Intent-Based Management for Software-Defined Vehicles in Intelligent Transportation Systems

Yiwen Shen*, Yoseop Ahn*, Mose Gu*, and Jaehoon (Paul) Jeong*
*Department of Computer Science & Engineering,
Sungkyunkwan University, Suwon, Republic of Korea
Email: {chrisshen, ahnjs124, rna0415, pauljeong}@skku.edu

***Abstract*—Software-Defined Vehicle (SDV) is a new player towards autonomous vehicles in smart road networks. An SDV is constructed by a software platform like a cloud-native system like Kubernetes and has its internal network. To facilitate the easy and efficient configuration of networks in the SDV, an intent-based management is an appropriate direction. In this paper, we introduce a framework of intent-based management for networks and applications in SDVs so that they can communicate with other SDVs and infrastructure nodes for safe driving and infotainment services in the road networks. The proposed framework can also be extended to configure security policies for SDVs. We show a preliminary implementation of the proposed framework and suggest several research issues and challenges when designing such a framework.**

***Index Terms*—Software-Defined Vehicle, Intent-Based Management, Vehicular Networks, Intelligent Transportation Systems, 5G V2X.**

I. INTRODUCTION

In the past decades, the network management has been evolving dramatically from manual configuration to advanced automatic management. This evolution leads to the intent-based network (IBN) management and automation [1], which has been driven by several factors, including complexity of networks, scale, cost and efficiency, dynamic environments, service delivery, and security [2]. An "intent" to manage networks means that an network administrator only needs to give what to accomplish instead of how to achieve it. This paradigm shift greatly alleviates network management burdens stemmed from the ever growing complexity of networks. If a network allows an user (i.e., administrator) to manage it by intents, then we call this kind of network an IBN. A system that manages the IBN is called an intent-based system (IBS) [1]. The IBS has gained significance due to the complexity and scalability of modern networks. It offers efficiency by minimizing manual intervention, reducing costs, and enabling rapid adaptation to dynamic changes. IBN automation leverages advanced technologies such as artificial intelligence (AI), machine learning (ML), and software-defined networking (SDN) to automate network management. It facilitates agile service delivery, provisioning, and helps enforce security policies [3] and compliance [4].

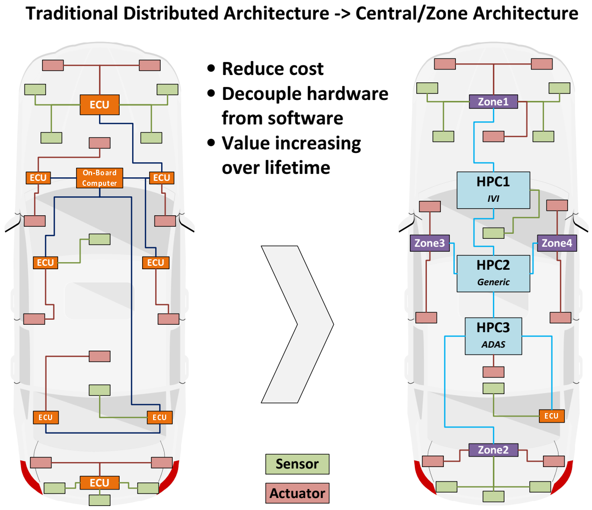

Fig. 1. Vehicle Architecture Transition from Discrete Architecture to Central/Zone Architecture.

Apart from network management and automation, the automotive industry is also witnessing a fundamental transformation, particularly with the advent of software-defined vehicles (SDVs). Traditional automotive electrical/electronic (EE) architecture was based on discrete electronic control units (ECUs) responsible for specific functions such as engine control, braking, and in-vehicle infotainment (IVI), as shown in Fig. 1. These ECUs communicate through various bus systems like controller area network (CAN), local interconnect network (LIN), or FlexRay. As vehicle complexity increased, there was a shift towards centralized architectures, where fewer, more powerful ECUs control multiple functions. This reduces wiring complexity and weight, improving reliability and manufacturability. With the rise of SDVs, the distinction between hardware and software functions becomes more blurred. SDVs leverage powerful onboard high-performance computers (HPCs) and a high-speed network backbone, typically Ethernet-based Internet Protocol (IP) network [5], to enable flexible and dynamic allocation of functions and resources. SDVs can also employ virtualization/containerization technology to consolidate multiple software functions, including virtualized network functions (VNFs), onto a single hardware platform [6]. This consolidation reduces the number of physical ECUs as well as network devices (e.g., switches and routers) required, leading to cost savings, weight reduction, and improved energy efficiency. Moreover, virtualization enables the isolation of software applications, enhancing security and reliability. The Connected Vehicle Systems Alliance (COVESA) [7] has developed a common vehicle signal specification (VSS) to

