



HAL
open science

Enabling programmable deterministic communications in 6G

Minh-Thuyen Thi, Siwar Ben Hadj Said, Adrien Roberty, Fadlallah Chbib,
Rida Khatoun, Leonardo Linguaglossa

► **To cite this version:**

Minh-Thuyen Thi, Siwar Ben Hadj Said, Adrien Roberty, Fadlallah Chbib, Rida Khatoun, et al..
Enabling programmable deterministic communications in 6G. 6G-PDN Workshop - 6G Programmable
Deterministic Networking with AI, Oct 2023, Washington, United States. hal-04210299

HAL Id: hal-04210299

<https://telecom-paris.hal.science/hal-04210299v1>

Submitted on 18 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Enabling Programmable Deterministic Communications in 6G

Minh-Thuyen Thi
Université Paris-Saclay, CEA, List,
France

Siwar Ben Hadj Said
Université Paris-Saclay, CEA, List,
France

Adrien Roberty
Université Paris-Saclay, CEA, List,
France

Fadlallah Chbib
Telecom Paris, Institut Polytechnique
de Paris, France

Rida Khatoun
Telecom Paris, Institut Polytechnique
de Paris, France

Leonardo Linguaglossa
Telecom Paris, Institut Polytechnique
de Paris, France

ABSTRACT

Emerging applications and technologies such as vehicle-to-everything (V2X), edge-computing, and artificial intelligence have emphasized the demand for low-latency and deterministic communication. Although the 5G network has taken several efforts to fulfill this demand, such as with 5G-Time-Sensitive Networking (TSN) integration and DetNet, these efforts must be significantly expanded in 6G to fully achieve end-to-end deterministic communication. In this paper, we explore the problem of programmable deterministic communication in the new architecture of 6G. To deal with this problem, we rely on TSN, which has been proven to be a promising solution for deterministic communication. We take V2X as a use case, then investigate the two greatest challenges of this use case: low-latency communication and programmable network management for deterministic communication. To deal with these challenges, we introduce two solutions: (i) TSN low-latency scheduling supported by Multi-Agent Deep Reinforcement Learning, and (ii) programmable network management supported by SDN and joint cloud-infrastructure control. For each solution, detailed architecture and functionality design are presented. We show their high feasibility, applicability, and potentialities through comprehensive definitions, detailed explanations and in-depth qualitative analysis.

KEYWORDS

6G, Time-sensitive Networking, V2X

ACM Reference Format:

Minh-Thuyen Thi, Siwar Ben Hadj Said, Adrien Roberty, Fadlallah Chbib, Rida Khatoun, and Leonardo Linguaglossa. 2018. Enabling Programmable Deterministic Communications in 6G. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

While the fifth-generation network (5G) is currently being deployed around the world, visions of the sixth-generation network (6G) are being introduced by several actors in the industry and academia.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

From a user perspective, 6G is expected to connect the human, physical, and digital worlds; while from a technical perspective, various advancements will be made, especially in Artificial Intelligence (AI), eXtreme Ultra-Reliable Low-Latency Communication (xURLLC), and network virtualization.

In 5G and 6G, the applications and technologies such as Vehicle-to-Everything (V2X), Virtual and Augmented Reality (VAR), and AI have significantly increased the demand for low-latency and deterministic communications. The current 5G network has introduced several solutions to address this demand, such as the integration of 5G and Time-Sensitive Networking (TSN), DetNet, URLLC standardization, and network slicing. However, for the following reasons, these solutions need to be much more enhanced in 6G to achieve End-to-End (E2E) deterministic communication.

First, 5G cannot adequately meet the requirements of massively data-intensive real-time applications such as cloud computing-enabled connected vehicles and VAR-based remote surgery [15]. For example, in those applications, the target of E2E latency is in the order of milliseconds [14]. Second, even though 5G network slicing can enable the provision of isolated resources for different services, this slicing does not sufficiently take into account deterministic communication [9]. Finally, the efforts in 5G should also be expanded to address new architecture and technologies in 6G (e.g., RAN-Core converged cell-free cloud-native architecture, AI techniques).

