IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 23, 2022, accepted April 6, 2022, date of publication April 18, 2022, date of current version April 27, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3168354

A Survey of Vehicular Network Systems for Road Traffic Management

KAROL JURCZENIA AND JACEK RAK, (Senior Member, IEEE)
Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland
Corresponding author: Karol Jurczenia (karol.jurczenia@pg.edu.pl)

ABSTRACT In this survey, we analyze the proposals of vehicular communication systems in the context of road traffic management. Starting with the definition of communications between vehicles (V2V), vehicles-to-infrastructure (V2I) and vehicles-to-everything (V2X), we first focus on the requirements and current standards for the Intelligent Transport Systems (ITS), including the maximum communication delay, the communication range and the size of messages (in the case of V2I transmission). After that, we analyze the use cases in line with the implementation of intelligent traffic management and review the respective methods that support or directly manage traffic on roads. One of the primary objectives of this paper is to highlight the architectures of four classes of systems able to support vehicular traffic management and communication between vehicles and roadside infrastructure, namely: vehicular cloud computing (VCC), cloudlets, Mobile Edge Computing (MEC) and fog computing. In this context, we also present our classification of the methods for these four classes of architectures. In the end, we provide our opinion on problems and limitations concerning the deployment of mechanisms belonging to each considered architecture class.

INDEX TERMS Vehicular networks, VANETs, intelligent vehicles, vehicular cloud computing, cloudlets, cloud infrastructure, mobile edge computing, fog computing, road traffic management, intelligent transportation systems, ITS, machine-to-machine communications.

I. INTRODUCTION

The scope of Vehicular Ad-hoc NETworks (VANETs) is to facilitate information exchange between vehicles – known as vehicle-to-vehicle (V2V) communications, as well as to provide vehicle-to-infrastructure (V2I) or even vehicle-to-everything (V2X) communication possibilities mainly to improve safety on the road, and enhance road traffic management [1]. Such networks are formed by vehicles equipped with On-Board Units (OBUs) and the components of the roadside infrastructure (referred to as Roadside Units – RSUs). They can be the basis for applications like route planning for drivers or suggesting the speed of vehicles to minimize their fuel consumption. Most vehicular applications and services require low-latency communications as the conditions on the road may change quickly, and there is little time for reaction. Implementing traffic management services by machines is a non-trivial problem as there is little time to exchange the data, process the information, and react to road events. Therefore, there is a need to design vehicular networks able to operate in real-time.

Road safety can be improved, e.g., by disseminating information about collisions in a specific area or informing the drivers about the conditions on roads. The faster such data is disseminated, the more time drivers have for their reaction. Road safety can also be improved by systems managed by machines processing the trajectories of vehicles and warning the drivers if they are heading to an accident.

The roadside infrastructure can be considered as a management system in VANETs. Current vehicular cloud computing (VCC) systems offer high computing power that is needed to analyze the data and send replies to vehicles. The problem is that the localization of those clouds is often distant from the drivers and, because of that, communication latency gets increased. Minimizing the latency is, thus, a critical issue and particular solutions, for instance, based on fog computing, mobile edge computing and cloudlets can be used to reduce it.

The Intelligent Transportation System (ITS) is viewed as an advanced architecture using a multitude of technologies (networking equipment, sensors, etc.) to manage the road traffic [2]. It enables the implementation of services to

The associate editor coordinating the review of this manuscript and approving it for publication was Zhe Xiao.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

improve road safety, reduce road congestion, travel time, etc. It is vital as there might be no possibility of redesigning the roads in the cities while the number of vehicles is getting larger. The use of ITSs allows for managing road traffic so that vehicles can move in a coordinated manner.

As an example, Traffic England [3] is a web platform implemented by Highways England's National Traffic Information to inform drivers about congestion and events on the roads. Drivers are, thus, able to plan their travel routes better and, because of that, the travel time can be reduced significantly. The operation of Traffic England includes gathering data from local police patrols and sensors (i.e., cameras) located on roads.

The deployment of photo radars to reduce the speed in specific areas can be considered as another example of ITS implementation [4]. The goal is to motivate drivers to reduce the speed and, as a result, improve safety on the road.

Communications between vehicles and traffic lights is investigated in [5], where vehicles are assumed to be equipped with OBUs to gather information from traffic lights, i.e., the distance to the nearest intersection, time to switching to green light, etc. A driver can use such information to adjust the vehicle speed and reduce the waiting time at the crossroad for the green light. The authors of [5] claim that, by using their approach, air pollution can be reduced by 5% while gas usage can be decreased by 18%.

Concerning the related literature, there are few surveys on the considered problem, and they address it in a limited way. In particular, Chen *et al.* provide a survey of ITS methods to improve cooperation between the roadside infrastructure units at intersections [6]. The available overview is divided into two categories: signalized and non-signalized intersections. The authors overview the systems for planning the trajectories of vehicles and the time/space reservation for entrance to crossroads. Additionally, they discuss the virtual traffic lights methodologies.