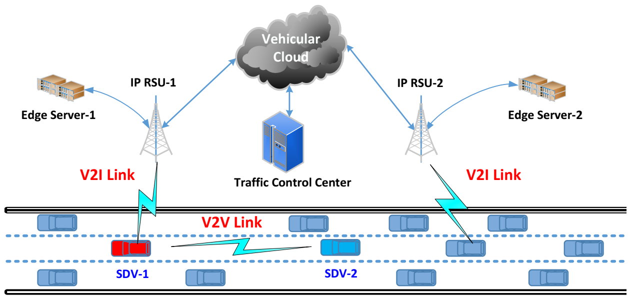

Fig. 2. Vehicular Networks for Software-Defined Vehicles.

represent the vehicle data for vehicle information services used by both in-vehicle and vehicle-to-cloud networks. VSS defines a taxonomy of vehicle signals that can include different vehicle components with attributes, sensors, and actuators, e.g., chassis, fuel system, steering wheel, and battery information for electric vehicles. With the VSS definition, one can better allocate network resources for vehicles in different scenarios.

When integrating IBN and SDVs, it is imperative to investigate an IBS for SDVs that considers both in-vehicle IP network and vehicle to everything (V2X) networks. As automotive applications, e.g., advanced driver-assistance systems (ADAS), automatic emergency braking (AEB), forward collision warning (FCW), and lane keeping assist (LKA) applications, inside an SDV become more various, data communications via VNFs among those in-vehicle applications also proliferate along with inter-vehicle communications by vehicle-to-vehicle (V2V) networks and communications with cloud (or edge) services by vehicle-to-infrastructure (V2I) networks. Fig. 2 shows vehicular networks for SDVs, where they can communicate with each other and with the vehicular cloud. Moving to in-vehicle networks, as depicted in Fig. 3, Vehicle-1 as an SDV and EN-1 (i.e. Edge Network-1) as an edge server can communicate with each other. Hosts inside Vehicle-1 are automotive applications connected by an in-vehicle Ethernet network, which can be virtualized or containerized. EN-1 together with the vehicular cloud can provide multiple services and applications for Vehicle-1.

In this paper, we present a framework for the management of IBNs in SDVs. This framework automates the configuration and monitoring of networks and security in each SDV through a vehicular cloud and the SDV's mobile network. A user (i.e., an administrator) responsible for the management of SDVs can configure and monitor networks and security functions through intents. The user's intents are communicated to a controller in a vehicular cloud for SDVs. The controller interprets the intent and translates it into the corresponding high-level policy. This high-level policy is then delivered to an SDV controller responsible for managing a specific SDV. The SDV controller is responsible for translating the high-level policy into the corresponding low-level policy. This low-level policy is then communicated to an appropriate Service Function (SF) for a network service (e.g., router, DNS resolver, or firewall) or an application service (e.g., safe driver for an autonomous vehicle and navigator for a human driver) within the SDV.

The remainder of this paper is organized as follows. Section II discusses related work for IBN-based SDVs. Section III presents the primary design of the proposed IBN architecture for SDV. Section IV shows a preliminary implementation of the proposed architecture. Section V suggests a number of issues and challenges for IBN in SDVs. Section VI concludes this paper along with future work.

