Application-centric network management – Addressing safety and real-time in V2X applications

ROLF ERNST, TU Braunschweig, Germany
DOMINIK STÖHRMANN, TU Braunschweig, Germany
ALEX BENDRICK, TU Braunschweig, Germany
ADAM KOSTRZEWA, TU Braunschweig, Germany

The current roadmaps and surveys for future wireless networking typically focus on communication and networking technologies and use representative applications to derive future network requirements. Such a benchmarking approach, however, does not cover the application integration challenge that arises from the many distributed applications sharing a network infrastructure, each with their individual topology and data structure. The paper addresses V2X networks as an important example. Crucial end-to-end application constraints including real-time and safety encourage a closer look at application interference and systematic integration. This perspective paper proposes a two-layer resource management that divides the problem into an application integration and a network management task. Valet parking with high-resolution infrastructure camera support is elaborated as a use case that overarches vehicle network and wireless network management. Experiments demonstrate the benefits of complementing the current network-centric management by an application-centric integration.

CCS Concepts: • **Computer systems organization → Embedded and cyber-physical systems**; **Real-time systems**; **Dependable and fault-tolerant systems and networks**; • **Networks → Wireless access networks**; **Network reliability**.

Additional Key Words and Phrases: Application-aware Resource Manager, Resource Manager, Network, RAN, V2X

1 INTRODUCTION

1.1 Motivation

Recently numerous roadmaps and surveys on the future of 5G and 6G networks appeared addressing the network capabilities, parameters, and applications. Many of them involve vehicles as a mobile platform or as a subject in automated driving and related services.

Examples can be found in 3GPP [2] [3] and 5GAA [4] standards and roadmaps. The roadmaps of 5G Automotive Association (5GAA) [4], GSMA [13] or VDA [31] include future use cases, e.g. valet parking, platooning, co-operative driving or tele-operated driving, which are enabled by emerging wireless technologies. Similarly, surveys addressing future wireless networks, e.g. [10] [12] [28] [29] [32], use such applications for motivation, to derive requirements, and to outline future trends in networking technology.

Authors' addresses: Rolf Ernst, ernst@ida.ing.tu-braunschweig.de, TU Braunschweig, Braunschweig, Germany; Dominik Stöhrmann, stoehrmann@ida.ing.tu-braunschweig.de, TU Braunschweig, Braunschweig, Germany; Alex Bendrick, bendrick@ida.ing.tu-braunschweig.de, TU Braunschweig, Braunschweig, Germany; Adam Kostrzewa, kostrzewa@ida.ing.tu-braunschweig.de, TU Braunschweig, Braunschweig, Germany.

1539-9087/2022/4-ART $15.00
https://doi.org/10.1145/3528411

The surveys typically take a network-centric view, discussing wireless communication opportunities all the way from the physical medium to protocols to network optimization thereby considering quality metrics of latency and throughput, reliability, or age of information. In that network-centric view, the vehicles and their applications are customers that require and receive communication service. Resulting application requirements are discussed per application to determine their feasibility and give orientation to wireless technology development. Such application benchmarks are helpful to evaluate and compare technologies, but they omit the important aspect of application integration. Future vehicles will integrate an increasing number of applications that use the same wireless communication channels. These applications have different requirements, importance and criticalities, and they involve a different number of participating vehicles. Moreover, integration will be vehicle specific, just as the applications in today's smartphones. In consequence, each vehicle will have a different and dynamically changing communication profile. Other than in smartphones, where side effects of communication resource sharing are largely tolerated, vehicles integrate wireless traffic that needs end-to-end real-time and safety guarantees. Roadmaps and surveys give ample application examples. Automated driving is just one challenging use case.

However, extending network management to end-to-end distributed application control would lead to a complex problem with many dependencies between network and application control. In this paper, we opt for a dual-layer approach of separated but interacting distributed network and application control. Layer interaction occurs in local application-aware resource managers paving the way for a simple modular architecture.

To motivate the solution, we will first reverse the established network-centric perspective and take a vehicle- and application-centric view in section 1. In section 2 we will then introduce the goal of application-aware resource management. Following in section 3 the application-based two-layer resource management components and protocol is shown in detail. Then, we continue with a valet parking use case in section 4 to show the configuration process in different network segments. Furthermore, a evaluation of V2X is given followed by an outlook in section 5 and research perspectives in section 5.2. Section 6 brings to an end with a conclusion.

1.2 Safety

While today's advanced driver assistance functions leave the responsibility with the driver, automated driving from SAE level 3 up becomes safety-critical. While there are good reasons to exploit V2X communication to improve automated driving performance, the wireless channel becomes safety relevant as soon as automated driving decisions depend on that channel. This does not mean that V2X is only useful for automated driving if it provides extreme reliability. It rather means that wireless communication must become part of a multi-faceted safety concept that, in turn, is governed by safety standards.

Safety always refers to a function. We generally distinguish two types of safety concerns. The first safety concern is a potential risk that arises from the intended functionality itself, such as an unwanted effect of a vehicle maneuver or an object detection that overlooked or misinterpreted an object on the scene. This concern is called Safety of the Intended Functionality (SOTIF) and is addressed in the standard ISO 21448 (for vehicles) [18]. Obviously, using V2X communication to get access to further environmental information, e.g. from infrastructure cameras or from the status and intention of other traffic participants will improve the SOTIF. The second safety concern is the risk that the implemented function is failing, because of, e.g., a software error or a component or connection failure. This traditional safety concern is called Functional Safety (FS), which, for vehicles, is addressed in the ISO26262 functional safety standard [19]. Because V2X provides far less reliability than in-vehicle communication, it increases the risk of functional failures compared to a function that only uses the vehicle resources. So, while an improved SOTIF is a strong incentive to include V2X communication towards safer automated driving, the increased risk of functional failures calls for a suitable functional safety concept. That functional safety concept should contain a (degraded) mode where the vehicle operates exclusively on its own resources without violating SOTIF concerns. We will give an example of valet parking later.

At this point, we note that a degraded local mode could also address IT security concerns arising from new intrusion vectors via V2X communication and function distribution. In such a degraded mode, Threat Analysis and Risk Assessment (TARA), such as defined in the standard ISO/SAE 21434 [17] could be focused on the individual vehicle. An investigation of how IT security could profit from an application control layer as proposed in this paper is left for future work.

1.3 Application Impact

The next issue is the impact of applications and their profiles. Applications are considered in all surveys to derive wireless network requirements and justify research directions and deployment scenarios [10] [12] [28] [29] [32].

Ongoing standardization derived basic wireless services for vehicle cooperation that require low data volume and can be universally employed on all vehicles, such as the SAE J3216 [20]. SAE J3216 focuses on information sharing and negotiation, ranging from status sharing and intent sharing to agreement seeking (negotiation) to driving control (prescriptive). The use of such services is open and not tightly linked to a specific application. Such standards follow the general trend towards service-oriented architectures.

The roadmaps and public announcements of large players working on vehicle-based services, such as Amazon [8], Microsoft [25], Huawei [15] and many more, show a focus on applications that involve several network segments. They reach from the vehicle network over the wireless channel, radio access network, edge computing, often to a cloud in the background. Such cases include (1) remote fleet management all the way to remote control of vehicles, or (2) location-based (navigation) services that are based on layered maps collecting traffic information with different dynamics. Such maps overlay static information (streets, signs) with transient information (construction, weather) and highly dynamic information such as hidden vehicles or pedestrians [6]. Those are just two of many examples that indicate were the commercial exploitation of V2X technology is likely to go. The examples are also interesting because of their requirements. Both applications (1) and (2) can only be safely deployed if the network guarantees continuity and a maximum latency across multiple network segments. How otherwise should a remotely controlled vehicle operate safely?

In all these cases, communication connects cooperating computers, each executing part of a larger application. Most applications explained in the roadmaps and surveys are examples of distributed embedded computing and data management, and are developed as such. From the distributed computing perspective, the many different communication segments interact to provide a coherent communication platform, subject to end-to-end QoS requirements, as outlined in figure. 1.

1.4 Network Management

But how to establish end-to-end QoS guarantees if the network segments have very different properties and are separately optimized? There is a trend to keep the largely independent network-centric optimization of network segments, but to coordinate their management for coherent communication services. A good example is the Automotive Edge Computing Consortium (AECC). AECC is a collaboration of major industrial players that raises awareness for the coordination challenges and develops suggestions for standardization of network segment interactions and coordination, cf. [5]. In effect, the architecture developed by AECC and the interaction of network segments defines what we can call a network control layer.

However, this first step will not be sufficient. A coherent network management must serve a multitude of end-to-end applications with different and potentially conflicting requirements, with different application owners and customers. Focused on network operation and segment interaction, it is likely that the current network-centric approach alone will not be able to effectively serve and coordinate the application needs, in particular when

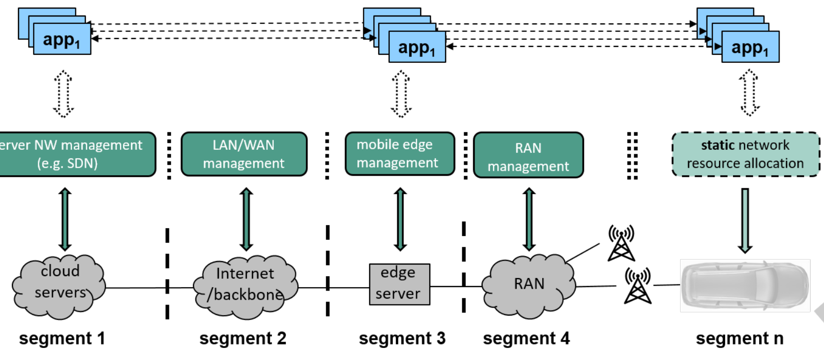

Fig. 1. Overview of the communication platform and its communication paths divided into different network segments, each controlled by a management layer to enable a data link between applications. This management is done by different providers which own and control network segments

it comes to integration of critical services. In the following, we will suggest an approach that addresses that challenge.

