Configuration of Wireless TSN Networks

Kofi Atta Nsiah, Kamil Alkhouri and Axel Sikora

October 15, 2020

The 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems
17-18 September, 2020, Dortmund, Germany

Configuration of Wireless TSN Networks

Kofi Atta Nsiah[1], Kamil Alkhouri[2], Axel Sikora[1,2]
[1] Hahn-Schickard Association of Applied Research e.V., Wilhelm-Schickard-Str. 10, 78052 Villingen-Schwenningen, Germany, {kofi.nsiah, axel.sikora}@hahn-schickard.de, https://www.hahn-schickard.de/
[2]Institute of Reliable Embedded Systems and Communication Electronics (ivESK), Offenburg University of Applied Sciences, Badstrasse 24, 77652 Offenburg, Germany, {kamil.alkhouri, axel.sikora}@hs-offenburg.de, https://ivesk.hs-offenburg.de/

***Abstract*—Time Sensitive Networking (TSN) provides mechanisms to enable deterministic and real-time networking in industrial networks. Configuration of these mechanisms is key to fully deploy and integrate TSN in the networks. The IEEE 802.1 Qcc standard has proposed different configuration models to implement a TSN configuration. Up until now, TSN and its configuration have been explored mostly for Ethernet-based industrial networks. However, they are still considered “work-in-progress” for wireless networks. This work focuses on the fully centralized model and describes a generic concept to enable the configuration of TSN mechanisms in wireless industrial networks. To this end, a configuration entity is implemented to configure the wireless end stations to satisfy their requirements. The proposed solution is then validated with the Digital Enhanced Cordless Telecommunication ultra-low energy (DECT ULE) wireless communication protocol.**

Keywords—wireless TSN; CUC; CNC; NETCONF; YANG model; IEEE 802.1 Qcc; DECT ULE

I. Introduction

In today’s industrial networks, providing support for the strict real-time requirements of industrial applications are becoming increasingly important. This holds true for both wired and wireless networks. Time Sensitive Networking (TSN) defines set of standards [1] that enables determinism and real-time behavior particularly for standard IEEE 802 Ethernet networks. Even though TSN offers mechanisms to support deterministic and real-time communication, configuration of the mechanisms in the deployed networks is key to achieve the desired performance. Key elements required in this context include the configuration of communication streams in the network and configuration of the end devices.[1]

The IEEE 802.1 Qcc [2] proposes three models for the configuration of TSN networks. They include fully centralized model, fully distributed model and hybrid model. Hybrid model combines both centralized and distributed approaches. Fully centralized model (Fig. 1) defines a centralized user configuration (CUC), which

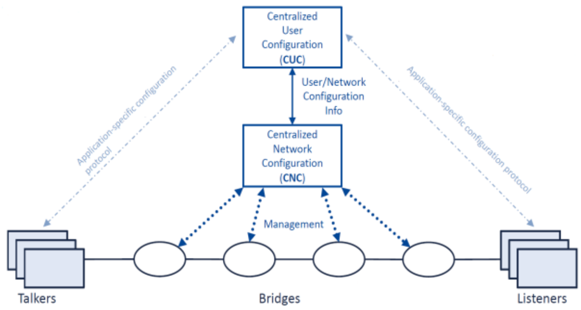

Figure 1. Fully Centralized Configuration Model.

communicates the requirements and capabilities of the end stations (i.e. talkers and listeners) to the central network configuration (CNC) via the User Network Interface (UNI). CUC is additionally responsible for configuring the end stations over an application specific configuration protocol. CNC on the other hand has full knowledge of the available network resources and topology information. Based on the requirements received from CUC, the CNC computes the schedule and configures the network elements. Configuration information of end stations is communicated back to CUC, which then configures them. Compared with the fully centralized model, the fully distributed configuration model works completely without a dedicated configuration entity. Although this model is highly dynamic, the extent and scope of configuration are limited to the individual devices. The hybrid model leverages the advantages of both models.