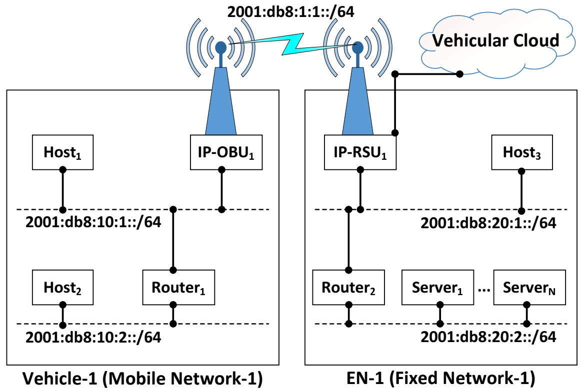

Fig. 3. In-Vehicle Network and Edge Network.

II. RELATED WORK

In this section, we introduce related work for IBN and SDVs. We first introduce the progress of IBN. Next, we discuss the industrial and standardization development for SDVs.

One of the most important driven factors for IBN is that networks have become immensely complex, with numerous interconnected devices, services, and protocols [2]. Managing such complexity manually is not feasible, leading to the need for IBN approaches. Another aspect is the scalability of modern networks, which spans across geographical locations and serves millions of users. This distinguishes the IBN to be more efficient at scale. Considering operational costs, IBN methods can minimize the need for human intervention, and thus improve efficiency by executing tasks faster and with fewer errors. Meanwhile, networks are dynamic, with changes occurring frequently [8] due to factors like user demands, traffic patterns, mobility, and security threats. IBN enables networks to adapt rapidly to these changes. The IBN is naturally involved with the advancements in technologies like AI, ML, and SDN, which have made automation more sophisticated and capable [9], [10]. With the increasing demand for new services and applications, IBN facilitates rapid service delivery and provisioning. Furthermore, IBN helps enforce security policies [3] and ensures compliance with regulations by consistently applying configurations and policies [4] across the network.

In the SDV field, a great number of initiatives have been proposed in the industry. The AUTomotive Open System ARchitecture (AUTOSAR) [11] organization has been extending its architecture design to integrate SDVs with cloud/edge services. The Scalable Open Architecture For Embedded Edge (SOAFEE) [12] aims at bringing the cloud-native development paradigm to the highly diverse, heterogeneous compute platforms for the next generation of automotive and safety critical systems. Eclipse SDV working group [13] facilitates

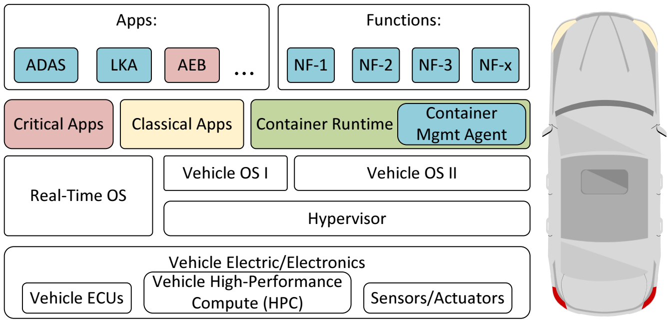

Fig. 4. A Vehicular Platform for SDV.

open-source development of automotive software, including software stacks and associated tooling for the core functionality of SDVs. The Connected Vehicle Systems Alliance (COVESA) [7] focuses on connected vehicles and their related technologies to unlock the full potential of connected vehicles for SDVs. Particularly, they design the vehicle signal specification to unify vehicle data models. Fig. 4 shows a typical SDV platform that integrates both classical and software-oriented architectures. Traditional vehicle E/E hardware devices, e.g., ECUs, sensors, and actuators, are grouped for vehicle HPCs, which can maximally explore the potential of the data generated from those hardware devices.

III. INTENT-BASED MANAGEMENT ARCHITECTURE FOR SOFTWARE-DEFINED VEHICLES

According to the life cycle design of IBN [1], Fig. 5 shows the life cycle of IBN for SDV management. It divides the life cycle into three spaces, namely user space, IBS space, and network operations & applications space. Each space is further divided into two sections, fulfillment and assurance. The fulfillment section pipelines the steps (i.e., intent input, translation/refinement, learning/planning/installation, and configuration/provision) toward the final service functions (SFs) such as network functions (NFs) and applications in SDVs. The assurance section monitors final results of the intent fulfillment to validate and analyze the resulted NFs and applications for SDVs. In this section, we present our intent-based management (IBM) architecture for SDVs that covers both NFs and applications in SDV environments.