2 THE GOAL: AN APPLICATION-AWARE RESOURCE MANAGEMENT

2.1 A generic Resource Manager (RM)

The core proposal of this paper is the introduction of an application-aware resource management network that orchestrates end-to-end applications and their QoS requirements including prioritization, access and possibly degradation. It complements the management of network segments and their emerging coordination by an application control layer that manages application properties and requirements end to end across the system.

As a first step, the overall network of figure 1 is divided into regions, with each region corresponding to a network segment. Each network segment is complemented by and coupled with a component that manages application resource requests. This way, the original structure and hierarchy of network segments and their management with the RM is preserved. In figure 2 the network segments and the corresponding RM are shown. This illustrations is derived from figure 1 and provides a different perspective on the communications path of the highlighted application from figure 1. The RMs are connected to a network that is the basis for an end-to-end network and application management shown in figure 2. The connected RMs enable network segment coordination as, e.g., intended by the AECC. The formation of regions includes the vehicle network.

2.2 RM Architecture

Figure 3 shows the **generic architecture of an RM** and its interaction with network nodes and with other regions. The software components of distributed applications (app$_i$) are hosted on several network nodes that belong to the region of an RM. We may assume that the network nodes include some kind of network client (NW client) that controls network access. Such network access control bounds the communication load and is a basis for systematic network management.

The network client typically formulates resource requests, such as stream reservation requests known, e.g., from real-time versions of switched Ethernet protocols [16]. The network management may grant, reject or modify these requests. Today, a network client formulates requests at the level of network objects or parameters, such as packet size or data rates. Because network control parameters partly depend on the underlying network

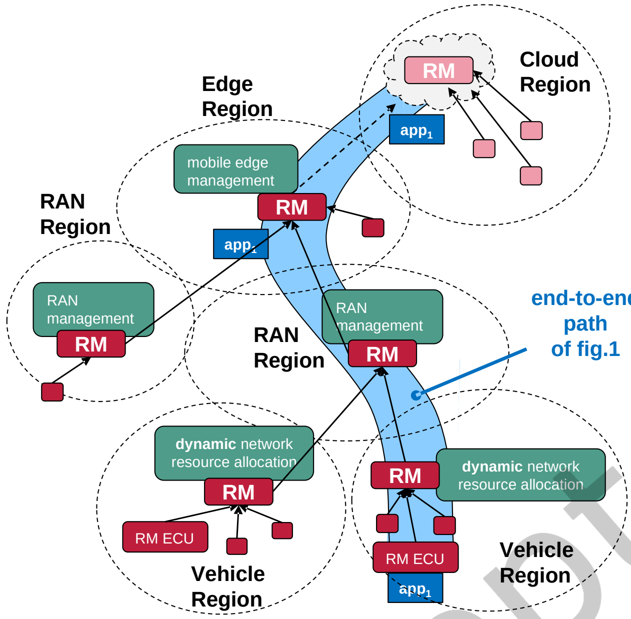

Fig. 2. Structure and interconnection of an example network consisting of different network regions, which are managed by RM management. The RM structure orchestrate the path of the example application which is presented in figure 1.

technology and management of a network segment, the network client protocols serving the components of a distributed application, as in figure 1, are network specific.

The application components, however, communicate at the level of data objects and control. The DDS (Data Distribution Service) standard [26] is a popular and important example. It synchronizes distributed databases using the Real-Time Publish-Subscribe (RTPS) protocol, that handles the actual data exchange in DDS. In the following we will refer to DDS and RTPS synonymously. The databases store application data objects and their history under so-called topics. DDS is independent and agnostic of the underlying network technology. Because DDS couples distributed databases, all protocols and all requests refer to end-to-end communication. Every application instance may have its own structure that involves different network segments and a dynamically changing number of participating end nodes (e.g. vehicles).

Translation between application requests & behavior and network requests & behavior is not the only challenge. The coherent adjustment of requirements and guarantees is a second one. Real-time and continuation requirements of distributed applications enforce real-time network management which appears overly difficult if based on current distributed negotiation and supervision.

2.3 RM coordination and application integration

Therefore, one goal is to **separate constraint resolution and optimization from run-time operation**. Constraint resolution and optimization precedes application communication and can, e.g., use the path of a service discovery mechanism to set up connections and prepare the adjustment of resource utilization. Only if this process has finished, the prepared adjustment may be applied. Activating the adjustment impacts run-time operation. To guarantee continuous operation during adjustment, we envision a pre-planned **coordinated transition process**. Coordinated transitions are subject to real-time constraints, but can exploit application properties, such as slack

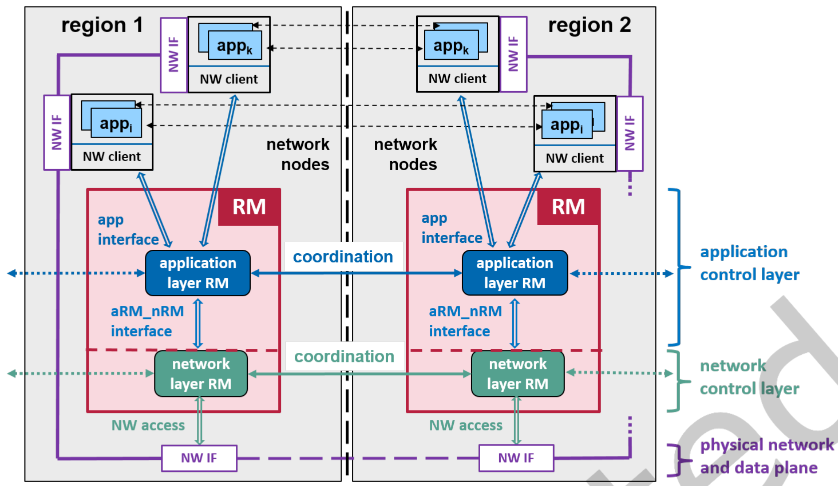

Fig. 3. Detailed view of two network regions with two network nodes each. The architecture of a network region is divided into the corresponding layer consisting of an RM and multiple applications connected with the RM. The RM is divided into two main parts (application layer RM and network layer RM). Both parts communicate with each other as well as with other RMs from other regions in the respective layer. The two regions in the figure will typically be in different levels of hierarchy (cp. fig 2).

in application timing, as demonstrated in earlier work [22]. That work shows that exploitation of application timing allows network state transitions that are not feasible in application agnostic network control. More details will be given later.

A second goal is the **integration of all application requirements**, including those propagated from other nodes, into a single set of local QoS request interacting with network management. A single set of QoS requests simplifies network management because the individual applications have very different resource requirements and communication profiles, are from different sources and have different owners, as we saw above. Applications will often be dynamically instantiated, so the combination of profiles will change. Finally, the applications are of mixed criticality and require an integration method that adheres to a common safety concept.

This multi-faceted application integration problem is known from vehicle networks, where design complexity is currently mitigated by static network configuration, determined at design time. Static configurations, however, do not scale to changing workloads with highly dynamic profiles, and the known concept of dynamic stream reservation neither guarantees transitions with continuous operation nor does it balance application requirements. Instead, a new adaptive integration solution that matches safety requirements will be needed.

2.4 Application and network control layers

Figure 3 shows that the RMs of connected regions communicate on two control layers. The network control layer is an established concept that is already used in Software Defined Networking (SDN). The second layer, the **application control layer** is shown on top of the **network control layer**. Like the network control layer, it uses the existing physical network for communication.

The application control layer consist of interacting local **application layer RMs (aRM)**. The aRMs distribute and coordinate the application communication and resource requests that they receive from applications by propagating these requests among the involved nodes. From a networking perspective, such a layer of cooperating aRMs has several advantages:

Table 1. Interfaces of the aRM and the detailed message type and corresponding function

Unit	IF ID	Function	Description
aRM		**Internal function**	**integrate the different app requests according to a system level application strategy**
Coordination Interface	A1.S	Setup	set up connection to other aRMs that serve an end-to-end application
	A1.R	Request	request communication service from RM of connected region
	A1.G	Grant	grant communication service to RM of connected region
App Interface (API)	A2.R	Request	QoS requests of applications
	A2.A	Adjust	adjust the network access parameters of NW interface and applications
	A2.G	Grant	grant QoS requirements of an app, optionally with limited guarantees
	A2.N	Notify	notify individual apps asap if NW guarantees cannot be met (reject or connection failure)
aRM_nRM Interface	A3.R	Request	integrated app QoS requests
	A3.A	Adjust	adapt app integration to changing network guarantees
	A3.G	Grant	grant QoS requirements of integrated apps
	A3.N	Notify	notify if NW guarantees cannot be met (reject or connection failure)

- Application requirement localization: The application control layer coordinates all distributed applications and their requirements, eventually mapping them to only local requirements.
- Application integration: It integrates and bundles all application requests in a single network requirement interface thereby offloading network management from serving all application requirements on the network level including requests from other parts of the network
- Application control localization: The management of a network segment has a single entry point to application control. Changes of network parameters, such as connection degradation or improvement, are communicated to a single component. Communication with other network segments can stay on the level of network management without encoding of user requests.

The next section will outline how the aRMs can be connected and combined in an overarching cooperative resource management.