This paper proposes to rely on IEEE 802.1 TSN [3] for deterministic communication in 6G because TSN has been shown to be a promising technology for low latency and reliable communications. TSN, DetNet, and URLLC standardization are among the main approaches for deterministic communication in 5G/6G. While the 3GPP standardization of URLLC focuses on physical layer, DetNet focuses on network layer, and TSN covers link layer. Even though several efforts have been made to integrate TSN into 5G [17], a native integration of TSN into 6G will greatly support deterministic communications [1]. Indeed, 6G is expected to have a RAN-Core mesh connectivity, and have real-time exchanges among its cloud servers (i.e., far edge-, edge- and central-clouds). This mesh topology and real-time exchanges are highly compatible with Ethernet TSN. In addition, TSN allows to mix heterogeneous traffic constraints on the same medium, then to support multiple applications with different requirements.

Programmability for deterministic communication is another important aspect of 6G. First, since 6G is envisioned to have a new service-centric and cloud-based architecture, programmability is needed to leverage the interoperability among these cloud-based

services. Second, programmability technologies especially Software-Defined Networking (SDN) and Network Function Virtualization (NFV) are essential for network slicing in multi-tenant operations. This is because the resources and network functions across each slice should be managed and orchestrated/coordinated in a dynamic manner by multiple tenants/service providers. Finally, SDN has been shown to be highly relevant for managing TSN, as presented in several studies [6, 10, 12, 16].

To investigate deterministic communication in 6G, we take V2X as a use case because this use case is among the most challenging applications of 5G and 6G [2]. This is because the safety-critical scenarios of V2X create many stringent requirements, especially in terms of latency and reliability. Moreover, since computation-intensive applications such as autonomous driving pave the way for further deployments of far edge-, edge- or cloud-computing [4], the communication delay between vehicles and remote cloud servers poses a more significant challenge. We analyze two of V2X's most critical challenges, which are the requirement for low-latency communication and a sophisticated design for programmable network management.

To address the first challenge, we propose to use Multi-Agent Deep Reinforcement Learning (MADRL) for computing IEEE 802.1Qbv schedules. In TSN, IEEE 802.1Qbv is a standard that defines scheduling mechanisms for TSN bridges, allowing to prioritize network traffic and control the latency of critical flows. For simplicity, hereafter the standard IEEE 802.1Qbv can be referred to as Qbv. The Qbv schedules can be generated manually or via heuristic algorithms. However, besides heuristics, Deep Reinforcement Learning (DRL) has showed to be highly relevant for the planning problems like this scheduling problem. DRL is also relevant when we need to search for a time-varying and evolving solution, which is also applicable to scheduling problems. Moreover, due to the problems of scalability and complexity, we avoid using a global and centralized DRL approach. Instead, we use MADRL to compute Qbv schedules in a distributed and sequential manner.

To address the second challenge, we introduce two components for network management, namely cloud manager (i.e., cloud-based NS manager) and infrastructure manager (i.e., physical network devices manager). Then we show that these two components have a strong association between them, and that they have a common goal of adapting the physical network devices corresponding to the changes of V2X application and environment. As a result, we propose a network management solution that closely ties together the cloud manager and infrastructure manager. Moreover, in some scenarios, this solution allows these components to jointly contribute to a common decision.

In the literature, there have been several studies that investigate the deterministic communication in 5G [7, 9, 18, 19] and 6G [13]. Among them, some studies target V2X [18, 19], while some others focus on Cyber-Physical System (CPS) and industrial use cases [7, 13]. About methodology, some studies adopt network slicing [9], while some others adopt TSN scheduling [7, 19]. The authors in [18] introduce a system-level architecture for transmitting V2X control signals via TSN in 5G. The objective is to protect the delay and reliability of deterministic traffic when this traffic co-exists with non-deterministic traffic. Among studies considering 6G, the survey article [13] presents the state of the art for E2E deterministic

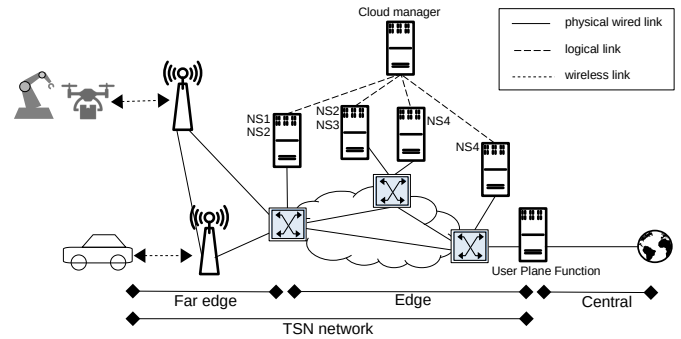


Figure 1: Cloud-based architecture in 6G

communications, then proposes several research directions. The authors show that 6G should address E2E deterministic characteristics across heterogeneous technologies and domains. Differing from the literature, this paper presents a TSN-supported deterministic communication for V2X inside the new architecture of 6G. After analyzing the two main challenges of this V2X use case, we propose two solutions to deal with these challenges.