This work focuses on the fully centralized model and implements the CUC entity to configure wireless end stations of the DECT ULE [3] [4] wireless technology, referred to as portable parts. Additionally, a user-only protocol for the communication between CUC and end stations is implemented and open source libraries are employed to create the UNI interface between CUC and CNC. Till the time of writing this paper, no open-source CNC implementation or complete CNC specification could be found. Therefore, functionalities of the CNC such as configuration of network devices and processing of stream

[1] This work was partially funded within TSN-CPS project at Hahn-Schickard by the Federal Ministry of Education and Research.

reservation requests from the CUC are not covered in this work.

The remainder of the paper is organized as follows: Section II presents a review of the recent works for TSN configuration in Ethernet networks. Section III, presents the approach for the configuration of TSN in wireless systems. Section IV describes the implementation details. Finally, results and analysis are presented in Section V.

II. State of the Art

Some works have been done to analyze the configuration and management of real-time industrial networks. The authors in [5] propose a configuration scheme based on the fully centralized model and the OPC UA protocol. Their results show that the proposed scheme can configure Ethernet-based industrial networks to meet the delay requirements of periodic time-sensitive data flows. As part of the research done in [6], the authors propose a configuration agent architecture based on the fully centralized model and OPC UA protocol to perform runtime network configuration and reconfiguration. In [7], the authors focus on the self-configuration of real-time networks by proposing a complete self-configuration framework for Ethernet-based TSN networks based on a configuration agent. The agent continuously monitors the network to detect changes and automatically updates the configuration to maintain the desired Quality of Service (QoS). Their work shows that by adding learning capabilities to TSN and communication protocols like OPC-UA and NETCONF, self-configuration of TSN features can be enabled. The authors in [8] extend the fully centralized model to develop a dynamic configuration model. As part of the model, they develop a central entity that detects topology changes due to link or node failures and reconfigures the TSN devices using the NETCONF protocol.

Summarizing the contributions from the related works, it is clear that most of the works are focused on TSN configuration in Ethernet-based networks. This paper presents a first-time approach to describe the configuration of TSN features in wireless-based networks.

III. Configuration of TSN in Wireless Networks

A. *Overview of TSN Configuration*

The following elements are building blocks for the general configuration of TSN

- User/Network Interface (UNI): UNI is the interface over which configuration information is exchanged between the CUC and CNC. Talkers and listeners are located in the end stations and represent the user side of the UNI interface. Bridges on the other hand represent the network side of the interface. CNC configures the bridges based on the end station requests in order to meet their requirements.
- Modelling of user/network configuration information: Network management protocols such as NETCONF [9] and SNMP are available to exchange configuration information over the UNI interface. NETCONF uses the YANG data model [10] and encodes configuration data in XML/JSON format. SNMP uses MIB modules as its data model and ASN.1 BER as its encoding format. Compared to SNMP, NETCONF scales better for larger networks and automatically discovers YANG modules on the devices. Due to its key advantages, NETCONF is selected for the implementation in this work.
- TSN configuration models: With the fully distributed model, CUC and CNC entities are absent. Configuration information is exchanged between end stations and network elements as well as between the network elements, both over UNI interfaces. In the hybrid model, CNC is introduced into the architecture to configure the network elements. The fully centralized model has both CUC and CNC entities and only a single UNI interface is located between these entities. Compared with the other two models, the fully centralized model allows more flexibility in the design of the CUC and UNI interfaces, since the UNI interfaces are not hardware specific to the end stations and network elements. In principle, CNC and CUC used for the configuration of Ethernet networks can be used for configuring wireless networks. The only requirement is that the network management protocol used must be compatible with the wireless system.