3 THE TWO LAYERS OF RESOURCE MANAGEMENT

3.1 Components and protocols

The application control layer introduced in the last section does not replace but complements an existing network control layer. The reason for the dual control layer approach is that network management follows its own rules coordinating network resources rather than applications. Network management is an established engineering discipline with technology and provider specific mechanisms that will not be re-invented for the sake of distributed applications. It will already be difficult enough to establish a control layer that extends beyond a single network segment, such as e.g. intended by the AECC consortium. Hence, a two-layer mechanism is a natural solution. Given the goals of the last section, efficient coupling of the two layers is essential for network supervision, in particular for real-time transitions between network states without disruption, i.e. without losses or overflows or unacceptable delays. To tightly couple application control and network control, we integrate them in the RMs. As figure 3 shows, each resource manager (RM) combines the local aRM with a network layer resource manager (nRM). All interactions between the layers occur inside an RM, between aRM and nRM. Because, physically, application control layer communication uses the existing network that is also utilized by network control, no new communication paths are needed. If the traffic of both layers is combined, only one protected channel with real-time guarantees is required for communication supervision.

To give a better impression of the approach, we will further detail the component interaction by elaborating the interfaces and their messages. These interfaces are just one example of a possible implementation that is also employed in the use case given later.

Table 2. Interfaces of the nRM and the detailed message type and corresponding function

Unit	IF ID	Function	Description
nRM		**Internal function**	**integrate and orchestrate all network components and their parameters of a network segment – as in *existing* networks**
Coordination Interface	N1.C	Setup	use *existing* network management to coordinate network management of connected regions
aRM_nRM Interface	A3.R	Request	integrated app QoS requests
	A3.A	Adjust	adapt app integration to changing network guarantees
	A3.G	Grant	grant QoS requirements of integrated apps
	A3.N	Notify	notify if NW guarantees cannot be met (reject or connection failure)
Network Access	N2.C	Control	control: adjust network parameters, routes, …
	N2.M	Monitor	monitor network health status, …

Tables 1 and 2 summarize the proposed implementation of RM components and their interaction. The first column shows the unit as shown in figure 3. The second column is the ID (IF ID) of an interface function that is described in the third column. As an example, A1.S is the setup function of the Application control layer interface. The function description is depicted in the fourth column. We will use the IF IDs later in the use case to identify the different messages in the elaborated protocols. For the sake of completeness, the tables also denote the internal functions.

The aRM has three interfaces, the application and network client interface (API), the coordination interface to other aRMs in the application control layer, and the interface to the local network resource manager (nRMs). The **Application Interface (API)** follows a contract-based approach. An application that wants to use network resources must formulate **QoS requests**. The requests define the expected network guarantees and application objectives. Examples are the streams to be communicated including the object size (bounds/pattern) and type (video, …), a stream profile (e.g. work load or arrival curve), deadlines for object transmission, action on deadline miss (e.g. notification), error protection, an end-to-end safety concept (incl. requested communication continuation, …), criticality, or security requirements. The aRM can **grant or reject** the requests. If accepted, the network might later be forced to withdraw the guarantees, e.g., due to a network failure. In this case, the aRM must **notify** the applications, because a functional safety concept can require transition to a degraded mode, as explained in the beginning. However, only controlling application access is not sufficient, because of delays and inaccuracy in software control. Therefore, the application interface also includes messages to NW clients to **adjust network access parameters**, such as shapers or network access schedules.

The second interface, **the Coordination Interface**, connects the aRMs on the application control layer. As introduced before, it serves two types of messages, the data needed for connection setup and for cooperative resource assignment and the real-time and error protected messages for coordinated transitions. Definition of messages for connection setup and cooperative resource assignment is left to the selected algorithm, while the protocol for coordinated transitions is part of the run-time operation and must be unified to enable continuity and real-time guarantees. An example is given in [27].

The third interface, the **aRM_nRM Interface**, connects the aRM with the local network resource management, the nRM. The aRM bundles all application requirements transformed into a set of network QoS requirements for a single interface with the nRM. That bundle includes all communication on the application control layer. From a network management perspective, the application control layer is just another distributed application. The nRM returns the network status as a single interface to the application control layer.

The **nRMs** enable and coordinate network management. Because the aRMs of the application control layer bundle the application requests per RM, the nRMs and their network control layer can integrate and orchestrate all NW components and their parameters of a network segment in a network-centric optimization, as it is done in current networks. If the combined application requirements cannot be met by the current network status, then they are not granted and the aRM must find a solution with reduced resource utilization. If a network service fails or communication quality drops, the nRM can notify the aRM and actively request an action on the application control layer. Similarly, the aRM can be notified of a network improvement. More details are summarized in table 2.

3.2 RM hierarchy and network virtualization

Figure 1 and 2 show that there is a natural hierarchy of regions. Several vehicle regions are dynamically connected to a RAN (Radio Access Network) region and several RAN regions are connected to what we called an edge region in figure 2. At the vehicle level, an RM controls dynamic allocation in the vehicle network, typically a Time Sensitive Network (TSN) that connects the computers (the Electronic Control Units, ECUs). For even higher communication efficiency, the internal network of a high performance ECU can be included as a region or can even further be extended to a lower level, where a network-on-chip (NoC) of a high-performance MpSoC (Multiprocessor-system-on-Chip) connects compute cores and memory components [23]. Such extensions can be useful, because database services like DDS eventually access local ECU memories.

This natural hierarchy of regions and their RMs greatly simplifies coordination, because resource management can be centralized on each level, provided such a hierarchy of regions exists, as in a vehicle. This is particularly helpful in regions with highly dynamic connections, as in the case of wireless connections. As an example, the RAN RM in figure 2 manages the RAN strategy assigning channels and timing resources to connected vehicle RMs using the two control layers. Since the nodes, i.e. the vehicles, of a RAN change dynamically, the nodes of both control layers change, as well. The management of such dynamic networks has been well explored and is part of the wireless standards, but handling the impact on end-to-end connections across the hierarchy can be challenging. It requires immediate notification of affected applications and NW clients as well as coordinated transitions under real-time constraints. That challenge is on both sides, in the vehicle, where the vehicle RM is disconnected from the RAN, and in the RAN where an existing end-to-end connection to a vehicle is failing. Here, the hierarchy helps to establish fast failure notification, as requested in the last section. All notifications follow a unique path through the hierarchy reaching the receivers of distributed applications. Similarly, coordinated transitions can exploit the hierarchy and its signaling paths. Furthermore, the hierarchical RM approach allows for fast reconfiguration, even if multiple regions are involved. RM requests are only exchanged with higher level RMs, making time-intensive communication between all affected nodes in the regions unnecessary. Hence, the hierarchy enables coordination of safety- and timing-critical applications under consideration of the respective timing constraints.

However, the region hierarchy does not reflect control hierarchy. As an example, vehicle network management must stay independent when the vehicle is connected to a RAN. The RAN is not aware of the status of a vehicle network and must not interfere with vehicle network operation, due to safety (and security) concerns. The relevance of this safety concern was explained above and will be elaborated in the use case. Consequently, resource management will always be cooperative, even in a region hierarchy.

The independence of region management has further consequences for the management of variety. The applications running on a platform as in figure 1 and 2, will have their individual structures, data and parameter sets. For instance, a service in the mobile edge might have instances with different data sets for each vehicle that uses the service. A hierarchical RM network must respect that service variety. Here, we can exploit the RM integration capabilities: An aRM can combine the joint requirements of many application services and their instances, e.g. running on a mobile edge server. This way, the RM interactions can be reduced to the connections of a network hierarchy. The resulting control traffic will be smaller and the re-configuration faster.

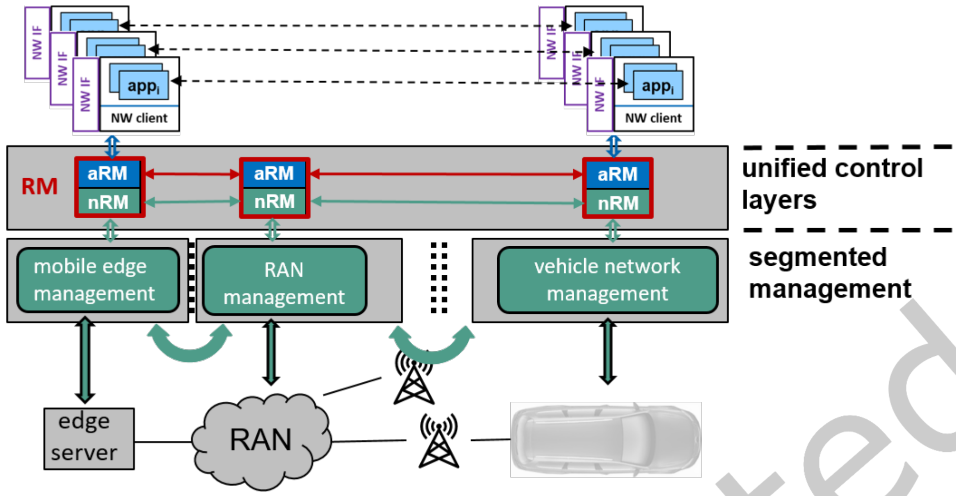

Fig. 4. Unified RM layer with distributed application connections orchestrates over different managed network segments

The resulting RM hierarchy has further advantages. An RM network that oversees a complete end-to-end connection can break down global requirements, such as end-to-end deadlines, to shorter local deadlines that are assigned to the respective RMs.

So far, we considered a single RM per physical subsystem. But RMs can also be used in the context of virtualization. Assume different providers of an application service would like to bundle their distributed applications in a provider specific virtual subsystem. The provider would be assigned its own virtual region with its own aRMs and application control layer. Likewise, network service providers can create separate nRMs managing just their own resources. In this case, the RM hierarchy can be used to integrate the virtual networks.