2 TSN AS ENABLER FOR 6G DETERMINISTIC COMMUNICATION

TSN is a set of standards that are developed by the IEEE 802.1 TSN Working Group. These standards can be categorized into four main groups, namely time synchronization, low latency, high reliability, and network management. TSN network is designed to have high synchronicity, reliable communication, strict latency, and a highly manageable control plan. Especially, with reliable communication and strict latency, TSN allows supporting multiple traffic constraints on the same medium. In this way, the network can adapt to multiple services/applications and fulfill their different requirements.

TSN has been showed to be an important technology for 6G because (i) there are a growing number of applications that require deterministic communication in 6G, and (ii) TSN demonstrates strong alignment with the new Core-RAN mesh topology in 6G.

First, one important aspect of 6G is the service-centric architecture that hosts inside it various cloud-based/cloud-native Network Services (NS) [1, 20]. As in Fig. 1, this architecture leverages cloud computing, with the NSs deployed distributively at far edge-, edge- and central-cloud. These NSs can be controlled and orchestrated by a cloud manager. In a real system, end-devices such as vehicles or robots might need to transmit real-time data to the edge or cloud servers for processing. In the other direction, some command and control messages might need to be transmitted back to the end-devices. For many real-time and parallel processing applications, these exchanges require extremely low latency and a certain level of determinism.

Second, 6G is expected to have a converged RAN-Core mesh topology [8, 20]. On one hand, as this mesh domain has to support real-time exchanges among edge- and cloud- servers, deterministic communication is needed. Specifically, the NSs and physical network devices have to stay synchronized on information such as exchanged data, training data, or newly generated Qbv schedules. On the other hand, Ethernet TSN is highly compatible with

the new RAN-Core mesh topology, which is demonstrated by its mechanisms such as multi-path redundancy transmission (802.1CB), TSN for fronthaul networks (802.1CM), multi-domain time synchronization (802.1AS-Rev). As a result, TSN is a promising research direction to address deterministic communication in 6G [1].

3 PROGRAMMABLE DETERMINISTIC COMMUNICATION IN 6G

To support the high level of programmability, two important aspects are the openness of network infrastructure (i.e., network physical devices) and the interoperability among NSs. This openness facilitates programmability in situations where, for example, external developers need to interact with the network infrastructure. The interoperability among NSs allows providing cloud-native and softwareization architecture, which supports the scenarios such as multi-tenant operation (e.g., multi-network providers, multi-cloud service providers, multi-solution providers).

Programmability is applicable in multiple aspects of 6G. We define that network management in 6G includes two main components, namely cloud manager and infrastructure manager. The cloud manager is responsible for adapting (e.g. deploy, scale up, migrate) the NSs dynamically according to the Service-Level Agreement (SLA) and the change of environment. The infrastructure manager is in charge of controlling and configuring the TSN equipment in accordance with the SLA and environment. In the following, we show that the programmability is applicable to both of these two components.

First, in terms of cloud management, programmability is highly applicable, especially for network slicing. The cloud manager is in charge of controlling and orchestrating/coordinating the resources across the network slices. This manager ensures an appropriate allocation of resources, policies, and an efficient orchestration of NS deployments. Two main technologies for cloud management are NFV and SDN. NFV allows network functions to be virtualized and deployed on-demand, while SDN provides programmable control and dynamic configuration to network infrastructure. Using NFV and SDN, network operators can create virtual network slices with specific resource allocations, performance requirements, and quality of services (QoS). Each slice operates as an independent virtual network with its own set of virtualized network functions. This enables an efficient sharing of common physical infrastructure while providing tailored services for different applications, users, or service providers. In V2X, slice management becomes especially challenging when there is high mobility or there are insufficient local cloud resources. To address such scenarios, 6G should be equipped with a programmable and dynamic cloud manager. This manager will dynamically and flexibly orchestrate network resources to ensure end-to-end QoS.