B. *Configuration of TSN in DECT ULE*

Digital Enhanced Cordless Telecommunication (DECT) ultra-low energy (ULE) is a wireless communication protocol that operates in the 1880 – 1900 MHz band. It features a TDMA system with a frame size of 10ms split into 24 time slots available in 10 different frequency channels. DECT ULE has two major devices – DECT Fixed Part (FP) represents the base station and the DECT Portable Part (PP) represents end devices equipped with a DECT ULE modem. Out of the 24 time slots, timeslots (1 - 12) are for downlink transmission from the base station to the devices while timeslots (13 - 24) are for uplink transmissions. This means one base station can support simultaneous transmissions from 12 PPs.

TSN networks are generally characterized by a TSN backbone that connects the wired devices. Wireless devices can also be connected using the TSN backbone. As shown in Fig. 2, the TSN backbone provides connectivity between the FPs and the CNC. Since the FPs have full knowledge of the capabilities of the connected PPs, each FP can forward the requirements of each device to the CNC. Thus, CUC is implemented on the FP to interact with the CNC over the TSN backbone. This connection is identical to the UNI interface between the CUC and CNC.

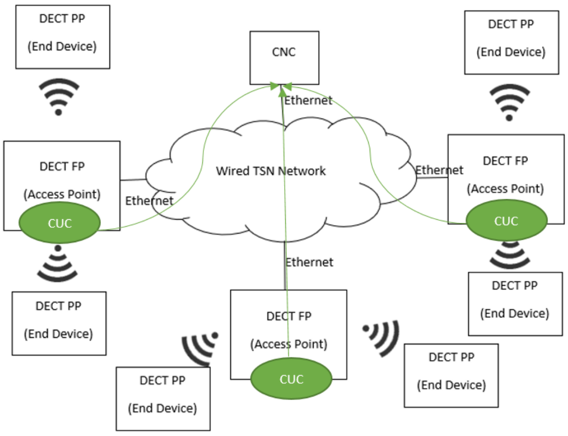

Figure 2. Configuration of TSN features in a DECT ULE wireless system

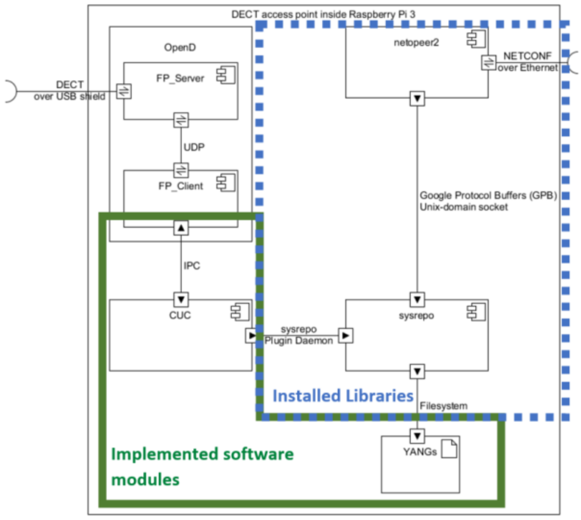

Figure 3. Software components implemented in the DECT ULE fixed part.

IV. Implementation of Wireless TSN Configuration

This section describes the implementation details and set up of TSN configuration of the DECT ULE wireless system. The CUC software module and libraries required to configure the DECT ULE wireless system are realized in the FP. In fact, all the implementation is realized in the FP as shown in Fig. 3. On one side, the FP connects to the PP over the DECT ULE air interface and on the other side, it connects to the CNC via the NETCONF over Ethernet interface. The different functional blocks that are required for the implementation include DECT ULE system, NETCONF/YANG and CUC.

A. *Software Components*

The DECT ULE functionality is enabled in the PP and FP by configuring and setting up them with the open-source openD framework. The framework integrates a DECT ULE stack for the core DECT ULE functionalities and HAN FUN library that provides core services such as device management, device information and attribute reporting. Within the FP, the framework configures two components, a server component "FP_Server" that communicates directly with the PP and a client component "FP_client" that allows integration of user applications. These two components communicate over a UDP interface.