3.3 The two RM layers – design and ownership

When introducing the two RM layers, we did not exploit specific network or application properties. The application control layer abstracts from the network segment details and can be mapped to different communication standards. Hence, the distributed applications remain portable and can adopt new communication standards, as long as the result adheres to the end-to-end application requirements.

The separation of application and network concerns has another major advantage. End-to-end requirements of distributed applications extend to all involved network segments and must be handled in the corresponding aRMs with compatible communication principles. Therefore, aRMs can have a unified function and API that can be adapted to different application combinations including dynamic adaptation at runtime. This unification reduces software cost and simplifies maintenance. The aRMs will be configured by system owners, e.g. vehicle OEMs, and utilized by the owners of distributed applications via their application APIs. Each distributed application extends over several network segments but defines only a (small) part of the total network load of each segment. This is a motivation for unified end-to-end mechanisms that coherently manage end-to-end guarantees in distributed applications. In contrast, the network segments are typically controlled by individual providers that own and manage the network for all network load assigned to them. Even initiatives like AECC address the interaction of segment based network control, rather than try to define a single overarching and dynamically changing network and application control. Figure 4 illustrates the resulting architecture. Such a unified RM layer can further save on design cost, tooling and maintenance for the control layer itself and for distributed applications.

4 A USE CASE – AUTOMATED VALET PARKING WITH ACCESS TO INFRASTRUCTURE CAMERA

4.1 The valet parking application

In this section, we will demonstrate the management and control mechanisms using the example of an autonomous (valet) parking service to show the ramifications of application-aware RAN management. In this scenario, the application in the edge infrastructure of the parking lot provides the map and the assigned slot. The vehicle can automatically approach the assigned slot at slow speed using its own sensors. For demonstration purposes, to present the management mechanism we use a simplified scenario where the infrastructure provides one camera to stream an optional video of the scene to the vehicle. With this video input, the vehicle can detect hidden obstacles earlier and move faster. In this application, the wireless connection improves efficiency, but becomes safety-critical if it is used. There is a maximum age for each camera frame that translates into a frame deadline. If the frame does not arrive in time, the vehicle must be slowed down immediately and continue on its own sensors. A discussion on the resulting real-time problem and a solution is presented in [27]. We will adopt this solution in the paper.

4.1.1 Application requirements. In state-of-the-art application setups for valet parking [27] the camera data transmitted over the wireless channel is constrained by the limitations of state-of-the-art V2X communication technology that is used in current vehicles. As 802.11p only offers data rates up to 27 Mbit/s the videos frames, called samples, in [27] are 20 kB in size each resulting in a total of 1.6 Mbit/s when transmitting 10 samples per second. For each of those samples a deadline of 100 ms applies. Such data rates will not challenge in-vehicle networks. However, for future applications considering sharing of sensor data, as predicted in the roadmaps, the object size and therefore the required data rates will grow significantly. For high-resolution camera and lidar data, required data rates of more than 200 Mbit/s per application are expected. Hence, challenges with regard to both the wireless and the in-vehicle networks arise to be able to transmit larger samples under similar deadline requirements. Transferring such a data rate is not possible with currently deployed V2X technologies. However, upcoming standards with higher data rates for WLAN or cellular networks, as expected in the roadmaps would allow such data rates. In case of valet parking, where vehicles move slowly and in a short range, it would even be possible to reach these data rates today, using current WLAN standards, e.g. IEEE 802.11ax. In the following, we will therefore assume the valet parking application of [27] with a camera data rate of 240 Mbits/s, consisting of 10 video frames per second, 3MB each. This rate matches the data rate of the in-vehicle cameras.

As in [27], we further assume that the application data management uses the popular Distributed Data Service (DDS). DDS is widely used and was adopted in the automotive software standard, AUTOSAR.

4.1.2 Network control. This application combines two very different networks. The vehicle network always contains safety-critical traffic as long as the vehicle moves, which is all the time. We assume that the infrastructure camera sends only slightly compressed data that must be routed to the vehicle's Sensor Fusion Unit. The camera stream has a high data rate that is added to the existing vehicle sensor data and other traffic. For that purpose, the network, is reconfigured to a different mode, as explained later. Reconfiguration must be done on a switched vehicle network that carries a complex traffic profile. The profile contains application data objects of very different size and data rates, criticality and latency requirements. We assume that the network switches between modes that were predefined including mode transition to assure critical stream continuation. In contrast, the wireless channel traffic is less complex, but highly dynamic and lossy with vehicles added and removed. Yet, we want to set up robust connections with end-to-end guarantees, which require continuous connections.

This use case will show the advantages of the proposed overarching resource management.

- Each application spans over several network segments and maps its requirements to a coherent set of local requirements, even under real-time and continuity constraints.

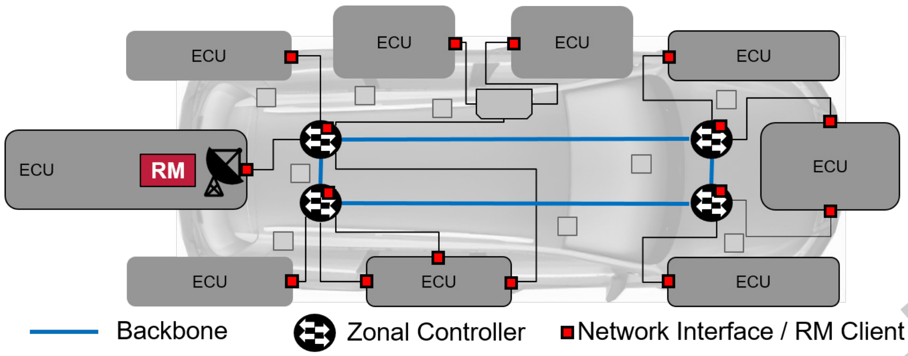

Fig. 5. Zonal architecture with Ethernet backbone including network interfaces

- Each network segment may keep its established network-centric management following completely different policies.
- Resource management can integrate multiple distributed applications with end-to-end requirements on multiple and dynamically changing network structures.

In the remainder of this chapter we will first introduce the vehicle network, the vehicle RM and it's reconfiguration management. This part will elaborate the complicated mode transition that exploits application properties orchestrated by a single central resource management. In the next step, we will first introduce the simpler RAN management, and then show the collaboration of RMs in setting up and operating an end-to-end application level protocol that efficiently supports the popular Distributed Data Service, DDS, that is used as a video frame database.

4.2 RM based dynamic resource management in the vehicle

4.2.1 A vehicle network with dynamic network control. This section describes the in-vehicle network reconfiguration process in detail. The vehicle network and its reconfiguration process is managed by the RM and is independent of other network sections.

There is an ongoing trend in automotive networks moving from heterogeneous federated systems connected by gateways of through domain architectures to zonal architectures that will dominate future vehicles (cp. e.g. [7]). Zonal architectures, as shown in figure 5, cluster all compute nodes into local zones that are connected by a switched network as backbone. Zonal controllers act as gateways to the backbone. The Time Sensitive Network is the candidate technology for this backbone. TSN is a switched Ethernet with real-time extensions (802.1Q family of standards [1]) with a data rate of 1Gbit/s in the current technology. Zonal controllers integrate all critical and non-critical traffic between the zones. Other than in current vehicle networks, a large part of the backbone traffic is safety relevant. The reason is the quickly growing traffic of high-resolution sensors that is needed for automated driving. At higher levels of driving automation, that sensor traffic will become safety-critical, which leads to high criticality of the integrating backbone [19].

That backbone criticality is the reason why up to now the automotive network is still statically configured. However, the traffic itself will not be static any longer. Traffic will depend on the vehicle mode of operation, e.g. the direction of travel, residential or city or highway traffic, manual or automated operation, or driving with degraded sensing in heavy weather. Static configurations under varying load requires network over-dimensioning or enforces quality compromise, such as lower camera image resolution or high compression.

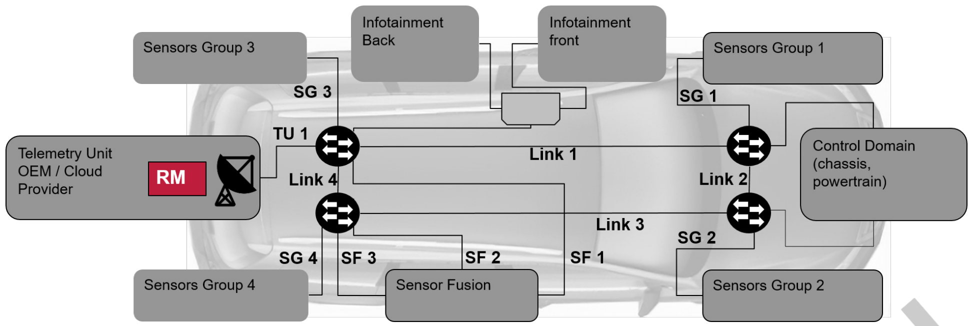

Fig. 6. Overview of the example in-vehicle network with link description

Besides sensor traffic variations, there are other reasons to assume that the future network will be dynamically controlled and re-configured, such as the need to compensate link failures, to include high data rate V2X traffic to different zones or to respond to the growing dynamics of the AUTOSAR software architectures. The core challenge of dynamic network control is the need to avoid frame losses during reconfiguration, because the critical sensor data streams are subject to real-time requirements. For a detailed investigation and a possible solution the reader is referred to [22]. In that paper, fast re-configuration is executed in a predefined sequence of centrally controlled steps under real-time constraints. Each step exploits object slack in the communication of sensor streams by holding the streams temporarily back at the network interfaces (NW IF) of the ECUs, such that no packets are lost. The configuration steps use IEEE 802.1Qcc for the NIC parameters, such as access control and shaping, while the zonal controllers are based on software-defined switches (cp. [24]), with the novelty that switch programming is executed with a bounded time protocol [30], [22].