Second, in terms of infrastructure management, programmability is also highly applicable. Since deterministic communication requires precise control over the timing of data exchange, we need programmability to dynamically control network devices and to respond timely to the network dynamics. For example, SDN can provide programmable control to dynamically manage and configure TSN network infrastructure (e.g., TSN bridges). In this way,

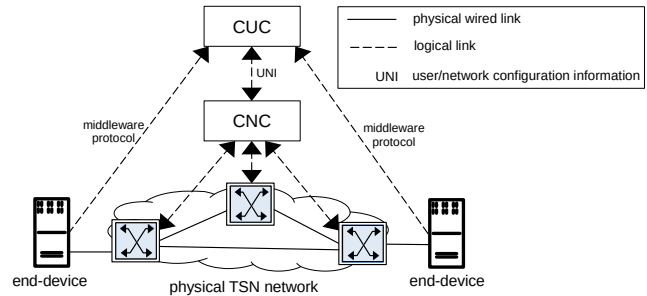


Figure 2: Fully centralized model of IEEE 802.1Qcc

we can program into the TSN infrastructures the traffic prioritization and resource sharing based on demand. Specifically, the main standard of the TSN control plane is IEEE 802.1Qcc [5]. In a recent amendment proposed by 802.1 TSN Working Group, three network management models are specified: fully distributed, centralized network/distributed user, and fully centralized. This standard recommends using the centralized configuration model (Figure 2) for the tasks such as TSN scheduling, network path control, redundant paths management, and time synchronization. This model is composed of Centralized User Configuration (CUC) and Centralized Network Configuration (CNC) entity. While CUC is mainly responsible for collecting the applications' requirements, CNC is in charge of computing and sending configurations (e.g. gates schedules in Qbv) to TSN bridges. In the literature, the CNC entity has been widely implemented by SDN [6, 10, 12]. In this paper, we also propose to use SDN for programmable infrastructure management in 6G.

4 V2X AS A USE CASE OF 6G DETERMINISTIC COMMUNICATION

4.1 V2X Use Case

V2X communication refers to the information exchange between vehicles and other entities in the surrounding environment. Some types of V2X are vehicle-to-cloud or Vehicle-to-Network (V2N), Vehicle-to-Network-to-Vehicle (V2N2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Vehicle (V2V). In the scope of this paper, we focus on the communication of V2N and V2N2V. With V2N, vehicles communicate with clouds and data centers to receive services such as real-time data analysis, remote control, storage, and database access. These services can support a variety of vehicular applications, e.g., real-time road traffic management, and fleet control. With V2N2V, vehicles can communicate with each other, besides direct V2V. This communication allows sharing of information such as speed, position, acceleration, supporting applications like advanced driver assistance systems.

As an example of a V2N scenario, the devices such as sensors and radars collect and transmit real-time data to the cloud, where this data will be processed. By ensuring sufficient data storage and processing resources, the cloud-computing can provide efficient control information for vehicles. This control information may include critical data such as collision warnings and road traffic conditions. Hence, the vehicles typically want to receive these control information in a timely manner.

4.2 Challenges of V2X in 6G

4.2.1 Low-latency Communication. V2X is among the most challenging applications in 5G because this is a safety-critical application in which the vehicles typically travel at high speeds and change direction frequently. V2X communication imposes high demands for latency, reliability, and the number of connected vehicles. Low latency and high reliability enable the rapid exchange of critical information for timely decision-making. As an example, in a platooning scenario with autonomous vehicles driving in a group, the communication latency between them must be less than 10ms [18]. Achieving such low latency with high mobility is a major challenge.

In addition, with the rapid developments of connected vehicles, there are a growing number of computation-intensive applications, e.g., autonomous driving, video-aided navigation. Since these applications usually require large computation resources, edge- and cloud-based processing is preferred, compared to local in-vehicle processing. Then, the communication latency and reliability between the vehicles and remote cloud servers also become a more challenging problem. As a result, 6G system has to align properly with these requirements of low latency and high reliability. The technologies such as TSN, network slicing, and edge computing are among promising approaches to address those requirements.

4.2.2 V2X Programmable Network Management. In terms of cloud management, there is a major challenge in allocating and orchestrating computing resources, NSs, and network slices, while at the same time ensuring different SLA for different V2X services. Firstly, this orchestration requires maintaining data synchronization among computing resources or NSs. Secondly, the cloud manager must also adapt to the dynamics of network traffic and vehicle mobility, in a typically extended geographical coverage. Finally, an extended coverage imposes another challenge on the orchestration and scalability, e.g., migrate/offload computing tasks, deploy, scale up, and activate/deactivate NSs.

