latency bus.
  Ether port to port transfer, CPU - Ether port to port transfer, broadcast transfer, multicast transfer are also possible. Various transfers featuring advanced filters are possible.
- This IP includes an Ethernet controller that conforms to the IEEE 802.3 standard, and standards related Ethernet TSN. This IP has standard and easy interface. You can easily implement this IP to your system.

Key Features

1. Ethernet Frame data routing / Forwarding using Various transfers featuring advanced filters
2. Integrated Ethernet Communication Controller (Ethernet MAC) according to standard IEEE 802.3 into each Ether port
3. The controller support these operation modes - 10/100/1000 Mbps - full-duplex operation
4. PHY Interface : MII, RMII,GMII, RGMII and SGMII (SGMII is Standard Ver1.8 and using External Converter)
5. Support Receive filtering (for L2~L4)
6. Support Jumbo Frame(up to 16KB)
7. Support IEEE 802.1AS-rev
8. Support IEEE 802.1Qci Ingress Policing
9. Support IEEE 802.1Qav Credit-based shaper
10. Support IEEE 802.1Qbv Time Aware Shaper
11. Support IEEE 802.3br Interspersed Express Traffic
12. Support IEEE 802.1Qbu Frame Preemption
13. Automatic Tx/Rx CRC computed by Ethernet Controller
14. Interrupt-free data transfer between system memory and network
15. Integrated DMA for data transfers from/to system memory
16. Integrated FIFO in Ethernet controller
17. Test Modes Internal loopback



Block diagram



1. **5 Ethernet port (Increase / decrease possible)**
2. **AXI Master 64 - 128 bits data bus**
3. **Built-in gPTP**
4. **Routing Table :**
   1. **Qci**
   2. **SRP**
   3. **MAC**
   4. **VLAN**

Annex 6

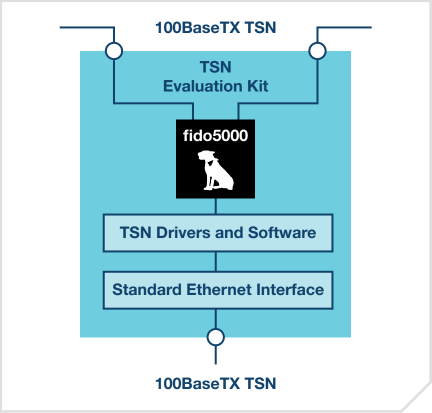
ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

TSN EVALUATION KIT

Evaluate TSN Features with Any Standard Ethernet Device

IEEE 802.1 TSN

The TSN Evaluation Kit is a complete TSN gateway solution that allows any standard 100BaseTX Ethernet device to connect to a TSN network



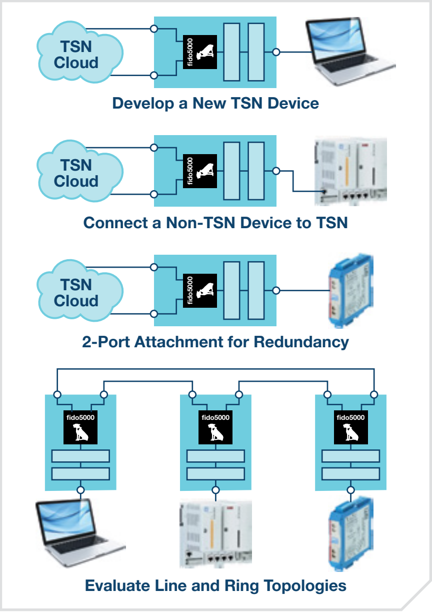
The TSN Evaluation Kit contains everything needed to evaluate the features of the emerging IEEE 802.1 Time Sensitive Networking (TSN) standards. The kit is pre-installed with the most mature TSN features starting with 802.1AS and 802.1Qbv. As the TSN standards evolve, new features and updates will be made available as free downloads through our Developer Portal. The kit is designed to support the following TSN standards:

- 802.1AS, AS-REV—Time synchronization
- 802.1Qbv—Scheduled traffic
- 802.1Qci—Ingress policing
- 802.1CB—Seamless redundancy
- 802.1Qcc—Stream reservation protocol
- 802.1Qbu/802.3br—Preemption

More on these TSN standards is available at *deterministicethernet.com*.