With that approach, each re-configuration step takes less than 1 ms in OMNET++ simulations under ongoing sensor traffic load, as applied from examples systems for automated driving [14], [11]. The simulation results were confirmed with a physical prototype for the vehicle network [21]. For details on the protocol, the simulation and the prototype setup see [22].

For the use case, we assume a configuration as shown in figure 6. It consists of four sensor groups (e.g., lidar, radar, video cameras), a gateway to the control domain (e.g., powertrain, chassis with CAN, CAN-FD, FlexRAY buses), a Sensor Fusion Unit (SF) for autonomous decision-making, and a Telemetry Unit that provides connectivity to the cloud and edge infrastructure and other V2X radio communications. Sensor streams from the sensor groups are routed to the sensor fusion unit. We assume high-resolution sensors as predicted in various publicly available benchmark sets. Some units, such as the Control Domain, have redundant connections for fault tolerance.

4.2.2 Vehicle network operation in the use case. The vehicle aRMs keep a list of all communication between the SW components of distributed applications. Here, we focus on sensor streams that are characterized by data object size, period and object deadline. The communication requirements of all applications of an ECU are integrated and translated to a network load description that is forwarded to the nRM. The network load description can, e.g., consist of an average data rate and a deadline representation, such as the demand bound function (dbf) [9].

As a reference scenario, we assume link and terminal traffic as summarized in table 3. For simplicity, we do not use time aware shaping (802.11Qbv). All streams are assigned to the priority class B. In the valet parking application, the external camera stream is routed to the sensor fusion unit. The current network configuration would be sufficient to provide the resources for a data stream of a few Mbit/s, as in [27], but requires reconfiguration for the 240Mb/s stream that we assume in the paper.

Table 3. Table with data stream description and transmission details which are important for the RM e.g camera stream deadline and sample rate

Data Type	Deadline	Object size	Sample Rate	Number of Fragments per Object	Bandwidth
Camera	10 Hz → 100ms	2 MP ~3MB per sample	10 Hz	2000 Eth Frames	240 Mbit/s
Lidar	5 Hz → 200ms	up to ~300,000 points/second ~3MB	5 Hz	2000 Eth Frames	120 Mbit/s
Radar		Continuous Stream			64 Mbit/s

Table 4. Detailed data stream path description influenced by the mode change (new link routing is marked, old route is strikeout)

Module	Target	Route	Bandwidth Mbit/s
SG1	SF	SG 1 - Link 1 - ~~Link 4~~ - SF 3 - **SF 1**	Camera 240 Lidar 120 Per Sensor Group
SG2	SF	SG 2 - Link 3 - SF 2	
SG3	SF	SG 3 - ~~Link 4~~ - ~~SF3~~ - **SF 1**	
SG4	SF	SG4 - ~~Link 4~~ - ~~SF 1~~ - **SF 2**	
TU	Control ECU	**TU 1 - Link 4 - SF 3**	Camera 240

To provide the additional bandwidth, we assume that the nRM decides to reconfigure the vehicle traffic as depicted in table 4. The reserved bandwidth is above the external video data rate, which is necessary to reach the video frame deadline. To keep the use case simple, the reserved bandwidth is set to 800Mbit/s corresponding to a link utilization of 80%.

If the network resources are insufficient or the demand bound function cannot be met (requires formal analysis), the request is rejected (A3.N). The re-configuration occurs using the stepwise safety protocol of [22] that consists of a configuration protocol between vehicle RM and the ECU network interfaces (NW IF, figure 5). Upon completion, the vehicle network provides the required data rate for the infrastructure camera, which is signaled to the aRM (A3.G).

4.3 Application awareness in RAN management

4.3.1 The vehicle as a RAN node. As already noted in the previous section and illustrated in figure 3, there is a communication hierarchy, but no control hierarchy between different network segments e.g. the vehicle and RAN RMs. The vehicle can reject requests from the RAN if the current mode of operation does not permit the requested traffic if the requested resources would jeopardy the safety of the current network mode, e.g., because of high vehicle network load or a vehicle network failure. Similarly, the RAN can reject requests from the vehicle, for example if the wireless air interface is not able to transmit another video stream without the error rate increasing to non-working areas.

From the RAN perspective, the vehicle RM is a single entry point that provides bundled information on the profiles of all applications that are using or want to use the V2X channel. For network management, it is sufficient to know the profile and QoS requirements. In the other direction, the vehicle RM receives all wireless access information to rapidly adapt the vehicle network and applications.

With this approach, the vehicle can get access to one or more applications, e.g., infrastructure cameras, under the control of RAN RM, allowing a flexible setup and tear-down of connections under end-to-end guarantees. The authorization of new connections depends on parameters such as network load, channel quality and number of vehicles. The larger the network grows and the more applications are involved, the more network control and optimization options arise from such application-aware resource management, even under end-to-end requirements.

4.3.2 RAN network and protocol in valet parking. As stated, the parking lot provides wireless access to its infrastructure camera at a data rate of 240Mbit/s using a current WLAN standard, e.g. IEEE 802.11ax. The infrastructure camera is directly attached to an edge server of the parking lot that controls the local RAN. The camera stream is routed from the edge computer through the V2X wireless link to the Telemetry Unit (TU) and, from there, to the SF in each connected vehicle where it is stored by the DDS middleware, cp. figure 5 and figure 6. In a larger parking lot, there will likely be more than one infrastructure camera, but the configuration is sufficient to explain the use case.

We assume a simple application protocol where vehicles are registered sequentially upon arrival, one vehicle at a time, and deregistered when leaving. We assume an external camera control application (ECCA) which is located in the vehicle's Telemetry Unit and discovers the camera service in the edge computer as part of registration. The ECCA only controls the application while the database is located in the Sensor Fusion Unit.

The RAN connection is not as stable and reliable as the wired in-vehicle connections due to the lossy and unstable medium. During data transmissions, interference and collisions with other transmitters occur, resulting in longer arbitration time and lower channel throughput. Because the application expects error free data objects, wireless communication will use an error protocol. DDS provides an efficient end-to-end error protection protocol at the level of application data objects that requires unicast communication. By using UDP/IP and turning off the MAC layer error protocol and adapting two simple MAC protocol parameters to the application frame pattern and the current number of moving vehicles, the connection becomes far more robust, as shown in [27]. These application specific parameters can be communicated from the RAN RM to the vehicle RM upon entering the parking lot and can stay constant for a well predictable protocol timing.

The protocol timing is sensitive to parameter setting, number of vehicles and error rates. Error correction is prioritized meaning that a noisy vehicle connection with standard MAC-layer error correction leads to higher delays for all other connections eventually forcing those vehicles in the degraded mode. However, if error correction knows when the object deadline is exceeded it can stop sending error messages to not disturb other vehicle operation.

Any error protocol with repeated transmissions adds uncertainty to the wireless network load per vehicle. That uncertainty will grow with packet loss rates, which further increases communication interference. One way to control interference in the wireless channel is a dynamic arbitration policy that uses few context dependent parameters for traffic shaping [27]. The context is defined by the size and rate of data objects, by the number of vehicles and by the quality of the vehicle communication channels.

4.3.3 Protocol options. The application combines two networks with different operation and protocols, but with a common end-to-end timing requirement for application data objects. There are several implementation options.

- Option 1: The network system is application agnostic and forwards packets with the standard MAC layer error correction protocol applied in both network segments.
- Option 2: Development of a new protocol that manages both network segments together, thereby considering the effect dynamics and interference of vehicle network and RAN.
- Option 3: The two segments are operated independently. In this option, the ECCA application in the Telemetry Unit serves as a proxy. It receives the incoming wireless packets and uses the DDS error protocol to reconstruct error-free application objects before they are forwarded over the loss-free network to the Sensor Fusion Unit.
- Option 4: The packages received on the wireless channel are repackaged and directly forwarded to the Sensor Fusion Unit including the whole bidirectional wireless network error protocol. As we will see in the evaluation section, the protocol is more robust for the given deadlines, but increases communication requirements in the vehicle.

Option 1 is the network-centric state of the art. The investigation in [27] shows that option 1 cannot handle higher frame loss rates due to protocol limitations, even if only considering the wireless network segment. The reason is that the MAC layer protection is limited in the number of repetitions and leads to queueing delays when combined with DDS error protection. The result are video frame deadline misses even at lower error rates, which will be aggravated when adding the vehicle network delay. Option 2 requires a new network protocol for this specific application that leads to a complex application and network integration problem. The following options avoid such a costly development.

The two remaining options require application knowledge to select and manage an efficient solution. Because every vehicle will have a different vehicle network configuration, the in-vehicle delay will differ and, hence, the available time for error correction. In essence, end-to-end communication robustness will depend on the selected option and the individual vehicle configuration.

On the RAN side, the protocol for option 3 and 4 is the same. Therefore, the RAN can serve vehicles with option 3 and 4 at the same time. On the vehicle side, the application will first require option 4 for higher robustness and, if the higher data rate of option 4 is not available, request resources for option 3. If the data rate for option 3 is also not available or if the request is rejected by the RAN, the external camera service is not activated for this vehicle.

