Digital Twin-Based Cloud-Native Vehicular Networks Architecture for Intelligent Driving

Xiaobin Tan, Qiushi Meng, Mingyang Wang, Quan Zheng, Jun Wu, and Jian Yang

Abstract

As a key technology in intelligent driving, Cooperative Vehicle Infrastructure System is an advanced communication framework that enables cooperative and information exchange between vehicles and infrastructure. However, this system encounters challenges in meeting the low latency, ultra reliability, and high efficiency requirements of task execution of intelligent driving applications. Meanwhile, the deployment and combination of some key technologies supporting this system are not well-established. To address these issues, this article proposes a Digital Twin-based Cloud-native Vehicular Networks (DT-CVN) architecture to enhance the efficiency of virtual-reality integration in real-world vehicle traffic scenarios. In DT-CVN, the digital twins, which can bridge the physical space and cyberspace gaps in real-time, are implemented and deployed in a distributed manner by leveraging the distributed features of microservices based on cloud-native technology. DT-CVN employs the cybertwin as a smart communication agent in cloud-native vehicular networks, enabling efficient communication between cyberspace and physical spaces. Moreover, a case study is presented to demonstrate the effectiveness of DT-CVN. Simulation result shows the potential to address the challenges of integrating resources in vehicular networks with our proposed DT-CVN.

Introduction

Intelligent driving have developed rapidly with the advances in sensing, artificial intelligence, and automatic control technologies. The current research on intelligent driving predominantly focuses on individual vehicle intelligence, which is hindered by limited onboard perception and computing capability to acquire comprehensive environmental information. To address these limitations, Cooperative Vehicle Infrastructure System (CVIS) has emerged as a promising solution to facilitate perception and computing sharing required for intelligent driving [1], which holds immense potential to revolutionize transportation systems. In CVIS, vehicular networks are constructed to achieve efficient interaction between vehicle to vehicle, vehicle to road infrastructures, etc.

To fulfill the requirement of vehicle-road collaboration for intelligent driving, vehicular networks must tackle two key challenges. First, vehicular network needs to ensure low-latency and highly reliable data interactions, especially for some latency-sensitive intelligent driving applications (e.g., lane changing and obstacle avoidance). Second, vehicular networks must accommodate the characteristics of massive connected smart vehicles, highly dynamic topology, and limited wireless spectrum [2]. This article focuses on the design of a vehicular network architecture that can effectively support intelligent driving applications within large-scale scenarios.

Mobile Edge Computing (MEC) technology offers an ultralow latency environment with high bandwidth and real-time access to network resources [3]. By introducing MEC technology into vehicular network, Vehicular Edge Computing Network (VECN) is proposed [4]. VECN leverages edge servers to bring computing and caching resources close to the driving scene, offering more sufficient resources compared to the vehicle and lower transmission latency compared to the cloud. By offloading intelligent driving tasks to edge servers, VECN significantly accelerate tasks execution to achieve low latency and high reliability. Moreover, in VECN, sensing data gathered from multiple sensing devices is fused at the edge computing devices. Then cyberspace is established based on the comprehensive perception results, which supports collaborative decision-making for intelligent driving.

However, the massive data interaction, inherent device heterogeneity, and dynamic connectivity topology give rise to a gap between cyberspace and physical space. The importance of Digital Twin (DT), which is characterized by the cyber-physical integration, is increasingly emphasized by both academia and industry [5]. With its capability to offer precise representations and real-time information retrieval of physical entities, DT technology enables the creation of highly immersive and interactive

Xiaobin Tan, Quan Zheng (corresponding author), and Jian Yang are with the Department of Automation, University of Science and Technology of China (USTC), Hefei 230026, China, and also with the Hefei Comprehensive National Science Center, Institute of Artificial Intelligence, Hefei 230039, China; Qiushi Meng and Mingyang Wang are with the Department of Automation, University of Science and Technology of China (USTC), Hefei 230026, China; Jun Wu is with the School of Computer Science, Fudan University, Shanghai 200433, China, and also with the Key Laboratory of Ministry of Education in Embedded System and Service Computing, Tongji University, Shanghai 201804, China.

Digital Object Identifier: 10.1109/MNET.2023.3337271
Date of Current Version: 18 April 2024
Date of Publication: 27 November 2023

> In DT-VECN, DTs help to realize the potential edge service matching among massive vehicle pairs and reduce the complexity of service management.

digital models [6]. By introducing DT technology into VECN, digital twin empowered vehicular edge computing networks (DT-VECN) is proposed [2]. In DT-VECN, DTs help to realize the potential edge service matching among massive vehicle pairs and reduce the complexity of service management. Vehicle can get comprehensive and accurate system state information and history record through its DT, even realizing collaborative perception and decision-making in real-time, so as to improve the efficiency of large-scale collaboration and the resource utilization.