As described in previous sections, the UNI in this implementation is based on NETCONF/YANG. This functionality is realized with two open source libraries – SYSREPO and NETOPEER2. SYSREPO is a configuration and operational state data store based on the YANG model. It ensures data consistency and enforces constraints specified by the YANG model. Configuration data and state data are processed differently by SYSREPO. While configuration data is configurable by CUC, state data can only be modified by the CNC. NETOPEER2 has components, a server component and client component. The server component implemented a network configuration management based on the NETCONF protocol and request data from the SYSREPO data store. The client is a command line interface that connects to the NETCONF server. In order to integrate the NETOPEER2/SYSREPO combination into the system, a SYSREPO plugin is implemented between them and the CUC implementation to subscribe to the relevant YANG modules. In this way, state data and configuration data can be provided and read from the CUC respectively.

The data structure of the configuration information for the UNI is implemented as a YANG module. Thus, a new module that holds all managed information of the CUC is specified and described. In addition, the list of all the identities of the PP (PMID) as shown in Fig. 4 is added to the module in order to allow the CNC to set the paths between the respective FP accordingly.

B. *Hardware Platform*

The FP hardware components consist of a Raspberry Pi 3 and a USB dongle from DSP group. USB dongle provides the RF interface for the communication with the PP. The PP runs on a NUCLEO-L476RG development board for the integration of user applications and a DECT shield based on DSP groups for the core DECT ULE functionalities.

C. *Implementation of CUC*

For the full functionality of the CUC, new software components are implemented, openD framework is modified while the SYSREPO and netopeer2 libraries are integrated. The new software modules that are implemented in this work include the CUC user interface and SYSREPO plugin. CUC user interface provides the CUC functionalities and the SYSREPO plugin acts as a plugin to the SYSREPO engine to handle the modules in the YANG repository. In addition, it creates and allocates the shared memory, which is used to exchange data amongst all the software components running on the FP as independent processes as shown in Fig. 5.

```
container devices-list{
    list device{
        config false;
        key "id";
        description
            "List of the end stations connected to the CUC
            This list is only maintained by the CUC"
            leaf "id" {
                type uint8_t;
                description
                    "unique id to identify the device in this module"
            }

            leaf "name" {
                type string;
                description
                "Given name for the DECT PP"
            }
            leaf "pmid" {
                type pmid-type;
                description
                    "20 Bit Portable part MAC IDentity"
            }
        }
}
```

Figure 4. Snapshot of YANG module of the PP devices list.

V. Results and Analysis

The interaction between the different software components is shown in Fig. 7. CUC user interface is the main application that the user can interact with to execute several operations such as configure PPs as talkers or listeners, list all configured talkers, list all configured listeners, list all streams etc. For some of these operations, the CUC user interface retrieves the data from the shared memory. The results show that the user can interact with the CUC user interface to send requests to the CNC. Based on positive feedback received from the CNC, the user configures the PPs. Fig. 6 shows the output of a configured PP as a talker with the parameters that are derived from the YANG module.

VI. Conclusion

Time Sensitive Networking (TSN) is expected to play a key role in providing real-time communication in wireless networks. Configuration of the TSN mechanisms is key to satisfy the requirements of the time-sensitive applications.