4.3.4 RAN and vehicle cooperation. Now, assume a vehicle enters the parking lot and establishes a wireless connection to the RAN manager and its RM followed by setting up the control protocols at both control layers of the RAN RM and the vehicle RM (A1.S, N1.C). We suggest the following protocol:

(1) The first step is the discovery protocol. In this example, we assume that the vehicle is equipped with a sensor fusion module that can utilize external cameras. In this case, service discovery would be initiated by the vehicle. For simplicity, we assume that the external camera control application (ECCA) is located in the telemetry unit and that the infrastructure camera format matches the format needed in the sensor unit. Otherwise, there must be an agreement protocol between infrastructure camera service and telemetry unit following established patterns.
(2) Next, the ECCA uses the aRM API (A2,R) to request the communication pattern needed for the camera stream, including data object size, object arrival curve, deadlines, criticality, etc. The aRM will use its coordination interface to send a request to the RAN aRM to provide access to the camera data (A1.R). The RAN aRM collects all these requests for RAN connections, but holds the grant until the additional connection has been approved by the network segment. At the end of this step, the aRMs reflect the application requirements broken down to local requests.
(3) In the third step, the application requests are translated to network requirements using the aRM_nRM interaction (A3.R). The vehicle nRM will not yet grant the request, because vehicle network reconfiguration is needed. The RAN nRM will check its set of wireless connections to determine if the new request can be served or if the wireless network will lose the contracted real-time capability when accepting another connection. If one of the two nRMs cannot serve the request, it would reject the request (A3.N). In this case, the aRM would send a no-grant message to the connected aRM (RAN) or cancel its request (vehicle).
(4) The fourth step is a coordination of the nRMs mainly for network protocol adaptation, where needed. Here, we are using the work of [27] to adjust traffic shaping in the telemetry unit.
(5) Every nRM that was successful in re-configuration grants its aRM request (A3.R). Every aRM that received a positive A3.R sends a grant to the requesting aRM (A1.G). In the use case, the RAN aRM grants service to the vehicle aRM. The aRMs that received all grants from their nRM and all aRMs with service requests notify their requesting application (A2.G). Then, the distributed camera streaming application can synchronize its start.

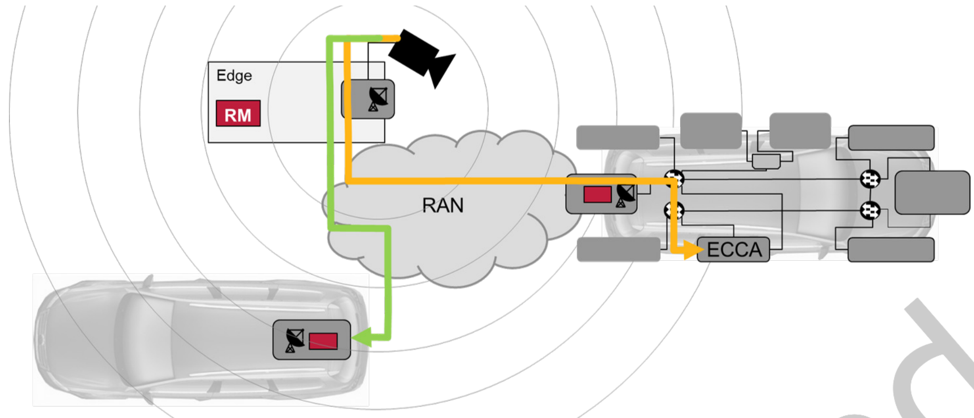

Fig. 7. Overview of the use case with two vehicles - each with its own RM - connected over the RAN with a camera as data source. Data from camera application is send over the RAN to the in-vehicle network

This example protocol shall serve to illustrate the concept. More protocol sequences will be needed in practice for checking vehicles out, or for protocol error handling etc.. The example protocol has three parts, the application level orchestration in steps 1-2, the network level re-configuration in steps 3-4, and the application activation in step 5. The first part determines the new configuration in an application-centric process that provides consolidated local input to a subsequent network-centric management of network segments. We did not use protocol or application properties to keep generality. Step 5 always assumes that a network failure leads to a fall back to the previous network status.

The infrastructure camera service to a vehicle can end in different ways, by canceling the registration because the vehicle has reached the slot or left the parking lot, or by discontinuation, e.g. because of a bad wireless connection or the need to cancel the reservation in the vehicle network, such as in case of a network failure [22]. Canceling the registration is an application action that will be passed to the RM using its API on the vehicle or RAN side. Discontinuation can be a network action that is passed to the application by the RM, or is detected by the application. The latter case is important for the valet parking safety case where this is a requirement. Here, the application has several possibilities, e.g. by time-out scenarios, such as the lack of heartbeat messages received on the receiver-side or the lack of acknowledgement bitmaps that should be sent back to the sender in response to heartbeats. This way, there is already a detection mechanism on all relevant nodes – in this example in the vehicle and in the edge, a requirement at least on the vehicle side. The detected network failure can then be used to cancel the connection by interaction of the respective aRMs and nRMs, just as in the case of an application initiated cancellation above.

4.4 Evaluation

The last part of this chapter will evaluate the efficiency of the application-aware options 3 and 4 in the valet parking use case. The comparison is based on an OMNeT++ simulation.

The options differ in the way in which the two segments are coordinated.

We assume that both the edge infrastructure and the vehicle are equipped with 802.11ax transmitters and receivers, supporting data rates of roughly 1200 Mbit/s for a single stream. At that data rate, the 3 MB video frame would be transmitted in just 20ms, leaving time for retransmission of lost frames using the DDS based error correction protocol of [27], even under higher error rates. The protocol exploits fragment-based transmission of

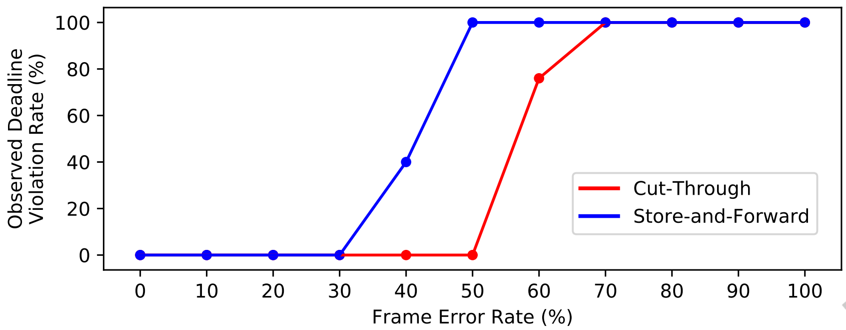

Fig. 8. Deadline violation rates comparing the cut-through and store-and-forward approaches in an optimal setup with a shaping period of 110us

large objects. The fragment size is set to 11.370 bytes, leading to a packet size of 11.454 bytes – the maximum MPDU size in 802.11ax.

Option 3 treats both segments as independent. The store-and-forward protocol of this option transfers a video frame from the Telemetry Unit to the Sensor Fusion Unit once it is complete and error free. This way, the vehicle-internal communication is not affected by the error protocol of the wireless link that includes out-of-order fragment transmission. Because the vehicle network is considered loss-free, the overhead for error correction can be neglected. Therefore, assuming a link budget of 800 Mbit/s after reconfiguration and a propagation time of 1ms (Class B), 31 ms are needed for transmitting the 3 MB sample over the vehicle network. As a result, the time budget for wireless video frame transmission is reduced to 100ms-31ms = 69ms.

On the network level, option 4 is a straightforward cut-through protocol, except that the different frame formats require repackaging in the Telemetry Unit. The error protection is handled end-to-end by the adapted DDS application protocol that considers the vehicle network as transparent loss-free medium. In this option, network segment adaptation occurs via network parameter settings, which are derived from the requirements of the valet application and adjusted by the two coordinating aRMs. However, it is important to note that adjusting the network and application parameters requires knowledge on all the segments. First, application information including sample size and deadlines as well as properties of the used error protection are considered. Second, we use properties such as available throughput, latencies and channel utilization attributed to the wireless channel. Third, in-vehicle network properties like link utilization, routing, available bandwidth and packet latency are taken into account to derive feasible parameters.

Different from option 3, option 4 adds delay to the error protocol, because all protocol messages are delayed by the vehicle network.

We compare the two approaches under different bit error rates on the wireless channel. We adopt the error model from [27] and calculate frame-error-rates accordingly for the packet size of 11.454 bytes. In a first experiment we assume an unoccupied channel, allowing for optimal shaping parameters. For packets the size of 11.454 bytes and a data rate of 1200 Mbit/s the packet transmission time equals roughly 72.8us. We set the shaping period to 110us to accommodate the transmission itself and the DCF Interframe Space (DIFS) of 34us. The DIFS is the time a node must sense an idle channel before transmission is permitted. This is the arbitration overhead in empty channels for 802.11.