A. IBS for Network Functions in SDVs

SDVs are managed and monitored by the vehicular cloud. They get help for software updates as well as the configuration of their networks and security from the vehicular cloud. Fig. 2 shows a vehicular network architecture for SDVs. SDVs as vehicles can communicate with each other via V2V and with infrastructure nodes such as IP Road-Side Unit (IP-RSU), for example, gNodeB in 5G networks, respectively. Edge servers can help SDVs to perform their safe driving by processing environmental data collected by the SDVs and giving maneuver guidance to the SDVs.

An SDV has its own internal networks (called in-vehicle networks), as shown in Fig. 3. The in-vehicle networks consist of multiple subnets connected with each other through routers.

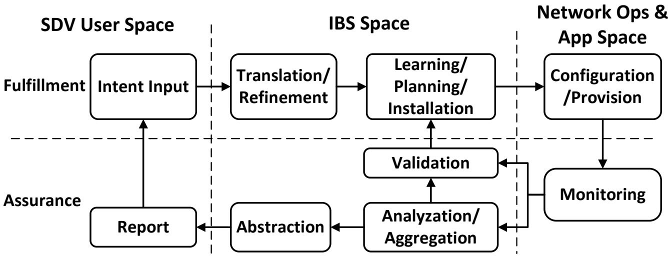

Fig. 5. The Life Cycle of IBS for SDV Management.

IP On-Board Unit (IP-OBU) is a network device in an SDV that has a basic processing ability and can be driven by a low-power CPU (e.g., ARM) with 5G Vehicle-to-Everything (V2X) communication device [14]. IP Road-Side Unit (IP-RSU) is a network device situated along the road as an infrastructure node. It has at least two distinct IP-enabled interfaces where one is for 5G V2X and the other is for the wired network connected to the vehicular cloud [14]. An Edge Network (EN) is a radio access network which has an IP-RSU for wireless communication with other SDVs having an IP-OBU and wired communication with other network devices (e.g., routers, IP-RSUs, and edge servers) [14]. As shown in Fig. 3, the IPv6 prefixes should be configured for the in-vehicle network (called mobile network) and Edge Network (called EN). Also, for V2X IP networking, the wireless interfaces of IP-OBU and IP-RSU should be configured with appropriate IPv6 network prefixes and default gateways towards the infrastructure network connected to the vehicular cloud.

For the automatic network configuration of SDVs, an intent-based management is required between the vehicular cloud and SDVs. Fig. 6 shows a framework of intent-based management for SDVs. The framework consists of a vehicular cloud and SDVs. The vehicular cloud consists of SDV User, Cloud Controller, SDV Database, Cloud Analyzer, and Cloud Vendor's Management System.

- SDV User: It is the software (e.g., web-browser-based user interface) used by SDV administrators to deliver network intents to SDV controllers. In the 3GPP intent-driven management service document [15], it is assumed that network intent is configured by the intent data model.
- Cloud Controller: It is a component that controls and manages other system components of the vehicular cloud. From a security point of view, a security service policy can be transmitted to the service function (SF) by converting the SDV User's security service intent into the corresponding security service policy and selecting an SF that provides an appropriate security service.
- Cloud Vendor's Management System: It is a component that provides images of virtualized SFs for vehicular cloud services and registers the SFs and access information with Cloud Controller.
- Cloud Analyzer: It gathers and evaluates monitoring data from SDV Analyzers to ensure the functionality and performance of SFs, e.g., the network data analytics function (NWDAF) in 5G networks.
- SDV Database: It is a database for managing SDVs,