In terms of infrastructure management, programmability is also a significant challenge in V2X. Especially, V2X network typically spans multiple domains that encompass heterogeneous technologies, e.g., Radio Access Network (RAN), edge and cloud computing, and tunneling network. Each technology or domain has its own programmable paradigms, interfaces, and control mechanisms. Hence, the programmability across multiple technologies usually requires an interoperability design. This interoperability is possibly provided by standardized interfaces and abstracted underlying technologies. Therefore, the proposed infrastructure manager investigates a fine-grained and intelligent resource allocation mechanism to provide consistent QoS across different domains. To use SDN for infrastructure management, its implementation has to support interfacing with different network devices such as TSN bridges, edge and cloud compute nodes, and 6G radio units. For example, when integrating 5G and TSN, the 5G system is viewed as a virtual TSN bridge; however, this bridge cannot be controlled directly by the CNC as described in 802.1Qcc. Instead, CNC can only do that by interfacing with a new component called TSN Application Function (AF). This TSN AF serves as an interface between CNC and 5G control plane. Note that regarding programmability, there are some challenges that not only relate to research activities but also the hardware development, such as the support of TSN in 6G radio units.

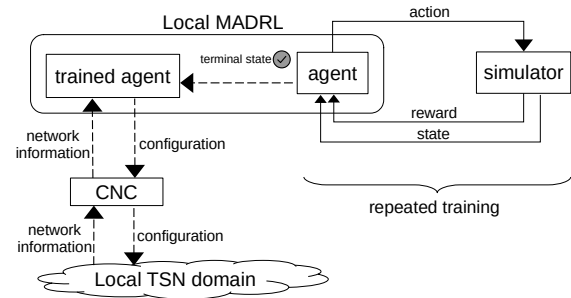


Figure 3: The training of DRL agent and interaction with CNC

5 SOLUTIONS FOR V2X IN 6G

5.1 Low-latency Communication Supported by MADRL

To address the first challenge which is described in section 4.2.1, we propose to use the DRL technique for computing Qbv schedules. The reason is that this technique has been widely adopted to deal with the planning problems like this scheduling problem. Moreover, DRL targets to achieve time-varying and long-term solutions through interacting with the environment, which is also applicable to scheduling problem. The idea behind RL is that an agent will take actions over an environment, then receive in return a reward and a new state of the environment. This process is repeated until the agent found a terminal state that solves the problem (Fig. 3). The reward represents the impact of the action, whereas the state serves as an observation of the environment. In the V2X use case, the state consists of data about network topology, network flows, and their E2E delay. The actions are the Qbv schedules, and the reward is derived from the E2E delay of critical network flows (also called TSN flows). More detailed descriptions can be found in [11]. In Fig. 3, after the agent found a terminal state that solves the problem, the trained agent will be used as a Qbv scheduler.

An enhanced version of DRL is MADRL, in which multiple agents coexist in the same environment. These agents can act towards their own reward or a common reward. The environment might be partially observed by each agent. In this paper, we use MADRL instead of DRL due to scalability and complexity reasons. First, as discussed in section 4.2.1, V2X usually deals with high mobility and extended geographical coverage; hence, deploying a single DRL agent for the entire system will pose a serious problem of scalability. Second, as showed in the experiments of our previous study [11], DRL training is a highly time-consuming task. For example, when running DRL on a machine of a 4-core processor 1.6GHz and 16GB memory, with a linear topology network, the agent takes few tens of seconds to train and generate Qbv schedules. Therefore, the MADRL approach is more adaptable in a large-scale system.

Fig. 4 depicts the implementation of the proposed solution, which includes an E2E TSN data plan and a service-based control plan. The NSs such as *cloud manager*, *local MADRL*, and *global MADRL* are deployed distributively on far edge-, edge- and central-cloud. The *local MADRL* is in charge of computing Qbv schedules for the local network domain. Note that in the proposed MADRL, the local agents act in a cooperative manner. Before applying those Qbv

schedules into the TSN bridges, IEEE 802.1Qbv also requires that the clocks on the bridges are first synchronized. This synchronization is provided by 802.1AS for both wired and wireless domains of the cellular system, as described in our previous research [17].