Nevertheless, DT-VECN and most of existing schemes just employ simple monolithic DT implementation methods (i.e., implementing a DT using a virtual machine), which suffers from low resource utilization and lack of scalability [2]. This kind of DT implementation methods fails to adequately support dynamic large-scale driving scenarios, while vehicles are often in high-speed mobility (frequently switching between base stations or edge servers) with stringent intelligent driving latency requirements [7], [8]. Zhan et al. [9] introduced cloud-native based distributed implementation of DT, supporting innovative virtual-reality fusion-based applications.Cloud-native refers to a set of practices and design patterns for developing and deploying applications that take advantage of the distributed computing capabilities provided by the cloud model [10]. In cloud-native network, traditional monolithic applications can be decomposed into multiple distributed microservices (i.e., container) and managed on container orchestration platform such as Kubernetes, which can support vehicular high-speed movement. Additionally, microservices in cloud-native network support auto-scaling and load balancing. This implies that resources can automatically adjust to handle massive devices access.

Based on the above analysis, this article proposes the Digital Twin-based Cloud-native Vehicular Networks architecture (DT-CVN). We introduce cloud-native into vehicular network, realizing the distributed implementation of DT by leveraging the modularity features of cloud-native network. The modules of a digital twin can be deployed in different network locations with the form of microservice, which decreases latency and improves scalability to adapt to massive mobile vehicles access and dynamic driving scenarios. Besides, we design the mechanisms of module reuse and requests aggregation to improve task execution efficiency and system scalability, and employ cybertwin [11] to achieve large-scale heterogeneous devices interconnection between cyberspace and physical spaces. By bridging the gap between cyberspace and physical space, DT-CVN can not only meet stringent latency and reliability requirements imposed by intelligent driving, but also enhance the efficiency of interaction and coordination among vehicles and other components in large-scale scenario.

The rest of this article is organized as follows. We provide a detailed explanation of the components and scenario of the proposed architecture. The mechanisms and workflow of DT-CVN are also introduced. A case study tailored to the scenario is presented to demonstrate the effectiveness of DT-CVN. Finally, we conclude the article, summarizing our findings and contributions.

Architecture of DT-CVN

Overview

In vehicular networks enabled intelligent driving scenario, there exist the physical space and the cyberspace. The physical space of vehicular networks encompasses all nodes with information resources, while the cyberspace encompasses communication agents and real-time model of each physical entity achieved through the integration of information resources. The real-time requirements of data interactions, inherent device heterogeneity, and dynamic topology of vehicular networks necessitate bridging the gap between the cyberspace and the physical space. To address this, DT technology is proposed to enhance the efficiency of interaction between the physical space and the cyberspace. By leveraging cloud-native technology, we implement DTs in a distributed manner within cloud-native based vehicular networks to support intelligent driving applications in large-scale scenarios, leading to the proposal of the DT-CVN architecture.

As Fig. 1 shows, the architecture of DT-CVN consists of four layers: physical layer, resource layer, digital twin layer, and task layer. The physical layer represents all physical entities with resources where vehicle submit task requests via cybertwin. Tasks will be decomposed and scheduled in the task layer. Then, subtasks are allocated to DT in the digital twin layer. The digital twin layer is constructed based on the resource layer, which is composed of functions and data. Each layer will be described in detail as follows.

Physical Layer

The physical layer represents all physical entities with computing, caching, or communication resource in cloud-native vehicular networks, including vehicles, roadside units and cloud. This layer organizes the resources of all network nodes and provides virtualized computation and communication resources. Additionally, vehicles in the physical layer also receive the task processing results from the connected vehicles, edge servers or the cloud. Edge servers are the network nodes with relatively sufficient resources, which receive vehicular tasks from mobile vehicles through radio access points (APs), internally link resources and tasks via cybertwin. To simplify this expression, the roadside unit (RSU) can be thought of as combining an edge server with a wireless access point and cybertwins.

Vehicle: In vehicular network, each vehicle serves as both a basic element and controlled object in traffic conditions. Vehicle can process certain types of tasks independently with its computing power, storage capacity, and sensor units. In more complex scenarios, vehicles may