The experimental results are illustrated in figure 8. For interpretation, it should be considered that a single deadline violation already violates the specification for a safe use of the infrastructure camera. The results highlight the advantages of the tight coordination between the two network segments. Applying a cut-through

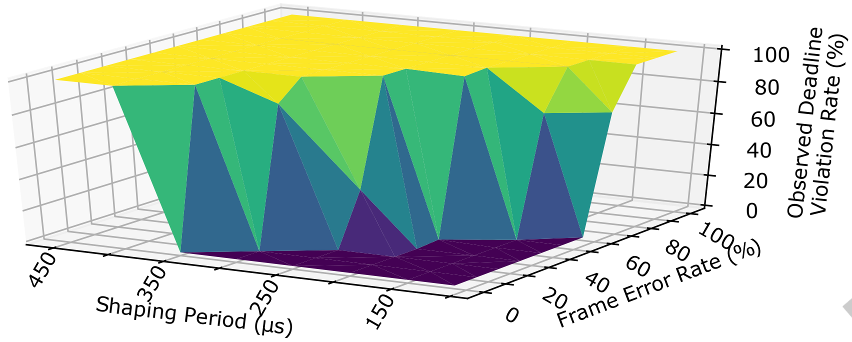

Fig. 9. Deadline violation rates in different configurations as a function of the shaping period. Tightly coupled network and application coordination using a cut-through approach (option 4)

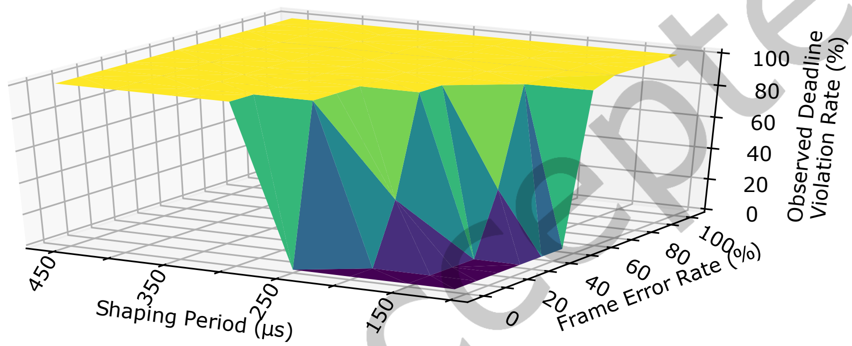

Fig. 10. Deadline violation rates in different configurations as a function of the shaping period. Coordination using the store-and-forward approach (option 3)

approach by coordinating the RAN and in-vehicle shaping tolerates 50% frame error rate in our tests. Only at frame error rates higher than 50% deadline violations can be observed. In contrast, using the store-and-forward approach first deadline violations already occur above frame error rates of 30%. This result can be explained by the significant deadline reduction for the wireless transmission in option 3. The additional error traffic in the vehicle in option 4 and the delay introduced by the vehicle network, however, does not impair the advantage of the cut-through protocol.

So far, we have assumed a single vehicle per wireless channel. If more vehicles use the camera service concurrently on the same channel, load and average arbitration time will grow. The shaping period is a key parameter for controlling the wireless channel interference, such that several vehicles can be served at the cost of a diminished channel robustness [27].

Figures 9 and 10 visualize the impact of varying shaping periods on options 3 and 4. As expected, an increase in shaping period leads to worse performance at given frame error rates, because less fragments can be transmitted in the same time interval. Therefore, less slack is available for retransmissions. The robustness of the cut-through approach of option 4 dominates the robustness of option 3 for all evaluated frame error rates. Even for an error free transmission, option 3 already fails for shaping periods above 200 us (figure 10) whereas option 4 still transmits successfully at a 20% frame error rate using the same shaping (figure 9). Even at shaping periods of 300us option 4 showed no deadline violations up to frame error rates of 10%. As a result, it would be possible to support three vehicles with the same configuration simultaneously, on the same channel. This would not be possible using the store-and-forward approach of option 3. The figures show that network segments tightly coordinated by an application-aware resource management offer a far greater parameter space for finding feasible configurations in degraded channel conditions (under increasing frame error rate) as well as when more applications or vehicles require access to the same channel. For the valet parking application, the results have several consequences. Option 4 should be selected whenever possible on the vehicle network. The shaping period is the key parameter to control interference for several vehicles. It should be implemented as an application decision prioritizing high robustness or a larger number of concurrently served vehicles. Translating the application request to a shaping value, however, is the task of the RAN RM in order to achieve portability to different wireless standards.

4.5 Case study conclusion

The case study in this chapter is quite extensive. This is due to the difficult matter. The complex challenge of integrating the established domains of vehicular technology and wireless networks with many different requirements, aspects and dependencies cannot be sufficiently addressed by focusing on only part of the overall network. The modularity of the proposed resource management enables a systematic approach that keeps application integration and network management as separate tasks. As an example, we could replace the single camera stream by a set of lower data rate streams from several infrastructure cameras without changing the network parameters, as long as the aggregation does not change the wireless network requirements. The vehicle network management can dynamically reconfigure the network, as long as the aggregated streams continue to be routed to the Sensor Fusion Unit, thereby meeting the requirements of data rate and maximum data object latency without losses. This works under the condition that the wireless channel is sufficiently reliable. E.g., in scenarios with very high vehicle speeds accompanying physical signal propagation effects such as Doppler shift, shadowing or reflections gain more relevance and decrease the signal to noise ratio. As a result, such scenarios are more likely to not allow for any critical communication in a wireless channel. However, this is an issue related to the wireless channel and not the proposed RM protocol. If a sufficiently reliable channel exists, the established network-centric approach is still necessary, because the application profile alone is insufficient to determine an efficient end-to-end communication. This becomes obvious when comparing options 3 and 4. However, network configuration and parametrization can be localized, as demonstrated in the use case.

The use case considers a single distributed application only. The vehicle network is shared by multiple applications with resource assignment and reconfiguration controlled by the vehicle RM [22]. The concept of RM is not limited to a single application. For example, the RAN RM can keep a similar directory for multiple applications sharing the same wireless channel. Like in the vehicle RM, where every network node has a different application profile, the RAN RM can keep a different application profile per connected vehicle. If a vehicle prefers to keep all applications private, the approach would still work with an aggregated profile, at a reduced optimization potential for that particular vehicle.

5 OUTLOOK

5.1 Business perspective: incentives and challenges

Like all infrastructure technologies, application-aware resource management will only be successful if there is a path to practical innovation. Because of the huge number and diversity of involved players, a single disruptive step is unlikely to be successful. However, the valet parking use case shows that the introduction can be broken down into incremental improvements. A vehicle with dynamic network configuration will use its network more efficiently, with performance, cost, and energy benefits. The added readiness to support high performance application services is just an additional advantage that only increases the incentive to proceed from static to dynamic vehicle network configuration. Once implemented, this readiness might turn into a competitive advantage for the vehicle and, given the long vehicle lifetime, for the customer who can participate in future high-end services. Improving the use of an infrastructure camera with by an application-level error protocol is a first step towards an application-aware network management that is already useful in current wireless networks [27]. A RAN and a mobile edge computer that provide an RM infrastructure can already be used in confined environments, such as for automated valet parking, where it can be used stand-alone to optimize the use of the infrastructure cameras service with application-aware selection of shaping parameters and access control. That is a cost and efficiency benefit for a valet parking service thereby respecting safety requirements. Cooperation with vehicle RMs for an advanced camera service with high data rates, as in this paper, is a second step of improvement. It is even possible to cooperate with vehicle RMs, if only part of the vehicles are equipped with a vehicle RM, such that legacy vehicles can still be served at a lower service level.

Beyond edge computing, achieving application guarantees will be more difficult, while application, user space, and dynamics grow. At that level, the introduction of an application-aware resource management is more likely to follow, once it is used in the edge, or it might be limited to few applications with safety and real-time requirements, such as vehicle tele-operation.

5.2 Research Topics

The application-centric approach presented in this paper shows that the change of perspective from communication to distributed computation has a large potential for new solutions and related research. The separation of application control layer and network control layer is a novel concept that deserves more attention. The application control layer captures the communication profile of all distributed applications of a compute node rather than of a single application, as usual in distributed computing. It represents the integrated network load resulting from the mapping of all application components to compute nodes thereby keeping the structure of application data objects where needed for network management. In traditional real-time and safety-critical systems, this integration is performed at design time. Because applications, mappings and network change dynamically, integration under real-time and safety constraints turns into a run-time task. The application control layer and its protocol elaborated in this paper are only an example how this integration task could be mastered at run-time. There is much room for other ideas and improvements.

A second research direction is the interaction between integrated application requirements and network management. Already in our use case, lossy communication changes a deterministic application load into a non-deterministic network load with impact beyond a single network segment, as seen in option 4. A closer look reveals that this non-determinism has its origin in the DDS application protocol that is shaped by the lossy wireless network, corresponding to a feedback from network to application. Modeling, analyzing and optimization under such feedback, as in option 4, is an open issue.

A third research direction is the design and verification of dynamic network management. A very helpful side effect of the localized application-network interaction is the opportunity to keep network management and reconfiguration local, with local protocols. The valet parking use case was a good example where we could use the protocol of [22] for local reconfiguration of the mixed criticality vehicle network. However, the reconfiguration of RAN resources and the critical inter-vehicle communication over a unreliable channel requires adaption of the in-vehicle resource management protocol. The modularity supported by the two control layers could give an incentive to approach the larger challenge of proving distributed application properties across multiple network segments, such as needed for vehicle tele-operation, by a systematic combination of local guarantees. As the results in section 4.4 made apparent, simplistic approaches of breaking down requirements, i.e, sample deadlines, between network segments is not efficient in complex systems as considered here. Hence, novel mechanisms for systematically breaking down requirements based on the capabilities and properties of each involved network segment are needed to ensure a reasonable degree of composability. Furthermore, with the

RM protocol being subject to safety constraints, there are timing implications as well. While there have been evaluations of RM protocols with respect to reconfiguration time for in-vehicle networks [22], due to the lack of actual implementations it remains uncertain whether a hierarchical RM protocol, that also utilizes a wireless channel, can achieve similar reconfiguration times. Therefore, protocol implementations need to be developed and tested in the future.

There are many more research opportunities in vehicular computer and communication technology, such as redundant architectures for increased dependability or functional safety concepts, to name just a few further topics. And, last but not the least, research into applications that exploit the application-centric approach, as in the valet parking example, can pave the way for many of the applications that are anticipated in the popular V2X surveys and roadmaps. This includes systems hosting multiple applications, either of the same type (multiple cameras) or of different type (mixed criticality), using the same wireless channel and requiring coordination by a RM. Also, more dynamic scenarios, e.g. truck platooning on a highway, are reasonable. Despite the protocol being designed with scaling and generality in mind, a detailed evaluation of such systems, applications and scenarios remains an open issue.