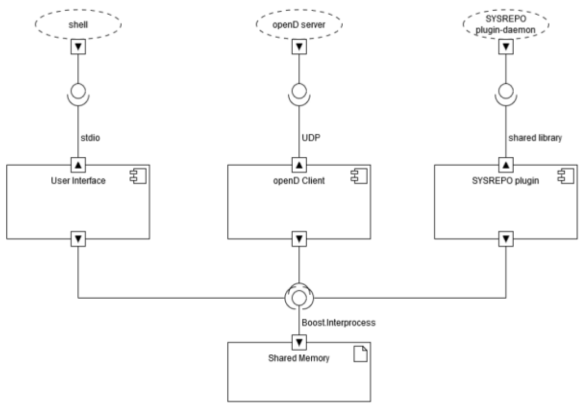

Figure 5. Component Diagram showing the different software components of the CUC.

```
Number of registered talkers: 2

Device IDs registered as talkers:
    0: Device ID 1 with PMID 11110000000000001110
        1: Device ID 2 with PMID 11110100000000001111

Ener entry number to list data or (q) to quit
0

!---- PRINTING TALKER DATA -----!
    talker-id: 1
        stream-rank:
            rank: 1
        end-station-interfaces:
            0
                name: eth0
                address: AA-AA-AA-AA-AA-AA
            1
                name: dect0
                address: CC-CC-CC-CC-CC-CC
    data-frame-specification:
        index: 0
            ieee802-vlan-tag
                pcp: 4
                vlan_id: 88
        index: 1
            ieee802-mac-addresses
                source_mac_address: AA-AA-AA-AA-AA-AA
                destination_mac_address: BB-BB-BB-BB-BB-BB
    traffic-specification:
        interval: 1/2
        max-frames-per-interval: 20
        max-frame-size: 1024
        transmission-selection: 1234
        time-aware:
            earliest transmission offset: 1
            latest-transmit-offset: 10
            jitter: 5
```

Figure 6. CUC user interface list showing PP configured as a talker.

In this work, CUC is implemented to configure the DECT ULE portable parts as a proof of concept. Even though IEEE 802.1 Qcc standard specifies the data structure between the CUC and CNC, there are still open issues such as complete specification definition of the CNC, complete UNI interface definition and the CUC/CNC request and response of stream reservation operations. In the future, the plan is to extend the proposed solution to implement the stream reservation requests from CUC to CNC. In addition, as soon as CNC specifications are available, wired CNC implementation will be ported to wireless systems.

Acknowledgment

This work was partially supported by S. Schrade and executed as part of his master thesis at the Albert-Ludwig-University of Freiburg, being hosted at ivESK.