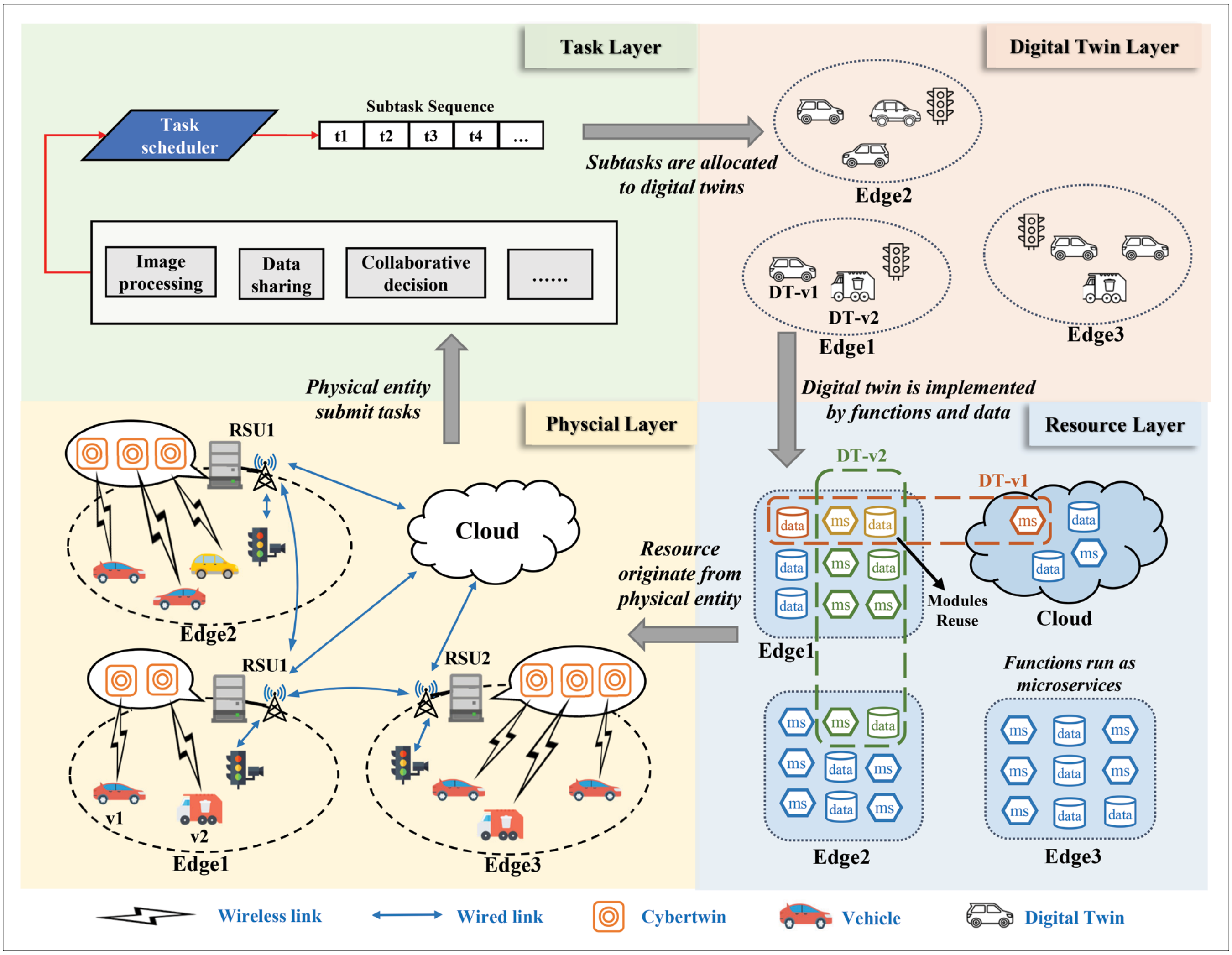

FIGURE 1. The architecture and workflow of DT-CVN.

offload their tasks to roadside devices or reuse the results obtained by other vehicles due to their limited resources. As such, a vehicle is not only a task processing unit but also a resource provider node. Meanwhile, vehicles are often in high speed moving and frequently access to different roadside units.

RSU: The roadside unit resides between the cloud and the vehicles. RSU is equipped with ample sensors along the roadside, relatively sufficient computing resources and storage resources (edge server), and APs. Roadside devices such as traffic lights or cameras will preheat a certain number of function modules for long-term application scenarios for vehicle access. Although its information resource is not as extensive as that of the cloud, its proximity to the ends allows for lower transmission latency in responding to requests versus the cloud. As illustrated in Fig. 1, RSU and all devices within its range are collectively referred to as the Edge. Moreover, an RSU has the capability to offload its tasks to other connected RSUs or the cloud.

Cloud: In DT-CVN, cloud is fully connected with each RSU to build the entire networks through high-speed optical links, allowing for the creation of a comprehensive network. The cloud in DT-CVN provide sufficient computing, caching and communication resources for the vehicles as a network infrastructure service. The processing time of intelligent driving applications could be considerably reduced in cloud compared to vehicle or RSU. However, it's necessary to note that the considerable distance and potentially unstable connections in end-to-cloud may incur much higher transmission delays and energy consumption.

> The cloud in DT-CVN provide sufficient computing, caching and communication resources for the vehicles as a network infrastructure service.

Resource Layer

The resource layer is an abstraction of the information resources, including computing, cache, bandwidth resources, etc, which exist distributedly in the physical space. The resource layer mainly focuses on two requirements. First, the inherent resource heterogeneity and vehicular dynamism require a cross-space resource management framework. Second, the cross-space intelligent driving service require a unified and complete network resource expression to support complex information interaction [12].

In vehicular network, the resources of the cloud, edge and vehicles are heterogeneous both horizontally and vertically. And the high-speed mobility of large-scale vehicles brings dynamic resource status. The resource layer abstracts the resources of physical entities into modules. Resource layer primarily focuses on allocating resources for modules, which exist in the form of microservices and constitute digital twins. The abstraction mapping of information resources is aimed at better resource management, converting diverse service demands for intelligent driving into the cross-space resource requirements. This modular approach is more suitable for large-scale access scenarios when vehicles switch between different RSUs, improving the efficiency of resource management.

Digital Twin Layer

Digital twins offer precise representations and real-time information retrieval of physical entities, enabling the interaction between physical entities and digital models. Most of the current research agree that digital twin is consist of data and models [5]. Based on [9], we propose a distributed implementation method of digital twin by using cloud-native technologies. As shown in Fig. 1, a digital twin consists of function modules and data modules. These modules exist on multiple RSUs and cloud simultaneously in the form of microservices, which relies on a set of cloud-native technologies for execution (e.g., Docker, Containerd), communication (e.g., Istio), management (e.g. Kubernetes), etc.