Flexible and Easy to Use

The Evaluation Kit acts as a gateway between a standard 100BaseTX Ethernet port and two 100BaseTX TSN ports. The ability to use two TSN ports greatly expands the use of TSN by enabling evaluation of any network topology including star, redundant star, lines, and rings. At the heart of the kit is the embedded fido5000 real-time Ethernet multiprotocol (REM) programmable 3-port switch chip that manages TSN functionality.



As new features are released or when changes to the TSN standards are announced, firmware for the fido5000 REM switch is programmed using the downloads from the Developer Portal. This unique reprogramming capability is important as TSN features continue to evolve through the standards process over time.

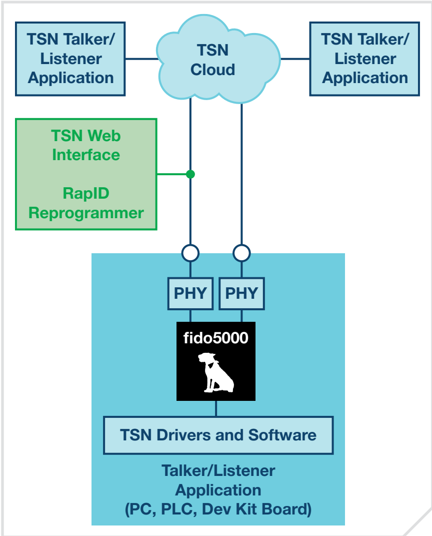
The kit is designed to enable you to quickly connect any existing Ethernet device, or even a laptop, via the standard Ethernet port. Then you configure the gateway to use the TSN features you wish to evaluate. This configuration allows your non-TSN application to exchange TSN datastreams with TSN talkers or listeners on the TSN network. You can also evaluate highly reliable, deterministic operation on a converged network, even in the presence of other data streams and best effort traffic. Evaluation of full redundancy can also be achieved—ensuring a guarantee of service under any network conditions.

A PC-based tool is provided in the kit that supports selection of the desired TSN standards. A second PC-based tool is provided that supports configuration of specific features for the selected TSN standards.

Migration to TSN-Compliant Device

The TSN Evaluation Kit also serves as a reference design where elements of the platform's hardware and software can be integrated directly into your new product. On the hardware side, the fido5000 REM switch along with any suitable PHY can be integrated with your application's host processor. On the software side, your application software interfaces to the APIs in the TSN drivers and software. You now have a complete solution that can connect to any TSN network in any topology.

As the TSN standards emerge, your applications can keep up with the evolving standards. The kit is the perfect vehicle for testing and development. The kit is already being used as part of the Industrial Internet Consortium's TSN Testbed. Please see, *iiconsortium.org/time-sensitive-networks.htm*.



Where to Find What You Need

The TSN Evaluation Kit is available from Analog Devices' distribution network.

Order your kit now at *innovasic.com/products/tsn-kit*. All of the software and the User's Guide is available through our Developer Portal. Login to the Developer Portal and go to the section for the TSN Evaluation Kit. Everything is available as a free download.

TSN Evaluation Kit Contents

The TSN Evaluation Kit consists of hardware and software. The hardware items are delivered in a box containing the following items:

- 2-port TSN Ethernet module
- Standard Ethernet module
- Baseboard with power supply
- RJ-45 Ethernet cables

The software items are available via download on the Innovasic website in the Developer Portal:

- TSN drivers and software
- TSN Software User's Guide

Updates to the drivers, software, and documentation will be made available on an ongoing basis. Users will be notified when updates are uploaded to the Developer Portal.