We designed it such that, to find the global scheduling solution (i.e., *joint action* as defined in MADRL), the global agent receives local solutions from local agents in a distributed and hierarchical manner. From a game theory perspective, this is a sequential and cooperative process. From a federated learning perspective, the training of local agents is done with the local data taken from the local domain. As an example in V2X, when certain geographical domains experience high vehicle density, local agents within those domains should be activated. After generating local Qbv schedules/configurations, local agents apply these schedules to TSN bridges inside the local domain, and at the same time transmit these schedules to the global agent for computing global schedules. Then these global schedules are sent back to local agents for applying to TSN bridges. This process is repeated if E2E QoS is not reached.

In Fig. 4, we also illustrate the exchanges when applying Qbv schedules to the TSN bridges. Specifically, after being generated by *local MADRL*, the Qbv schedules are transmitted from the *local MADRL* to CNC, and then from CNC to the two TSN bridges of the local domain. Even though not illustrated, there is also another exchange before the exchange of Qbv schedules. Indeed, before computing Qbv schedules, the *local MADRL* also receives network information from CNC. This network information can include data such as the topology of the local domain, Qbv-capable network interfaces, and a list of active network flows. Note that the local solutions are not one-shot globally optimal but they are highly valuable because they are in-time, responsive, and they support to build a scalable solution.

5.2 Joint Programmable Cloud-Infrastructure Management

To address the second challenge described in section 4.2.2, we propose a solution that jointly manages the cloud and infrastructure in 6G. The two important components of this solution are the cloud manager and infrastructure manager (Fig. 5). Cloud manager is comprised of *slice orchestrator* and *NS orchestrator*, whereas the infrastructure manager consists of CNC and CUC. The CNC exchanges information with physical TSN network based on SDN paradigm. More detailed descriptions of this implementation can be found in [12, 16]. As showed in Fig. 1, the cloud manager controls the NSs in a centralized manner, i.e., this manager has a global view of the systems of far edge-, edge- and central-cloud.

Note that in this proposed solution, there is only one CNC, one global MADRL agent, and multiple local MADRL agents corresponding to multiple disjoint local network domains. The clustering of local network domains is decided by the infrastructure manager. An architecture of multiple CNCs and/or multiple overlapping local domains might be also feasible but requires a more sophisticated design and implementation.

In a programmable 6G system, the cloud manager and infrastructure manager have close association between them. Firstly,

as showed in Fig. 5, the two orchestrators inside the cloud manager need the infrastructure manager to interact with the physical network. Secondly, in certain scenarios, the cloud manager and infrastructure manager need to jointly make decisions about infrastructure configuration.

First, the *NS orchestrator* must coordinate and adapt NSs according to the change of V2X application and environment. However, when the NSs need to obtain information about network devices (called *network information*) or need to apply new configurations to network devices, these NSs must request to infrastructure manager. Similarly, the *slice orchestrator* also has to pass through the infrastructure manager to reach network devices. As an example, when a vehicle travels across different geographical domains, it possibly encounters situations where available computing resources in the new domain are insufficient to meet its requirements, or simply this traveling triggers a change in network flows. In such cases, the cloud manager needs to adapt the NSs accordingly. This manager will decide, for example, to deploy, activate/deactivate, scale up, migrate, or reallocate resources for the NSs. In the case of MADRL, the agents might need to dynamically recompute Qbv schedules. Then the infrastructure manager must also apply new changes to the network infrastructure accordingly, such as applying new MADRL-generated Qbv schedules to the TSN bridges. These dynamic adaptation processes allow that V2X services maintain a high degree of continuity, providing an optimal service under varying network conditions.

Second, in some scenarios, the cloud manager and infrastructure manager need to jointly optimize network infrastructure. As an example, before deciding to activate a local MADRL agent, the *NS orchestrator* requests information about TSN-capability of network devices. And after the decision, the infrastructure manager can jointly optimize these TSN devices, such as deciding which devices should activate their time synchronization process, whether 802.1AS grandmaster should be changed, or how the mapping of flow priority should be updated (e.g., in case there is a *join-request* for a new flow that has new traffic class with new priority). In this way, the proposed network management solution allows to reach a more optimal network configuration under varying environment.