With the connection in vehicles, RSUs and cloud, an interconnected digital twin layer is constructed based on the resource layer. Digital twin layer can fully sense the state information and real-time events of all physical entities, its distributed characteristics reduce the consumption and latency when vehicles are moving and access to different RSUs. As a bridge between virtual-reality, digital twin can map the intelligent driving tasks generated in the physical space with the information resource represented by function module and data module, supporting comprehensive perception and collaborative decision-making. The function module and data module will be described in detail as follows.

Function Module: Function is an abstract description for the basic ability of physical entities, which can be used to complete a certain task independently. For example, when constructing a digital twin for an engine, it can be represented by a function whose input is throttle opening and output is thrust. Microservice architecture is used to implement the functions of DT in DT-CVN. Given this perspective, computational tasks could be processed by DT. That is to say, a digital twin may contain hundreds of functions modules which are distributed in different network nodes. and they logically form the digital twin.

Data Module: Data module is responsible for managing the data of the digital twin. Data is the state information of physical entities collected by sensors and essential data and models are stored in advance. Only critical and updated data will be transmitted in real-time to maintain the authenticity and accuracy of the digital twin, including task requests. Data is divided into private data and public data. Private data is strictly protected while public data can be shared by all digital twins after being authenticated by cybertwin. In DT-CVN, the data module can be deployed in the form of database microservice.

Task Layer

Through the organization of the digital twin layer, the resources of both physical space and cyberspace are efficiently utilized for many applications, such as data sharing, collaborative decision, task processing, and so on. In DT-CVN, task based on DTs can be decomposed into a series of invocations of DT modules. Data sharing and collaborative decision mainly rely on the interconnection between the digital twins of vehicles, through the invocation of data modules among vehicles or the reuse of inherent modules in roadside devices, the tasks can be effectively accomplished, which can greatly improve the perception capabilities of vehicles in DT-CVN, and on this basis, achieve more efficient collaborative decision-making. Meanwhile, task scheduling and processing are optimized in DT-CVN. By designing an effective scheduling algorithm, the efficiency of task processing can be greatly improved.

Task layer contains a task scheduler, which is designed to receive the task request submitted by vehicle, such as task processing, collaborative decison and so on. After sending the request to the edge cloud, the task scheduler will decompose a complex task into multiple subtasks, each subtask can be executed on a separate microservice. Task scheduler will assign these subtasks to microservices according to the status informatio, which are contained in the digital twin corresponding to the vehicle.

Workflow and Mechanisms of DT-CVN

As Fig. 2 depicts, the scenario diagram of DT-CVN is comprised of physical space, cyberspace, and the mapping relationships between them. The physical space consists of vehicles and roadside devices (i.e., access points, traffic lights, edge server, sensors, etc.) while cyberspace includes digital twins and information resources. The mapping relationship is the cybertwin mentioned above, which serves in multiple capacities, that is, communication assistant, network data logger, and digital asset owner. This approach provides a new cybertwin based communication model that replaces the traditional end-to-end communication model. By leveraging the power of cybertwin technology, a digital twin can faithful map the entities in physical space with the information resources contain storage, computing, and dataset services that may used for executing intelligent driving tasks. Leveraging the virtual-real fusion capability of digital twin technology, all entities involved in a certain intelligent driving task may share information efficiently in cyberspace, and further map the actions in cyberspace into physical entities.

The design of digital twin in vehicular networks has many benefits. On the one hand, the system could obtain more accurate information about vehicular tasks, including resource requirements and deadlines, so that the task processing can be effectively enhanced by the algorithm. On