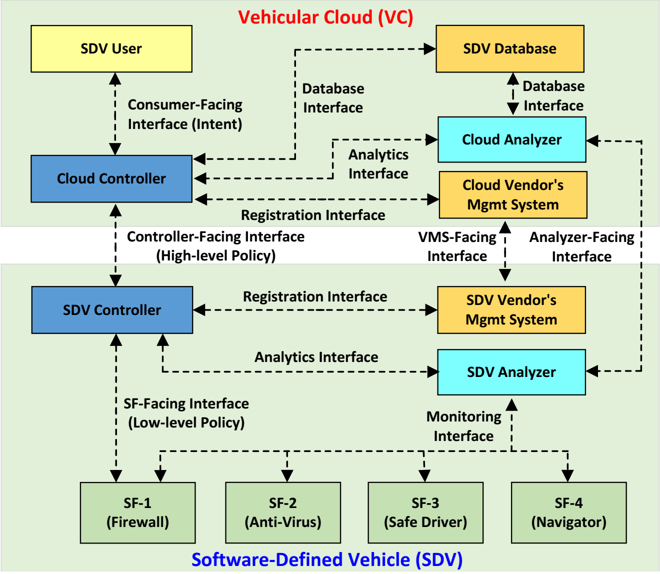

Fig. 6. Intent-Based Management Framework for SDVs.

including network and security configuration information of SDVs, current location and navigation path of SDVs, etc.

The SDV is composed of SDV Controller, SDV Analyzer, SDV Vendor's Management System, and Service Functions (SFs) such as NFs (e.g., router, DNS server, firewall, and anti-virus) and applications (e.g., safe driver and navigator).

- SDV Controller: It is a component that controls and manages other components of the SDV framework. It translates the high-level policy received from the Cloud Controller into a low-level policy that the SF can understand. An SF to perform this low-level service policy is selected, and the policy is transmitted to the SF.
- SDV Vendor's Management System: It is a component that provides an image of a virtualized SF for SDV services to the SDV framework and registers the function and access information of the SF with SDV Controller.
- Service Function: It is a component that refers to a virtual network function (VNF), cloud native network function (CNF), or physical network function (PNF) for a specific service. For security services, it provides security services such as firewalls, web filters, DDoS attack mitigators, and anti-viruses. In addition, networks and application services can also operate as SFs.
- SDV Analyzer: It is a component that collects monitoring data from SFs of SDVs and analyzes these data to confirm the activity and performance of SFs. SDV Analyzer acts as NWDAF in a 5G network. If there are problems (e.g., security attacks, traffic congestion, QoS degradation) in the SDV internal network, SDV Analyzer delivers either policy reconfiguration or feedback information to SDV Controller for security and network troubleshooting.

In Fig. 6, we also define a number of interfaces between a pair of system components in the vehicular cloud and SDV, respectively. These interfaces include Consumer-Facing Interface, Controller-Facing Interface, SF-Facing Interface, Registration Interface, Monitoring Interface, Analytics Interface, Analyzer-Facing Interface, VMS-Facing Interface, and Database Interface, which are described as follows.

- Consumer-Facing Interface: It is an interface between SDV User and Cloud Controller for conveying intents.
- Controller-Facing Interface: It is an interface between Cloud Controller and SDV Controller for high-level policy delivery with translated intents.
- SF-Facing Interface: It is an interface between SDV Controller and SF for the delivery of a translated lower-level policy.
- Registration Interface: It is an interface used to transfer SF capabilities and access information for registration to either Cloud Controller or SDV Controller, or deliver SF queries for searching the requested SFs. This interface can be an interface between Cloud Controller and Cloud Vendor's Management System (Cloud VMS), or between SDV Controller and SDV Vendor's Management System (SDV VMS).
- Monitoring Interface: It is an interface between the SF and the SDV Analyzer used to collect the SF's monitoring data to identify SF-related security, system, and network issues.
- Analytics Interface: It is an interface for delivering policy reconfiguration or feedback as a result of analyzing SF monitoring data. This interface is an interface between SDV Analyzer and SDV Controller, between SDV Analyzer and Cloud Analyzer, or between Cloud Analyzer and Cloud Controller.
- Analyzer-Facing Interface: It is an interface between SDV Analyzer and Cloud Analyzer for the exchange of security, network, and system-related analysis of SFs.
- VMS-Facing Interface: It is an interface between Cloud VMS and SDV VMS to exchange SF container images with SF feature information.
- Database Interface: It is an interface for exchanging data in an SDV database. It is an interface between SDV Database and Cloud Controller, or between SDV Database and Cloud Analyzer.

We can design the intent, high-level policy, and low-level policy by XML documents [16] or YAML documents [17]. They can be delivered to the destination components via NETCONF [18], RESTCONF [19], or REST API [20].