Analog Devices, Inc.
Worldwide Headquarters

Analog Devices, Inc.
One Technology Way
P.O. Box 9106
Norwood, MA 02062-9106
U.S.A.
Tel: 781.329.4700
(800.262.5643, U.S.A. only)
Fax: 781.461.3113

Analog Devices, Inc.
Europe Headquarters

Analog Devices GmbH
Otl-Aicher-Str. 60-64
80807 München
Germany
Tel: 49.89.76903.0
Fax: 49.89.76903.157

Analog Devices, Inc.
Japan Headquarters

Analog Devices, KK
New Pier Takeshiba
South Tower Building
1-16-1 Kaigan, Minato-ku,
Tokyo, 105-6891
Japan
Tel: 813.5402.8200
Fax: 813.5402.1064

Analog Devices, Inc.
Asia Pacific Headquarters

Analog Devices
5F, Sandhill Plaza
2290 Zuchongzhi Road
Zhangjiang Hi-Tech Park
Pudong New District
Shanghai, China 201203
Tel: 86.21.2320.8000
Fax: 86.21.2320.8222

BR15888-.3-5/17

analog.com

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

WINSYSTEMS®

NET-429

Industrial TSN and 1588v2-Enabled Network Switch with PoE-PD Support, 8x RJ45 plus 2x SFP Ports

Features

Performance for Industrial IoT Edge Computing

- High performance fabric for simultaneous switching on all ports
- Advanced Quality of Service (QoS) features to provide guaranteed delivery of time sensitive data
  - IEEE 802.1 Time Sensitive Networking (TSN)
  - IEEE1588v2 one-step Precision Time Protocol (PTP)
  - IEEE 802.1 Audio Video Bridging (AVB)
  - IEEE 802.1Qbb Priority Flow Control
- Eight 10/100/1000 Mbps RJ45 Ethernet ports
- Two 1000Base-X SGMII SFP ports

Integrated Security and Management Features

- 256 Entry TCAM for Deep Packet Inspection
- Supports 4096 802.1Q VLANs
- Supports 802.1D/s/w Spanning Tree Protocols
- IPv4 IGMP and IPv6 MLD snooping
- 802.1x MAC Authentication
- Three levels of 802.1Q security supported

Powered for Remote Deployment

- Redundant Wide Range 9-36V DC Inputs
- Power over Ethernet (PoE) PD 802.3at Type1 device
- Energy Efficient Ethernet PHYs reduce power consumption
- Low power with Typical Power Consumption of 6 Watts

Designed to survive Industrial Control Applications

- Fanless -40°C to +85°C Operating Temperature Range
- Optional DIN Rail Mountable
- Shock and Vibration Tested
- Long lifetime (10+years availability)

Product Description

WINSYSTEMS' NET-429 network switch is designed for the harsh environments of the factory floor and provides performance for time critical industrial networks. The switch has eight 10/100/1000 Mbps RJ45 Ethernet ports plus two 1000Base-X SGMII SFP ports and redundant power inputs with Power-over-Ethernet (PoE) PD support. Enabled for the latest IEEE 802.1 standards for Quality of Service (QoS) and Time Sensitive Networking (TSN), it includes advanced prioritization and timing features to provide guaranteed delivery of time-sensitive data.

The NET-429 is based on the Marvell® Link Street® family, which provides advanced Quality of Service (QoS) features with 8 egress queues. The high-performance switch fabric provides line rate switching on all ports simultaneously while providing advanced switch functionality. It also supports the latest IEEE 802.1 Audio Video Bridging (AVB) and Time Sensitive Networking (TSN) standards. These new standards overcome the latency and bandwidth limitations of Ethernet to allow for the efficient transmission of real-time content for industrial applications. The AVB/TSN protocols enable timing sensitive streams (such as digital video, audio, or industrial control traffic) to be sent over the Ethernet network with low latency and robust QoS guarantees.

Easing the burden of deployment, the NET-429 can be powered remotely as a PoE-PD Type I device or through the two wide range 9-36V DC power inputs. All three power inputs are redundant for maximum uptime and include overload protection to prevent damage to other systems.

WINSYSTEMS also combines design elements of our broad portfolio for application specific design for OEM clients. Contact an Application Engineer to schedule a consultation of your product requirements.