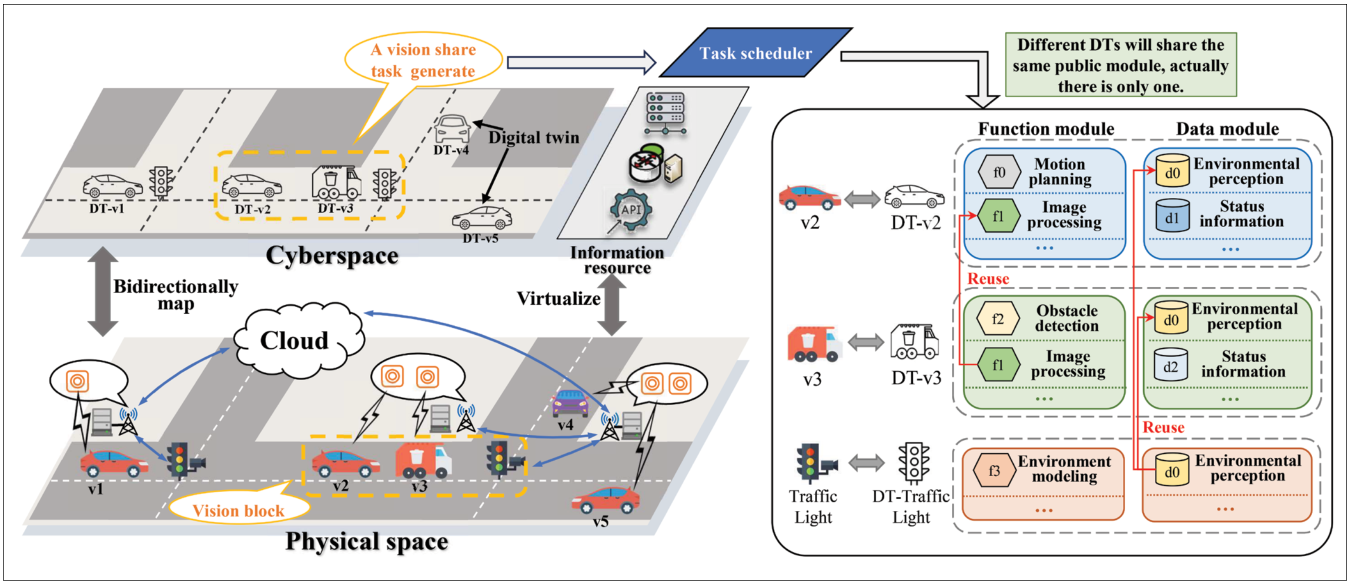

FIGURE 2. The scenario and mechanisms of DT-CVN.

the other hand, the distributed implementation refines the granularity of task processing. Modules of digital twins are composed of microservices, which means these functions and data can be flexibly deployed and managed to increase the utilization of resources. To better leverage the characteristics of DT and distributed nature, we have designed the following mechanisms to enhance system efficiency and showcase the workflow.

Mechanism

When using digital twins to execute tasks in DT-CVN, each task needs to be assigned enough resource. With a large number of tasks and constrained resources, it is impossible to independently allocate sufficient resources for each arriving task. In DT-CVN, digital twins are decomposed into multiple microservices, we can adopt the characteristics of microservices to help improve the efficiency of digital twins. Several mechanisms are proposed as follows.

Module Reuse Mechanism of DT Modules: To utilize the existing resources more efficiently, the reuse mechanism of digital twin modules is proposed. As mentioned before, digital twin modules are deployed in a distributed microservices manner. After being deployed, a microservice can be scheduled and executed multiple times for different users or requests [13]. If one microservice instance acts as same component of two digital twins, then the microservice is called reused. The proposal of the reuse mechanism is important and intuitive. From startup to shutdown, a microservice will occupy the resources allocated by the system during this period. When each digital twin calls the instance to execute a task, it will not occupy its entire period, which means there will be idle time, and the reuse mechanism is to maximize the use efficiency of each instance.

Requests Aggregation Mechanism of Task Requests: When different vehicles submit tasks with the same type and parameters to DTs, it identifies specific information about those tasks. For the tasks that meet the aggregation requirements,

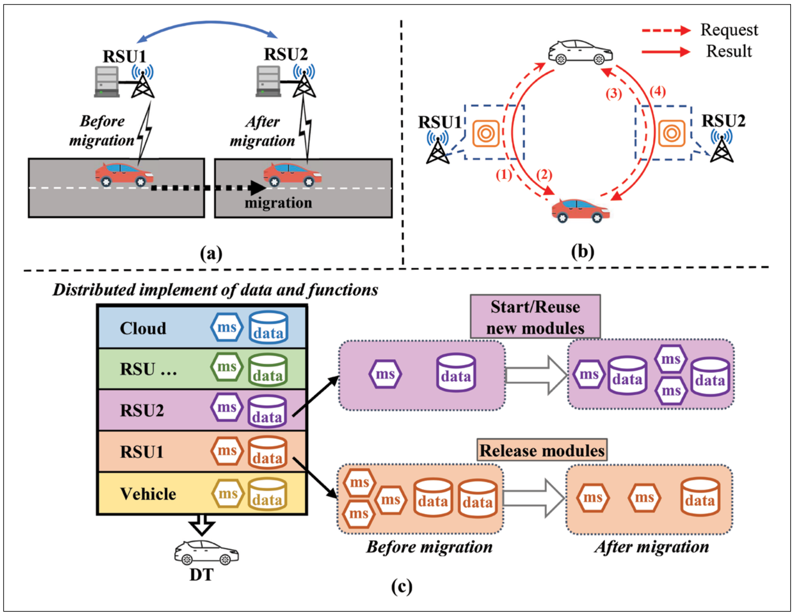

FIGURE 3. The migration of digital twins. a) Physical migration. b) Vehicle links to DT via CT. c) DT's migration

the task only needs to be executed once and the execution result of the task is sent back to the different vehicles. This is the aggregation mechanism of task requests. For example, the environment modeling requests of all vehicles in the same intersection at the same time, can be executed once and sent to all vehicles. The aggregation mechanism is triggered by stringent conditions, but it can further reduce the resource overhead and improve resource utilization.

Migration of DT: When vehicles are on the road, it's not efficient to transfer their corresponding digital twins, which contain a large amount of cached data and functions [2]. Our proposed digital twin is distributed in nature, allowing vehicle to manage its modules in distributed locations through the cybertwin. This enables digital twin to start and release only partial modules during

DT-VECN typically offloads an entire task to the edge or cloud, while DT-CVN focuses on the decomposition of tasks into subtasks.

vehicle migration. As illustrated in Fig. 3: (a) Vehicle connects to different RSUs before and after physical movement. (b) Vehicle will submit/accept its request/result by completing certification by linking the internal cybertwins to its mapping DT. (c) DT adjusts the distributed deployment of the vehicle's digital twin considering factors such as latency and energy consumption based on its specific location, starts, resues, or releases some modules. The distributed nature of DT will greatly reduce the migration consumption when vehicles are on high-speed mobility.