B. IBS for Applications in SDVs

By the proposed architecture, an IBS can also manage applications in SDVs. SDV applications can include safe driver (e.g., AI driver) for an autonomous vehicle and navigator for a human driver.

Safe driver in an SDV can make decision in driving such as steering angle and acceleration/deceleration of the SDV with environmental sensing data from its cameras and LiDAR. Also, the safe drivers in SDVs can collaborate with each to avoid physical collisions by exchanging their sensing data and mobility data via wireless communications such as 5G V2X

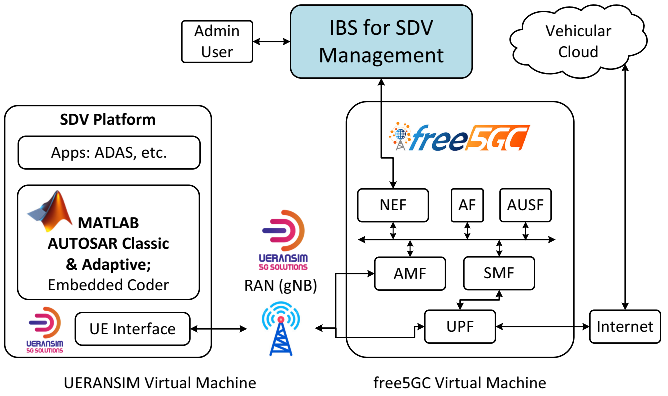

Fig. 7. An IBN Testbed for SDV Management.

communication. This safe driver can use a Context-Aware Navigation Protocol (CNP) [21] that cooperates with other safe drivers.

Navigator in an SDV can guide the SDV through a road network with an efficient navigation path. It can actively interact with a vehicular cloud for such an efficient navigation path in an autonomous way. Navigators in SDVs report their navigation paths to the vehicular cloud, so the vehicular cloud has up-to-date navigation information of the SDVs and can play a role of the coordinator for the SDVs so that they can take less congested paths. This navigator can use a Self-Adaptive Interactive Navigation Tool (SAINT) [22] for an SDV that is connected to the vehicular cloud.

C. *Examples of IBS in SDV Environments*

To give a more clear concept of IBS for SDV, here we design a number of examples of using the proposed architecture of IBS for SDV management.

One scenario can be that an automotive company needs to upgrade and install new applications on a group of automobiles sold to customers, i.e., over-the-air (OTA) update. The container images of the upgrading and installation applications have already been tested for deployment. Along with these applications, new NFs shall also be deployed for the new data traffic requirements in the OTA update, such as bandwidths, jitters, and delays. An SDV User in the automotive company can issue a request like "Please upgrade and install <application A> to the cars."

As another scenario, an SDV fleet management company needs to regroup their automobiles for an event to transport people from multiple locations. For the event, an SDV User in the company can give an intent like "The automobiles need to transport people among locations A, B, C, and D from next Monday to Wednesday." The IBS in the company will orchestrate multiple functions, policies, and network resources for both in-vehicle and V2X networks.

IV. IMPLEMENTATION

We implemented a preliminary IBN testbed for SDV management to show a proof of concept (PoC). This PoC uses an open-source cellular network framework to connect SDVs.

As shown in Fig. 7, for the constructed SDV platform, we use the AUTOSAR modules (i.e., Classic and Adaptive platform) of MATLAB to generate an automotive application basis with a simulated UE interface based on UERANSIM. The AUTOSAR modules can host ADAS having multiple safe driving applications, e.g., FCW, adaptive cruise control, and blind spot monitor. We also use UERANSIM to build a simulated RAN (e.g., gNodeB in 5G mobile networks) that is connected with the User Plane Function (UPF) and the Access and Mobility Management Function (AMF) of the open source 5G core system (5GCS) of free5GC [23]. We place an IBS for SDV management (ISM) outside the 5GCS that accesses the Network Exposure Function (NEF). The ISM module is a system that receives and translates intents, and enforces a series of network policy configurations toward SDVs and 5GCS through the NEF. The NEF in 5GCS provides secure APIs for any third-party applications to access the functions inside 5GCS. In our design, the ISM uses the APIs of NEF to send the translated low-level policies toward the policy control functions (PCF) in 5GCS. The PCF consolidates all the required configurations and installs them into 5GCS for serving SDVs. Note that the ISM can also be co-located with the vehicular cloud. In our 5G testbed, we have tested the connectivity between the UE interface of the SDV platform and the data network (i.e., the Internet) through the 5GCS, which is demonstrated online [24]. We will implement the above functionality in our IBS in future.