IGS-RX164GP+

Industrial advanced Layer 3 20-port managed Gigabit Ethernet switch with 16x10/100/1000Base-T(X) ports and 4x1G/10GBase-X, SFP+ socket

Features

- Support routing protocols – Static routing, RIP v1/v2, OSPF, PIM-SM, PIM-DM, VRRP
- Support TSN freature - IEEE 802.1AS for timing & Synchronization, Qav, Qat
- Support **O-Ring** (recovery time < 30ms) and MSTP(RSTP/STP compatible) for Ethernet Redundancy
- **O-Chain** allow multiple redundant network rings
- Provided HTTPS/SSH protocol to enhance network security
- Support SNTP client
- Support application-based QoS management
- Support DOS/DDOS auto prevention
- IGMP v2/v3 (IGMP snooping support) for filtering multicast traffic
- Support SNMP v1/v2c/v3 & RMON & 802.1Q VLAN Network Management
- Support ACL and 802.1x User Authentication for security
- Support 10K Bytes Jumbo Frame
- Multiple notification for warning of unexpected event
- Web-based ,Telnet, Console (CLI), and Windows utility (**Open-Vision**) configuration
- Rigid IP-30 housing design
- DIN-Rail and wall mounting enabled

MFN 100

Edge Computing Device

Key Benefits

- Resource virtualization
- Data connectivity
- Open Interoperability
- Centralized management

MFN 100 is an edge computing device, designed for harsh industrial environments (-40 °C to +70 °C). The device is based on an Intel Atom x5-E3940/50 CPU and offers 4GB/8GB RAM and up to 512GB SSD storage. MFN 100 supports fieldbus connectivity and also integrates a Time Sensitive Networking (TSN) switch for open, standard deterministic Ethernet communication. MFN 100 devices are part of the Nerve platform which integrates fogOS™ and fogSM™ software from Nebbiolo Technologies in a scalable fog computing architecture.

Resource virtualization

Computational resources of MFN 100 can be securely shared between different operating systems and applications enabling the convergence of various functionality (e.g. industrial PC, gateway, PLC and firewall) in one device.

Data connectivity

Nerve enables data to be used in three ways: at the edge, on your local server or remotely in a cloud. This gives users the flexibility to choose where and when data is stored, visualized and processed.

Open Interoperability

Nerve provides openness, interoperability and flexibility. The platform helps users to become vendor independent, giving the freedom to choose the solutions, from any provider, which best suit specific business needs.

Centralized management

fogOS and fogSM combine to put installed MFN 100 devices at the users fingertips. Software updates to machines can be applied without needing to be on site. Central management of device software and applications delivers new operational flexibility and reduces support and maintenance efforts.

Application Fields

- Manufacturing
- Industrial Automation
- Industrial Robotics

Product features

CPU	Intel E3940 4 cores, 1.8 GHz 4GB RAM		Intel E3950 4 cores, 8GB RAM
Storage	64GB SSD MLC	256GB SSD MLC	512GB SSD MLC
Interfaces	1 x 1000 Mbit/s SFP 3 x 100/1000 Mbit/s RJ45 1 x 100/1000Mbit/s Ethernet Console Port RJ45 2 x USB 2.0 1 x Display Port		
System software	fogOS™ from Nebbiolo Technologies - Hosts docker containers or binaries - Hosts Customer OS (Windows and Linux) - Interface to fogSM™ System Manager for secure application deployment - Role-based access control - Built-in firewall - Data path with visualization, time-series database and analytics framework		
Operating system hosting	Customer OS (Windows and Linux) hosted as Virtual Machine		
System manager	fogSM™ from Nebbiolo Technologies - On premise or hosted cloud service available - Remote management of multiple devices - Remote software lifecycle management - Remote screen viewing - Remote status viewing		
Connect button	Connection to fogSM™ can be manually controlled at the device. Indicator is lit when a connection to the fogSM™ is established.		
Integrated soft PLC	*CODESYS 3.5 (1ms cycle time, PROFINET Master)		
ProtocolsSupport	OPC/UA, MQTT, REST API		
Housing	DinRail or wall mount Dimensions (H x W x D) 177.5 x 75 x 141mm		
Power	2 x 24V redundant input Maximum power consumption 33.6W		
Environmental	Intended for use in control cabinets Operational temperature: -40°C to +70 °C IP 40 according to IEC 60529 Shock & vibration		