Workflow

As shown in Fig. 1, the workflow of task execution in DT-VCN is described as follows. (1) In intelligent driving, vehicles generate different types of tasks and submit these task requests to RSUs via cybertwins. (2) The tasks submitted by vehicles in the application layer issue task requests and form a subtask queue. (3) After being decomposed to Task scheduler, subtasks are scheduled and executed by the DTs in a microservice way. (4) The task scheduler acquires data about digital twins and effectively assigns each DT some resource.

For instance, in Fig. 2, we have designed a set of intelligent driving tasks inspired by [8]. These tasks come with stringent latency requirements and occur frequently. In Fig. 2, consider v2, a vehicle approaching a complex intersection with limited visibility due to the presence of vehicle v3. To make informed decisions about its next actions, v2 relies on a combination of information from preceding vehicles, traffic light, and roadside sensors. In response, v2 generates a vision sharing task request and transmits it to RSU after undergoing CT authentication. After analizing request and decomposing task, task scheduler enables v2 and v3 to collaboratively share data stored in the Environmental Perception data module within the roadside devices. Concurrently, owing to similarities in state and environment of v2 and v3, image processing requests from v2 and v3 can be handled by the Image Processing function module. Thus, the Image Processing function module and Environmental Perception data module serve as shared components within both DT-v2 and DT-v3 simultaneously.

Considering the mobility, vehicle will inevitably pass through the scope of multiple RSUs during migration. Therefore, we need to carefully consider the management of access and authentication for digital twin. We mainly manage the migration between digital twins through cybertwins, the process is shown in Fig. 3.

Simulation and Performance Evalution

This section provides a case study on DT-CVN architecture and compare it with the DT-VECN [2], aimed at verifying the system validity. In the following, we first show the reasonableness of simulation and present the specific parameters. Then we adopt a task scheduling algorithm [13] and design the performance metrics to evaluate DT-CVN and DT-VECN.

Simulation parameter name	Value
Num of Vehicles/RSUs/Cloud	20/4/1
Num of Tasks	500/1000/1500/2000/2500/3000
Num of subtasks contained in a task	1-5
Num of subtask types	107
Process time of subtasks	0.2-1s
Computational cost of Vehicle/RSU/Cloud	1/2/3 J/s^{-1}
Transmission delay of Vehicle/RSU/Cloud	60/30/100 ms
Subtasks capacities of Vehicle/RSU/Cloud	5/15/25

TABLE 1. Simulation parameters for DT-CVN/DT-VECN.

A Case Study for DT-CVN

Based on [1] and [8], we can observe that a complex intelligent driving task, such as lane changing, can be decomposed into a collection of multiple subtasks, such as image segmentation, recognition, prediction, etc. Each subtask can be regarded as a microservice. In DT-CVN, DT is deployed in the form of microservices, so we can transform the intelligent driving problem in the vehicular network into a virtualized microservice placement problem [13]. To simulate the diverse intelligent driving tasks, the generated task contains some subtasks with different types in the dataset.

Both DT-CVN and DT-VECN adopt MEC technology. Thus, we also simulate and establish a heterogeneous vehicle-edge-cloud collaborative network, the processing time varies across different nodes for the same task. Generally, vehicle node has limited information resources and it will take much more time to execute the tasks locally. Conversely, RSU and cloud offer higher computational power and abundant resources. Nevertheless, leveraging this computational capacity comes at the cost of transmission cost overhead. Additionally, we also support task offloading from vehicle to vehicle via RSU with high transmission delay. Compared to DT-VECN, the advantage of DT-CVN architecture lies in its greater flexibility. DT-VECN typically offloads an entire task to the edge or cloud, while DT-CVN focuses on the decomposition of tasks into subtasks. This approach reduces the resource scale required for offloading and enhances the utilization of idle resources. Besides, we design the mechanisms of module reuse and requests aggregation to improve task execution efficiency and system scalability.

For the computation-intensive and delay-sensitive intelligent driving tasks, shorter execution time means more safety and better driving experience. Under this motivation, it is important to design a task scheduling algorithm based on the digital twin running on cloud-native architectures to handle the scheduling and executing problem of massive intelligent driving tasks. In this simulation, we adopt the First Fit Offloading task scheduling [14], which is an effective online scheduling algorithm with low computational complexity, and ensures the fairness of task scheduling. We refer to the intelligent driving data in [7] and set simulation parameters. As shown in Table 1.

Performance Metrics

We conduct experiments and simulations in terms of the intelligent control and management of task scheduling under the digital twin model. The evaluation indicators mainly includes two aspects.

Average Time Consumption: Average time consumption is the main metric of the simulation experiment. Task time consumption refers to the duration from the generation of a task to the return of its result to the vehicle. The average time consumption represents the average value of all task time consumptions. Task time consumption comprises the processing time of the task and the waiting time. The waiting time includes queuing time, task decomposition time, transmission time, and other factors. The smaller the average time consumption, the more effective the algorithm is in reducing the time cost.