6 CONCLUSION

This paper explores the challenges of programmable deterministic communication in 6G. With V2X as a use case, we identify two of its most significant challenges, which are low-latency communication and programmable network management. Then we propose two solutions to address these challenges. One solution provides low-latency TSN scheduling with a support from MADRL; the other solution provides joint cloud-infrastructure management with a support from SDN. By providing detailed architecture design with comprehensive definitions and in-depth analysis, we show the high feasibility, applicability, and potentialities of these solutions. For future works, a desired direction is to evaluate the MADRL solution in a 6G proof-of-concept testbed, with new cloud-based architecture. This evaluation raises a number of challenges concerning both research and development. In terms of research, since 6G standardization is currently underway, a 6G proof-of-concept must be built in parallel with these standardization activities. In terms

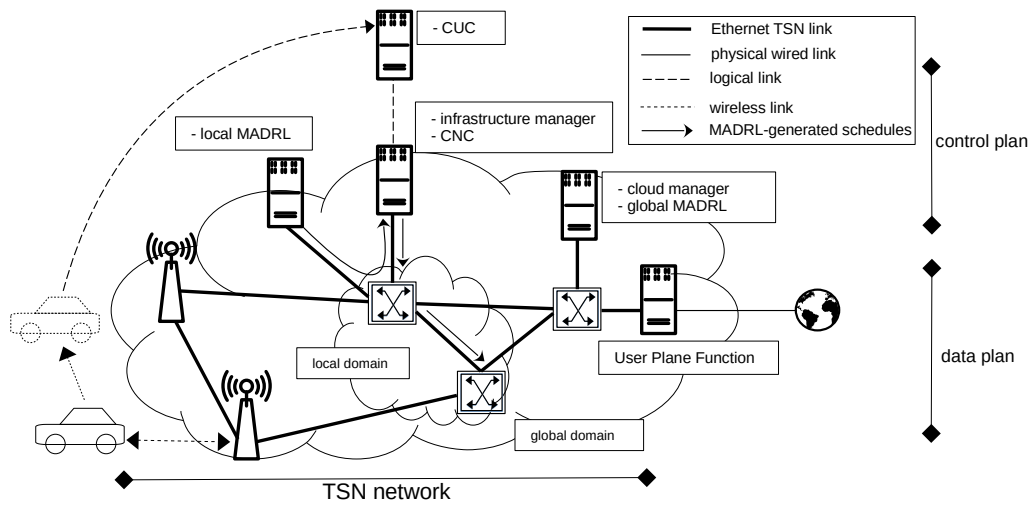


Figure 4: MADRL-supported low-latency communication for V2X

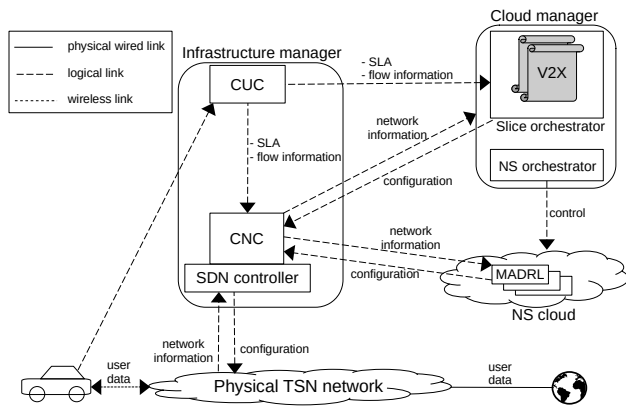


Figure 5: Joint cloud-infrastructure network management

of development, we must investigate non-stand-alone and stand-alone design for 6G architecture. A non-stand-alone 6G system is deployed on top of existing 5G, while a stand-alone system can operate independently using a completely new architecture.

REFERENCES

[1] Next G Alliance. 2022. *6G Technologies*. Technical Report.
 [2] Mate Boban, Konstantinos Manolakis, Mohamed Ibrahim, Samer Bazzi, and Wen Xu. 2016. Design aspects for 5G V2X physical layer. In *2016 IEEE conference on standards for communications and networking (CSCN)*. IEEE, 1–7.
 [3] Janos Farkas, Lucia Lo Bello, and Craig Gunther. 2018. Time-sensitive networking standards. *IEEE Communications Standards Magazine* 2, 2 (2018), 20–21.
 [4] Lei Feng, Wenjing Li, Yingxin Lin, Liang Zhu, Shaoyong Guo, and Zerui Zhen. 2020. Joint computation offloading and URLLC resource allocation for collaborative MEC assisted cellular-V2X networks. *IEEE Access* 8 (2020), 24914–24926.
 [5] IEEE. 2018. *Bridges and Bridged Networks Amendment: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements*. IEEE P802.1Qcc/D2.1.
 [6] Thomas Kobzan, Immanuel Blöcher, Maximilian Hendel, Simon Althoff, Alissa Gerhard, Sebastian Schriegel, and Jürgen Jaspermeite. 2020. Configuration Solution for TSN-based Industrial Networks utilizing SDN and OPC UA. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vol. 1. IEEE, 1629–1636.
 [7] Dennis Krummacker, Benedikt Veith, Christoph Fischer, and Hans Dieter Schotten. 2022. Analysis of 5G channel access for collaboration with TSN concluding at a 5G scheduling mechanism. *Network* 2, 3 (2022), 440–455.