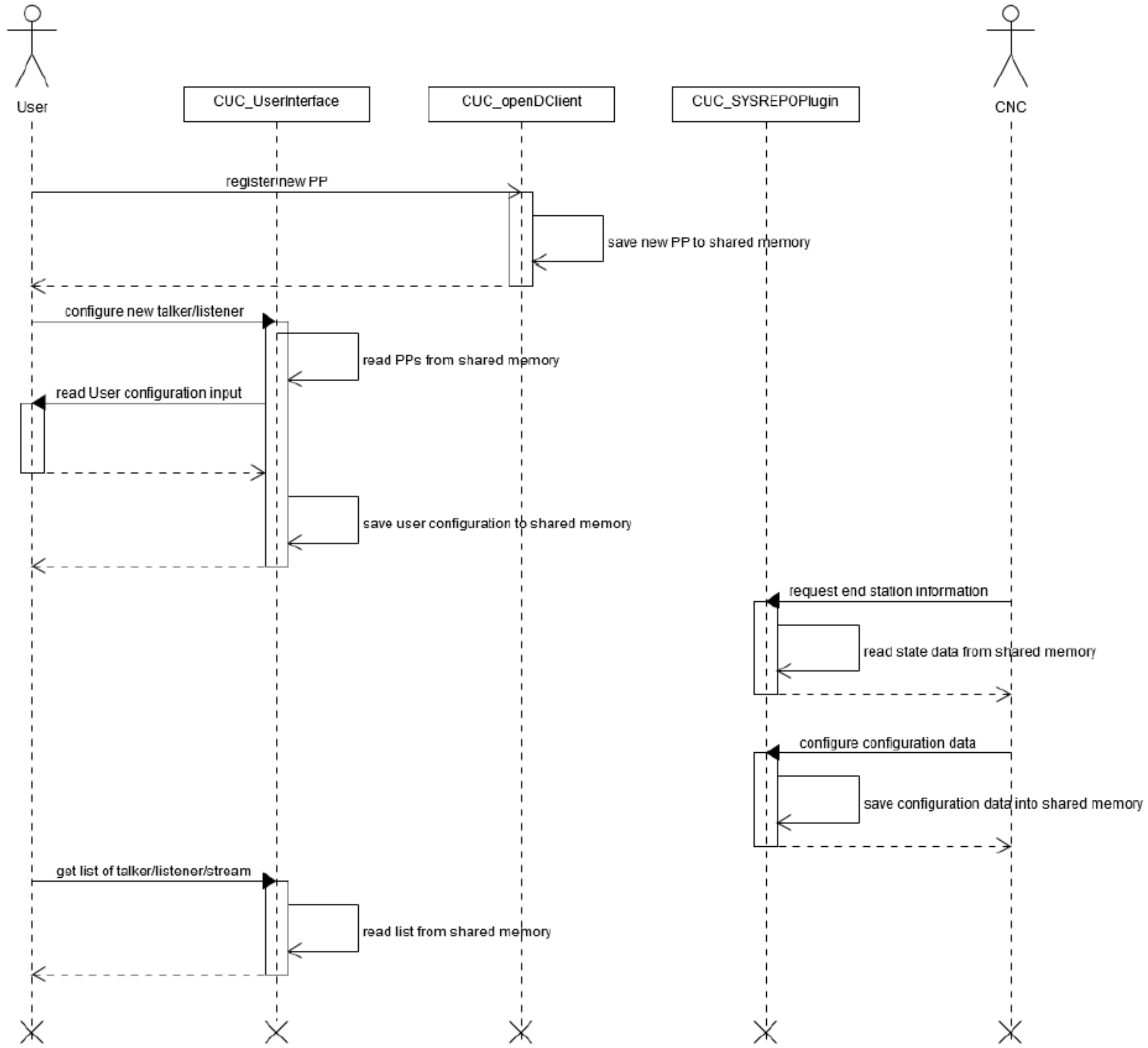

Figure 7. Sequence Diagram showing the interactions between the different software components of the CUC.

REFERENCES

[1] Institute of Electrical and Electronics Engineers, Inc, "Time-Sensitive Networking Task Group, "http://www.ieee802.org /1/pages/tsn.html". Accessed on June 1, 2020

[2] IEEE (2018): IEEE Std 802.1Qcc - Standard for Local and Metropolitan Area Networks. Bridges and Bridged Networks. Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements.

[3] Digital Enhanced Cordless Telecommunications (DECT), Common Inteiface (CI); Part 1-5, ETSI EN 300 175-1 to 5, v2.3.4 (2015).

[4] K. Das, P. Havinga, "Evaluation of DECT ULE for robust communication in dense wireless sensor networks", (2012), 3rd IEEE International Conference on Internet of Things.

[5] Z. Zhou, G. Shou, "An Efficient Configuration Scheme of OPC UA TSN in Industrial Internet", (2019), IEEE Conference Paper, Chinese Automation Congress (CAC), pp. 1548 – 1551.

[6] P. Pop, M. L. Raagaard, M. Gutierrez, W. Steiner, "Enabling Fog Computing for Industrial Automation Through Time-Sensitive Networking (TSN)", (2018), IEEE Communications Standards Magazine, Volume 2 Issue 2, pp. 55 – 61.

[7] A. A. Atallah, G. B. Hamad, O. A. Mohamed, "Self-configuration of IEEE 802.1 TSN networks", (2017), 22nd IEEE Conference on Emerging Technologies and Factory Automation (ETFA) 2017.

[8] M. Pahlevan, J. Schmeck, R. Obermaisser, "Evaluation of TSN Dynamic Configuration Model for Safety-Critical Applications" (2019), IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom).

[9] RFC6241 - Network Configuration Protocol (NETCONF). Available online at https://tools.ietf.org/html/rfc6241. Accessed on June 4, 2020.

[10] IEEE 802.1Qcw YANG Data Models for Scheduled Traffic, FramePreemption, and Per-Stream Filtering and Policing. Available online at https://1.ieee802.org/tsn/802-1qcw. Accessed on June 2, 2020.