Average Energy Consumption: The energy consumption can be obtained by multiplying the execution time of a task by its computational cost, and then adding the communication energy consumption. When calculating the average energy consumption, we consider the energy consumption per task generated during a period of 1 second. The smaller the average energy consumption, the more effective the algorithm is in reducing energy consumption.

We also need to consider the frequency of task generation as a reference, because once tasks are too intensive for a period of time, the efficiency of the system will decrease and the energy consumption will increase. Here, we count how many tasks are generated within 1 second. In addition, to better evaluate the effect of the design mechanism, we tested the processing time reduced by the different mechanisms introduced respectively.

Simulation Result

We conduct the simulation with the parameters listed in Table 1. The time consumption and energy consumption of these architectures are compared under different numbers of tasks. The performance of the proposed DT-CVN architecture and comparative DT-VECN architecture as baseline are shown in Figs. 4 and 5.

Fig. 4 shows the average time consumption and time reduction by mechanisms under different numbers of tasks. The most significant variance between DT-CVN and DT-VECN is reflected in the waiting time. This discrepancy arises from the distributed nature of DT-CVN. Once a task is decomposed, its execution is more flexible. There is a certain variance in the average time consumption between DT-CVN and DT-VECN. This is due to the impact of the designed module reuse and requests aggregation mechanisms. Moreover, when numerous tasks are congested within a short time frame, the likelihood of reuse increases due to the higher volume of similar requests and server load. Consequently, DT-CVN achieves superior results in large-scale scenario.

Fig. 5 provides insights into task arrival frequency and average energy consumption based on a dataset of 3000 tasks. It can be found that the average energy consumption mainly depends

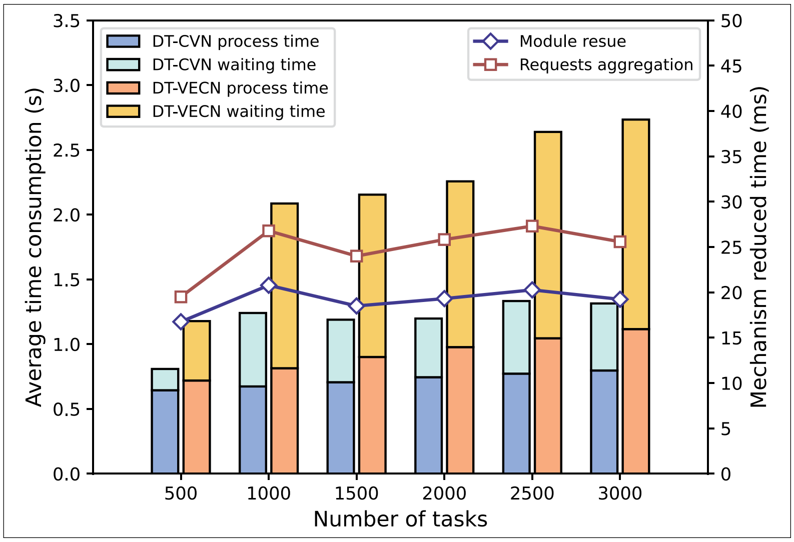

FIGURE 4. Average time consumption and time reduction by mechanisms under different numbers of tasks.

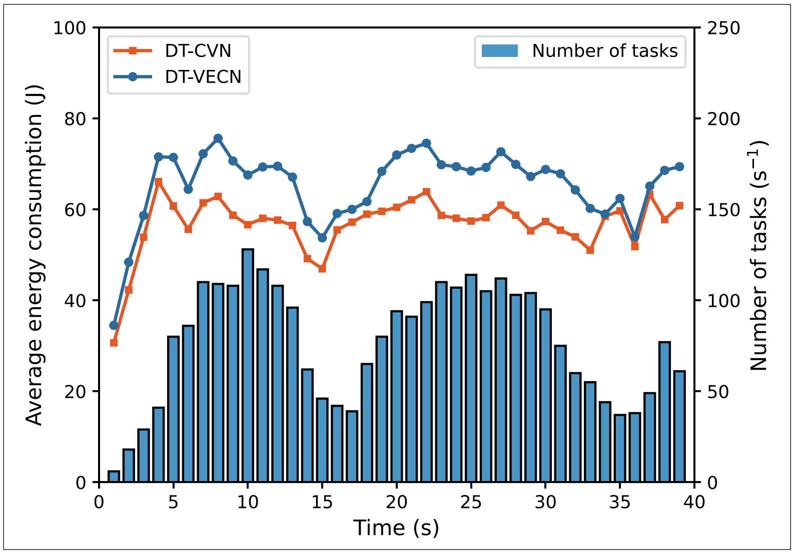

FIGURE 5. Average energy consumption under the distribution of task number.

on the frequency of tasks generation in the past period of time. When task requests are congested, relatively more energy is consumed and DT-CVN exhibits significantly lower energy consumption than DT-VECN during congestion. This phenomenon is partly attributable to reduced processing times and partly due to more granular management, which lowers resource utilization. Above all, due to the distributed characteristics brought by cloud-native technology and mechanisms introduced, the DT-CVN architecture has a significant improvement over the DT-VECN architecture in each indicator, with higher flexibility and scalability to support large-scale vehicles access in low-latency and high reliabible requirements.