* indicates optional extra

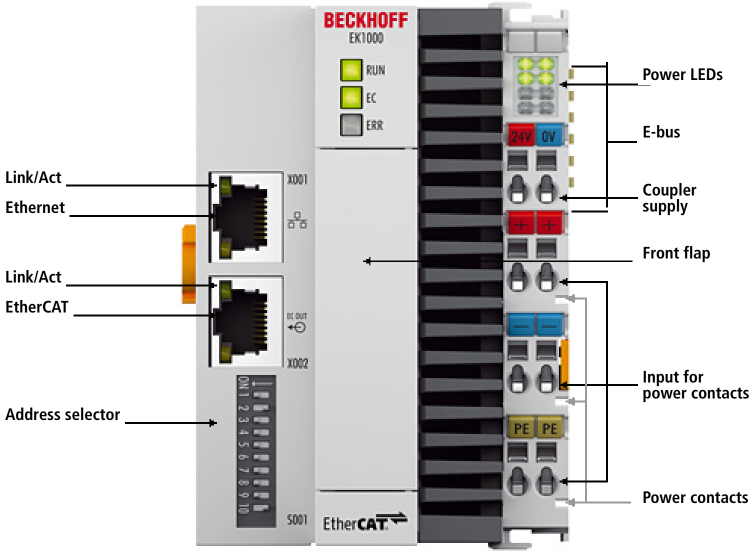
TTTech Europe, Austria (Headquarters)
Phone: +43 1 585 34 34-0

TTTech North America Inc.
Phone: +1 978 933-7979

TTTech Japan
Phone: +81 52 485-5898

TTTech China
Phone: +86 21 5015 2925-0

products@tttech.com www.tttech.com



EK1000 | EtherCAT TSN Coupler

The EK1000 EtherCAT Coupler supports EtherCAT communication via a switched Ethernet network, combining TSN (Time-Sensitive Networking) with the wide I/O range of EtherCAT Terminals. The EK1000 EtherCAT Coupler also enables EtherCAT communication in standard Ethernet networks. In addition, proven EtherCAT features such as distributed clocks and XFC can also be used within a TSN network. In this way, the EK1000 effectively links EtherCAT and TSN networks and combines their benefits: EtherCAT provides a real-time capable single-device level and is therefore ideally suited to combine the numerous small data packets of digital and analog inputs that are typical for industrial environments into a complete process image. TSN, on the other hand, defines a physical layer, which only relates to the switch layer in switched Ethernet networks. As a real-time extension of the general Ethernet technology, it controls the procedures and priorities for the transport of Ethernet telegrams through a TSN-capable, switched network. The EtherCAT Coupler has two RJ45 ports. One of these 100 Mbit/s ports connects the EK1000 to the Ethernet or TSN network. The second port enables EtherCAT network extensions, e.g. via the EK1100 EtherCAT Coupler. The EK1000 extends the TwinCAT automation software with a TSN-capable version of EtherCAT communication. In this way, TSN networks can also be managed seamlessly, and EtherCAT devices located in different terminal strands can be addressed via the TSN network.

Technical data	EK1000
Task within EtherCAT system	coupling of EtherCAT Terminals (ELxxxx) to TSN networks
Networks	mode 1: for unmanaged switch networks; mode 2: for TSN networks
Data transfer medium	Ethernet/EtherCAT cable (min. Cat. 5), shielded
Distance between stations	max. 100 m (100BASE-TX)
Number of EtherCAT Terminals	up to 65,534
Type/number of peripheral signals	–
Protocol	EtherCAT
Delay	–
Data transfer rates	100 Mbit/s
Configuration	not required
Bus interface	X001 (RJ45) = 100 Mbit/s Ethernet (TSN); X002 (RJ45) = 100 Mbit/s EtherCAT (EK1100 e.g.)
Power supply	24 V DC (-15 %/+20 %)
Current consumption from U_S	–
Current consumption from U_P	–
Current supply E-bus	2000 mA
Power contacts	max. 24 V DC/max. 10 A
Electrical isolation	500 V (power contact/supply voltage/Ethernet)
Operating/storage temperature	0…+55 °C/-25…+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protect. class/installation pos.	IP 20/variable
Approvals/markings	CE

Accessories	
Cordsets	cordsets and connectors

System	
EtherCAT	For further EtherCAT products please see the system overview

Product announcement	estimated market release on request