[8] Nokia Bell Labs. 2022. *Technology Innovations for 6G System Architecture*. Technical Report.
 [9] Sebastian Martin, Paolo Medagliani, and Jeremie Leguay. 2021. Network Slicing for Deterministic Latency. In *2021 17th International Conference on Network and Service Management (CNSM)*. 572–577.
 [10] Naresh Ganesh Nayak, Frank Dürr, and Kurt Rothermel. 2016. Time-sensitive software-defined network (TSSDN) for real-time applications. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems*. 193–202.
 [11] Adrien Roberty, Siwar Ben Hadj Said, Frederic Ridouard, Henri Bauer, and Annie Geniet. 2023. Configuring the IEEE 802.1Qbv Time-aware shaper with deep reinforcement learning. (July 2023). <https://cea.hal.science/cea-04156793> CoNEXT 2023 - 19th Conference on emerging Networking EXperiments and Technologies Mission, December 5-8, 2023, Paris, France.
 [12] Siwar Ben Hadj Said, Quang Huy Truong, and Michael Boc. 2019. SDN-Based Configuration Solution for IEEE 802.1 Time Sensitive Networking (TSN). *SIGBED Rev.* 16, 1 (feb 2019), 27–32.
 [13] Gourav Prateek Sharma, Dhruvin Patel, Joachim Sachs, Marilet De Andrade, Janos Farkas, Janos Harmatos, Balazs Varga, Hans-Peter Bernhard, Raheeb Muzaffar, Mahin K. Atiq, Frank Duerr, Dietmar Bruckner, Edgardo Montesdeoca, Drissa Houatra, Hongwei Zhang, and James Gross. 2023. Towards Deterministic Communications in 6G Networks: State of the Art, Open Challenges and the Way Forward. arXiv:2304.01299 [cs.NI]
 [14] Maraj Uddin Ahmed Siddiqui, Hanaa Abumarshoud, Lina Bariah, Sami Muhaidat, Muhammad Ali Imran, and Lina Mohjazi. 2023. URLLC in Beyond 5G and 6G Networks: An Interference Management Perspective. *IEEE Access* (2023).
 [15] Faisal Tariq, Muhammad RA Khandaker, Kai-Kit Wong, Muhammad A Imran, Mehdi Bennis, and Merouane Debbah. 2020. A speculative study on 6G. *IEEE Wireless Communications* 27, 4 (2020), 118–125.
 [16] Minh-Thuyen Thi, Siwar Ben Hadj Said, and Michael Boc. 2020. SDN-Based Management Solution for Time Synchronization in TSN Networks. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vol. 1. 361–368.
 [17] Minh-Thuyen Thi, Sosthène Guédon, Siwar Ben Hadj Said, Michael Boc, David Miras, Jean-Baptiste Dore, Marc Laugeois, Xavier Popon, and Benoit Miscopain. 2022. IEEE 802.1 TSN Time Synchronization over Wi-Fi and 5G Mobile Networks. In *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. 1–7.
 [18] Dan Wang and Tao Sun. 2020. Leveraging 5G TSN in V2X communication for cloud vehicle. In *2020 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 106–110.
 [19] Chenyu Zhang, Xiangming Wen, Cong Li, Wei Zheng, Longdan Yu, and Zhaoming Lu. 2023. Dynamic Rapid Scheduling Algorithm for Vehicle Time Sensitive Communication based on CILP and AGA. *IEEE Transactions on Vehicular Technology* (2023), 1–14.
 [20] Volker Ziegler, Harish Viswanathan, Hannu Flinck, Marco Hoffmann, Vilho Räisänen, and Kimmo Hätönen. 2020. 6G architecture to connect the worlds. *IEEE Access* 8 (2020), 173508–173520.