Conclusion

In this article, we have introduced a digital twin-based cloud-native vehicular networks architecture. Then, an implementation of digital twin in DT-CVN are presented by taking advantage of microservices in cloud-native networks. We also design some specific mechanisms to improve resource utilization and optimize system performance. A case study is presented to demonstrate effectiveness of DT-CVN, which shows a significant improvement compared to DT-VECN in both time consumption and energy consumption. In the future, we will further study the real-world scenarios, conduct further research on tasks with deadlines requirements and dynamic acyclic graph tasks.

Acknowledgment

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFA0711400; in part by the Key Science and Technology Project of Anhui under Grant 202103a05020007; in part by the China Environment for Network Innovations under Grant 2016-000052-73-01000515; and in part by the National Natural Science Foundation of China under Grant 62101525, Grant 62341113, and Grant 62021001 by the Fundamental Research Funds for the Central Universities.

References

[1] F. You et al., "Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system," *Expert Syst. Appl.*, vol. 42, pp. 5932–5946, Aug. 2015.

[2] K. Zhang, J. Cao, and Y. Zhang, "Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1405–1413, Feb. 2022.

[3] A. Bozorgchenani et al., "Computation offloading in heterogeneous vehicular edge networks: On-line and off-policy bandit solutions," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4233–4248, Dec. 2022.

[4] C. Yang et al., "Learning based channel allocation and task offloading in temporary UAV-assisted vehicular edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9884–9895, Sep. 2022.

[5] F. Tao et al., "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[6] L. Zhao et al., "Intelligent digital twin-based software-defined vehicular networks," *IEEE Netw.*, vol. 34, no. 5, pp. 178–184, Sep./Oct. 2020.

[7] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11618–11628.

[8] C. Xu et al., "AoI-centric task scheduling for autonomous driving systems," in *Proc. IEEE Conf. Comput. Commun.*, May 2022, pp. 1019–1028.

[9] Y. Zhan et al., "Implementation and deployment of digital twin in cloud-native network," in *Proc. Int. Conf. Emerg. Netw. Archit. Technol.*, vol. 1696, 2022, pp. 13–25.

[10] D. Gannon, R. S. Barga, and N. Sundaresan, "Cloud-native applications," *IEEE Cloud Comput.*, vol. 4, no. 5, pp. 16–21, Sep./Oct. 2017.

[11] Q. Yu et al., "Cybertwin: An origin of next generation network architecture," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 111–117, Dec. 2019.

[12] B. Tan et al., "Toward a future network architecture for intelligence services: A cyber digital twin-based approach," *IEEE Netw.*, vol. 36, pp. 98-104, 2022.

[13] V. Farhadi et al., "Service placement and request scheduling for data-intensive applications in edge clouds," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 779–792, Jan./Feb. 2021.

[14] X. Xu et al., "Service offloading with deep Q-network for digital twinning-empowered Internet of Vehicles in edge computing," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1414–1423, Feb. 2022.

Biographies

Xiaobin Tan (Member, IEEE) (xbtan@ustc.edu.cn) received the B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC), China, in 1996 and 2003, respectively. He is currently working as an Associate Professor with the Department of Automation, School of Information Science and Technology, USTC. His research interests include future network architecture, in-network computing, and multimedia communication.

Qiushi Meng (qsmeng@mail.ustc.edu.cn) received the B.S. degree from the Central South University (CSU), Changsha, Hunan, China, in 2022. He is currently pursuing the M.S. degree with the Laboratory for Future Networks, University of Science and Technology of China (USTC). His research interests include service deployment and task schedule in mobile edge computing scenarios.

Mingyang Wang (wmy903@mail.ustc.edu.cn) received the B.S. degree from the Department of Automation, Anhui University, in 2020. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China (USTC). His research interests include task scheduling, edge computing, cloud native, digital twin, etc.

Quan Zheng (Member, IEEE) (qzheng@ustc.edu.cn) received the B.S. degree in production process automation from the Dalian University of Technology, Dalian, Liaoning, China, in 1992, the M.S. degree in automatic control theory and application from the University of Science and Technology of China (USTC), Hefei, Anhui, China, in 1995, and the Ph.D. degree in computer software and theory from USTC in 2003. He is currently an Associate Professor with the Department of Automation and the Deputy Director of the Laboratory for Future Networks. His research interests include video semantic retrieval, media content distribution, video quality detection, and future networks.

Jun Wu (Senior Member, IEEE) (wujun@fudan.edu.cn) received the B.S. degree in information engineering and the M.S. degree in communication and electronic system from Xidian University in 1993 and 1996, respectively, and the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications in 1999. He is a Full Professor with Fudan University. He was a Professor with Tongji University. He served as a Principal Scientist with Huawei and Broadcom before joining Tongji. His research interests include wireless network, machine learning, and signal processing.

Jian Yang (Senior Member, IEEE) (jianyang@ustc.edu.cn) received the B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC), China, in 2001 and 2005, respectively. He is currently working as a Professor with the Department of Automation, School of Information Science and Technology, USTC. His research interests include future network architecture, multimedia over wired/wireless networks, and stochastic optimization theory and algorithm.