V. RESEARCH ISSUES AND CHALLENGES

Based on what we have discussed so far, designing an IBS for SDVs can face several issues and challenges.

- Data collection and analysis: To provide more agile networks for different applications (e.g., IVI and collaborative sensing) among SDVs, a challenge can arise on how an IBS collects and analyzes data generated from SDVs for intelligent service configurations.
- Domain-specific translation: From the intent user perspective, an intent translation engine shall have an SDV-domain natural language processing (NLP) ability so that intents can be correctly translated into an SDV-domain service configuration plan.
- Intent validation: When an intent has been translated into policies and configurations, which are further executed in the target platforms/devices, how to validate that the configured platforms/devices generate the desired effects based on the intent is an interesting topic.
- Intent conflict detection and resolution: When multiple administrators send their intents into an IBS, how to detect any possible conflicts among those intents, and if any conflicts exist, how to resolve those conflicts are critical issues during the execution of intents.
- Manufacturer agnosticism: SDVs from multiple manufacturers can have different designs for in-vehicle applications and functions. It is a challenge task to develop a standard manufacturer-agnostic orchestration and management tool chains for managing SDVs.

We summarize the research issues and challenges in Table I

TABLE I
RESEARCH ISSUES AND CHALLENGES

Topic	Issues and Challenges
SDV-domain NLP engine for intent translation	To correctly translate a user's intents in SDV context, an IBS for SDV management shall have the knowledge of SDV-domain device configuration.
Data collection and analysis of SDVs	Collecting and analyzing the data generated from SDVs for IBN management to provide more agile networks for different applications (e.g., IVI and collaborative sensing) is a challenge task,
Validation for the resulted configuration from intents	When a service configuration plan has been executed, how to validate the plan whether it has fulfilled the intents from the user or not is another important research issue.
Intent conflict detection and resolution	Detecting any possible conflicts among those intents, and if any conflicts exist, resolving those conflicts are critical issues during the execution of intents.
Manufacturer agnosticism	Developing a standard manufacturer-agnostic orchestration and management tool chains for managing SDVs.

VI. CONCLUSION

In this paper, we have introduced a framework of intent-based management for networks and applications in Software-Defined Vehicles (SDV). Through the proposed framework, both virtualized network functions and applications can be efficiently orchestrated for agile network resource reconfigurations and flexible SDV application updates. We showed a preliminary implementation for the proposed framework and suggested a number of research issues and challenges. As future work, we will design and implement the system components and interfaces in the proposed framework for SDVs. At the same time, we will investigate an intent assurance mechanism for the proposed framework.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government, Ministry of Science and ICT (MSIT) (No. 2023R1A2C2002990), and by Basic Science Research Program through the NRF of Korea funded by the Ministry of Education (No. 2022R1I1A1A01053915). Note that Jaehoon (Paul) Jeong is the corresponding author.

REFERENCES

[1] A. Clemm, L. Ciavaglia, L. Z. Granville, and J. Tantsura, "Intent-Based Networking - Concepts and Definitions," RFC 9315, Oct. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9315

[2] A. Leivadeas and M. Falkner, "A Survey on Intent-Based Networking," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 625–655, 2023.

[3] J. Kim, E. Kim, J. Yang, J. Jeong, H. Kim, S. Hyun, H. Yang, J. Oh, Y. Kim, S. Hares, and L. Dunbar, "IBCS: Intent-Based Cloud Services for Security Applications," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 45–51, 2020.

[4] P. Lingga, J. Jeong, J. Yang, and J. Kim, "SPT: Security Policy Translator for Network Security Functions in Cloud-Based Security Services," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–14, 2024.