6 CONCLUSIONS

The paper complemented the network-centric perspective of 5G and 6G roadmaps and survey papers with an application-centric perspective that approaches wireless networks as a platform for a large number of co-existing distributed applications. Important distributed applications come with essential real-time and dependability constraints that must be mapped to network requirements and guarantees. These distributed applications can have different topologies involving different network segments. Integrating many such applications under network and application dynamics is an even greater challenge to both application and network management. The paper addresses the challenges by a network of resource managers that are each responsible for a network domain and cooperate via two inter-operating control layers, one for integration and adjustment of distributed application requirements and one for providing input to network management. The application control layer reflects application topology and dependencies while the network control layer reflects the network topology. The network topology and the resulting hierarchy can be exploited for more efficient communication between the resource managers. The paper selected automotive applications with V2X communication as an important target of future wireless networks. A valet parking use case with access to an infrastructure camera service with high data rates using a DDS based protocol was elaborated. By combining two very different local networks for the vehicle and for the radio access network, a robust end-to-end communication with high data rate and low latency could be established and evaluated for different communication strategies as long as the channel quality allows transmissions. The approach targeted automotive applications but is general enough to be applied to other distributed applications with similar requirements, such as industrial networks.

ACKNOWLEDGMENTS

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant ER168/32-2.

ACRONYMS

REFERENCES

[1] 2018. IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks. *IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014)* (2018), 1–1993. https://doi.org/10.1109/IEEESTD.2018.8403927

[2] 3GPP. 2018. *3GPP TR 22.886 - 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on enhancement of 3GPP Support for 5G V2X Services (Release 16).*

3GPP	3rd Generation Partnership Project	**NIC**	network interface controller
5GAA	5G Automotive Association	**nRM**	network level RM
AECC	Automotive Edge Computing Consortium	**NoC**	network-on-Chip
aRM	application layer RM	**NW**	network
API	Application Interface	**NW IF**	network interface
AUTOSAR	Automotive Open System Architecture	**OEM**	original equipment manufacturer
CAN	Controller Area Network	**QoS**	Quality of Service
dbf	demand bound function	**RAN**	Radio Access Network
DCF	Distributed Coordination Function	**RTPS**	Real-Time Publish Subscribe protocol
DDS	Data Distribution Service	**RM**	Resource Manager
DIFS	DCF Interframe Space	**SAE**	Society of Automotive Engineers
ECU	Electronic Control Unit	**SDN**	Software Defined Network
ECCA	external camera control application	**SF**	Sensor Fusion Unit
FS	functional safety	**SG**	Sensor Group
GSM	Global System for Mobile Communications	**SOTIF**	safety of the intended function
GSMA	GSM Association	**TARA**	Threat Analysis and Risk Assessment
IF ID	interface ID	**TSN**	Time Sensitive Network
IP	Internet Protocol	**TU**	Telemetry Unit
ISO	International Organization for Standardization	**UDP**	User Datagram Protocol
LAN	Local Area Network	**VDA**	German Association of the Automotive Industry
MAC	Media Access Control	**V2X**	Vehicle-to-everything
MPDU	MAC Protocol Data Unit	**WAN**	Wide Area Network
MpSoC	Multiprocessor System-on-Chip	**WLAN**	Wireless LAN

[3] 3GPP. 2021. *3GPP TR 21.916 - 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Release 16 Description; Summary of Rel-16 Work Items (Release 16).*

[4] 5G Automotive Association (5GAA). 2020. A visionary roadmap for advanced driving use cases, connectivity technologies, and radio spectrum needs.

[5] AECC Consortium. 2020. Driving Data to the Edge: The Challenge of Traffic Distribution. https://aecc.org/resources/publications/ accessed: 21.9.2021.

[6] AECC Consortium. 2021. Connected Cars: On The Edge Of A Breakthrough. https://aecc.org/resources/publications/ accessed: 20.12.2021.

[7] Onur Alparslan, Shin'ichi Arakawa, and Masayuki Murata. 2021. Next Generation Intra-Vehicle Backbone Network Architectures. In *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*. 1–7. https://doi.org/10.1109/HPSR52026.2021.9481803

[8] Amazon. 2018. Connected Vehicles and the Cloud Enabling the Future of the Mobility User Experience. https://d1.awsstatic.com/Industries/Automotive/V2/Navigant%20Research%20-%20Amazon%20White%20Paper%20Full%20-%204-24-2018.pdf accessed: 21.9.2021.

[9] Sanjoy K. Baruah, Louis E. Rosier, and R. R. Howell. 1990. Algorithms and Complexity Concerning the Preemptive Scheduling of Periodic, Real-Time Tasks on One Processor. *Real-Time Syst.* 2, 4 (oct 1990), 301–324. https://doi.org/10.1007/BF01995675

[10] Mostafa Zaman Chowdhury, Md. Shahjalal, Shakil Ahmed, and Yeong Min Jang. 2020. 6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions. *IEEE Open Journal of the Communications Society* 1 (2020), 957–975. https://doi.org/10.1109/OJCOMS.2020.3010270

[11] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. 2020. A2D2: Audi Autonomous Driving Dataset. *CoRR* abs/2004.06320 (2020). arXiv:2004.06320 https://arxiv.org/abs/2004.06320

[12] Marco Giordani, Michele Polese, Marco Mezzavilla, Sundeep Rangan, and Michele Zorzi. 2020. Toward 6G Networks: Use Cases and Technologies. *IEEE Communications Magazine* 58, 3 (2020), 55–61. https://doi.org/10.1109/MCOM.001.1900411

[13] GSMA. 2017. Cellular Vehicle-To-Everything. https://www.gsma.com/iot/wp-content/uploads/2017/12/C-2VX-Enabling-Intelligent-Transport_2.pdf accessed: 21.9.2021.

[14] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. 2020. The ApolloScape Open Dataset for Autonomous Driving and Its Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 10 (2020), 2702–2719. https://doi.org/10.1109/TPAMI.2019.2926463

[15] Huawei. 2017. Connected Car on The Road to 5G. https://www.huawei.com/en/technology-insights/industry-insights/outlook/mobile-broadband/xlabs/insights-whitepapers/huawei-whitepaper-connected-car-on-the-road-to-5g accessed: 21.9.2021.

[16] IEEE Standards Association. 2018. *IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks – Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements*. Vol. 2018. 1–208 pages. https://doi.org/10.1109/IEEESTD.2018.8514112

[17] ISO21434. 2021. Road vehicles — Cybersecurity engineering.

[18] ISO21448. 2019. Road vehicles — Safety of the intended functionality.

[19] ISO26262. 2011. Road vehicles — Functional safety.

[20] SAE J3216. 2020. Taxonomy and Definitions for Terms Related to Cooperative Driving Automation for On-Road Motor Vehicles J3216 202107.

[21] Adam Kostrzewa and Rolf Ernst. 2020. Fast Failover in Ethernet-Based Automotive Networks. In *2020 IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC)*. 134–139. https://doi.org/10.1109/ISORC49007.2020.00027

[22] Adam Kostrzewa and Rolf Ernst. 2021. Achieving safety and performance with reconfiguration protocol for ethernet TSN in automotive systems. *Journal of Systems Architecture* 118 (2021), 102208. https://doi.org/10.1016/j.sysarc.2021.102208

[23] Adam Kostrzewa, Selma Saidi, and Rolf Ernst. 2015. Dynamic Control for Mixed-Critical Networks-on-Chip. In *2015 IEEE Real-Time Systems Symposium*. 317–326. https://doi.org/10.1109/RTSS.2015.37

[24] Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Veríssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. 2015. Software-Defined Networking: A Comprehensive Survey. *Proc. IEEE* 103, 1 (2015), 14–76. https://doi.org/10.1109/JPROC.2014.2371999

[25] Microsoft. 2017. Empowering automotive innovation. https://download.microsoft.com/download/6/9/D/69D92EB1-F1EE-4893-ABE1-C005D7F9FF57/Microsoft_Connected_Vehicle_Platform_Whitepaper_EN_US.pdf accessed: 21.9.2021.

[26] Object Management Group (OMG). 2015. Data Distribution Service, Version 1.4. OMG Document Number formal/2015-04-10.

[27] Jonas Peeck, Mischa Möstl, Tasuku Ishigooka, and Rolf Ernst. 2021. A middleware protocol for time-critical wireless communication of large data samples. In *2021 IEEE Real-Time Systems Symposium (RTSS)*. 1–13. https://doi.org/10.1109/RTSS52674.2021.00013

[28] Walid Saad, Mehdi Bennis, and Mingzhe Chen. 2020. A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems. *IEEE Network* 34, 3 (2020), 134–142. https://doi.org/10.1109/MNET.001.1900287

[29] Harsh Tataria, Mansoor Shafi, Andreas F. Molisch, Mischa Dohler, Henrik Sjöland, and Fredrik Tufvesson. 2021. 6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities. *Proc. IEEE* 109, 7 (2021), 1166–1199. https://doi.org/10.1109/JPROC.2021.3061701

[30] Daniel Thiele and Rolf Ernst. 2016. Formal analysis based evaluation of software defined networking for time-sensitive Ethernet. In *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*. 31–36.

[31] Verband der Automobilindustrie (VDA). 2019. Standardization Roadmap for Automated Driving. https://en.vda.de/en/services/Publications/standardization-roadmap-for-automatic-driving.html accessed: 21.9.2021.

[32] Zhengquan Zhang, Yue Xiao, Zheng Ma, Ming Xiao, Zhiguo Ding, Xianfu Lei, George K. Karagiannidis, and Pingzhi Fan. 2019. 6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies. *IEEE Vehicular Technology Magazine* 14, 3 (2019), 28–41. https://doi.org/10.1109/MVT.2019.2921208