[5] J. Jeong, Y. Shen, T. Oh, S. Céspedes, N. Benamar, M. Wetterwald, and J. Härri, "A Comprehensive Survey on Vehicular Networks for Smart Roads: A Focus on IP-based Approaches," *Vehicular Communications*, vol. 29, p. 100334, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209621000036

[6] D. F. Blanco, F. Le Mouël, T. Lin, and M.-P. Escudié, "A Comprehensive Survey on Software as a Service (SaaS) Transformation for the Automotive Systems," *IEEE Access*, vol. 11, pp. 73 688–73 753, 2023.

[7] The Connected Vehicle Systems Alliance (COVESA). [Online]. Available: https://covesa.global/

[8] M. Avgeris, A. Leivadeas, N. Athanasopoulos, I. Lambadaris, and M. Falkner, "Model Predictive Control for Automated Network Assurance in Intent-Based Networking enabled Service Function Chains," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, 2023, pp. 1–7.

[9] J. J. D. Rivera, M. M. S. Sarwar, S. Alam, T. A. Khan, M. Afaq, and W.-C. Song, "An Intent-Based Networking mechanism: A study case for efficient path selection using Graph Neural Networks," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, 2023, pp. 1–6.

[10] A. Collet, A. Banchs, and M. Fiore, "Lossleap: Learning to predict for intent-based networking," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 2138–2147.

[11] AUTomotive Open System ARchitecture (AUTOSAR) Adaptive Platform. [Online]. Available: https://www.autosar.org/standards/adaptive-platform

[12] Scalable Open Architecture for Embedded Edge (SOAFEE). [Online]. Available: https://www.soafee.io/

[13] Eclipse Software Defined Vehicle Working Group. Eclipse Software Defined Vehicle Working Group Charter. [Online]. Available: https://www.eclipse.org/org/workinggroups/sdv-charter.php

[14] J. P. Jeong, "IPv6 Wireless Access in Vehicular Environments (IPWAVE): Problem Statement and Use Cases," *RFC 9365*, Mar. 2023. [Online]. Available: https://datatracker.ietf.org/doc/rfc9365/

[15] 3GPP, "Management and orchestration; Intent driven management services for mobile networks," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 28.312, September 2023, version 18.1.1. [Online]. Available: http://www.3gpp.org/DynaReport/28312.htm

[16] M. Bjorklund, "The YANG 1.1 Data Modeling Language," *RFC 7950*, Aug. 2016. [Online]. Available: https://datatracker.ietf.org/doc/rfc7950/

[17] I. Brian, C. E. Clark, and O. Ben-Kiki, "Yet Another Markup Language (YAML) 1.0," October 2023. [Online]. Available: https://yaml.org/spec/history/2001-05-26.html

[18] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network Configuration Protocol (NETCONF)," *RFC 6241*, January 2011. [Online]. Available: https://datatracker.ietf.org/doc/rfc6241/

[19] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol," *RFC 8040*, Jan. 2017. [Online]. Available: https://datatracker.ietf.org/doc/rfc8040/

[20] R. T. Fielding and R. N. Taylor, "Principled Design of the Modern Web Architecture," *ACM Transactions on Internet Technology*, vol. 2, no. 2, May 2002. [Online]. Available: https://doi.org/10.1145/514183.514185

[21] B. A. Mugabarigira, Y. C. Shen, J. P. Jeong, T. T. Oh, and H.-Y. Jeong, "Context-Aware Navigation Protocol for Safe Driving in Vehicular Cyber-Physical Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 128–138, Jan. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9921182

[22] J. P. Jeong, H. Jeong, E. Lee, T. T. Oh, and D. H. Du, "Self-Adaptive Interactive Navigation Tool for Cloud-Based Vehicular Traffic Optimization," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 4053–4067, Jun. 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7243355/

[23] V. Jain, H.-T. Chu, S. Qi, C.-A. Lee, H.-C. Chang, C.-Y. Hsieh, K. K. Ramakrishnan, and J.-C. Chen, "L25GC: a low latency 5G core network based on high-performance NFV platforms," in *Proceedings of the ACM SIGCOMM 2022 Conference*, ser. SIGCOMM '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 143–157. [Online]. Available: https://doi.org/10.1145/3544216.3544267

[24] Demo for Testing Connectivity Between a UE interface of a SDV Platform and the Internet. [Online]. Available: https://youtu.be/881ji0AsN7c?si=l-dMQDEG9Rg6Pscf