In [7], Lin *et al.*, apart from defining a smart parking ecosystem, present an overview of methods to facilitate vehicle parking. Their discussion also includes the type of sensors used to collect information about the parking lots and the schemes of information collection from sensor networks (information sensing, wireless sensor network, crowdsensing). The core part of [7] is the classification and description of existing methods of improving the parking process for connected vehicles.

Souza *et al.* explore in [8] the concepts of computational offloading in vehicular environments, and provide an overview of solutions for pushing the application execution to edge servers in cloud and fog environments. Additionally, the authors evaluate the methods according to the introduced taxonomy, referring to (1) the communication standards, (2) the goals pursued, and (3) the tests performed. Also, they present the classification and review of road traffic management systems in the context of congestion and accident detection/avoidance, traffic light management, route and speed suggestion [9]. The systems are classified according to the achieved goals and based on their dependency on the infrastructure.

Our paper is the first to provide a detailed analysis of a broad range of road traffic management architectures from the perspective of location and operation of computing units. In particular, the achievements of this paper are as follows:

1) Identification (in Section II) of the major use cases of applications and services supporting road traffic management followed by a discussion of the ETSI and C2C-CC requirements for message dissemination linked by us with the analyzed use cases.
2) Classification and analysis of road traffic management architectures from the perspective of the location of the computing units into the cloud-, cloudlet-, MEC- and fog-based followed by the analysis of their advantages and limitations presented in Sections III-VI.
3) Discussion of open issues and limitations in the deployment of each considered group of architectures addressed in this survey provided in Section VII.

II. TRAFFIC MANAGEMENT APPLICATIONS AND REQUIREMENTS FOR COMMUNICATIONS

Future traffic management systems may improve road safety and facilitate driving by using vehicular wireless networks. Dissemination of data to vehicles may serve a multitude of goals, for instance, increasing road awareness, supporting safety on roads, etc. In this context, vehicles could assist drivers in a sophisticated manner with such data. Autonomous vehicles may use the received information to make more accurate decisions. To support drivers, autonomous vehicles and road traffic management systems, standardization institutions (e.g., ETSI, C2C-CC) define the use of ITS units – devices that are dedicated to communicating with each other. Such units can be deployed in motorbikes, vehicles (e.g., determined as OBUs) and advanced infrastructure (e.g., RSUs). It is worth noting that also smartphones may be considered as ITS units of pedestrians.

This section first identifies the use cases of ITSs dedicated to road traffic management. Next, it provides the classification and analysis of the requirements for future ITS solutions concerning major road management scenarios.

A. TRAFFIC MANAGEMENT APPLICATIONS

The main goals of vehicular communications dedicated to road traffic management systems are improving safety on roads, enabling cooperation between vehicles and the roadside infrastructure, making the vehicular environment greener, and deploying autonomous driving systems for vehicles. In order to achieve these objectives, the major use cases include:

- **Traffic management at intersections**: OBUs and RSUs monitor traffic at intersections, detect potential accidents and notify other vehicles about the danger by sending proper messages. There are two types of intersections: signalized and non-signalized. The controller may manage the traffic lights at signalized intersections

MOST WIEDZY Downloaded from mostwiedzy.pl

depending on the local or global traffic situation. Traffic management at non-signalized intersections requires cooperation between vehicles and RSUs (if available) to assign priority of entrance for every vehicle.

- **Start/stop operations reduction**: communication between vehicles and the ITS to reduce the number of start/stop operations and improve the traffic flow at intersections. This use case may also contribute to a substantial reduction of air pollution.
- **Route planning**: cooperation between the vehicle's GPS device and the ITS to define the optimal route.
- **Global traffic management**: route definition for a large-scale group of vehicles to reduce congestion on roads, and react to unpredictable events like accidents.
- **Traffic and road awareness**: monitoring by the ITS followed by dissemination of information (i.e., congestion, unknown objects on roads, accidents) to the approaching vehicles.
- **Congestion avoidance**: traffic management by the ITS in a way to avoid the high density of vehicles on roads.
- **Reduction of the average waiting time**: cooperation between vehicles and controllers at intersections (RSUs) to reduce the idleness of vehicles (i.e., waiting until they can continue moving).
- **Platooning**: movement of vehicles one after another separated by a short distance.
- **Lane change assistance**: cooperation between vehicles to facilitate lane change operations and reduce collisions.
- **Parking management**: supporting the drivers at parking areas, i.e., locating free parking spaces and assistance during the parking maneuvers.
- **Emergency vehicles prioritization**: informing drivers about priority vehicles (i.e., police cars, ambulances) to create special corridors that facilitate the passing of those vehicles.
- **Green traffic management**: traffic management in a way to reduce fuel consumption and air pollution.

B. REQUIREMENTS FOR MESSAGE DISSEMINATION BY ITS UNITS

European Telecommunications Standards Institute (ETSI) and CAR 2 CAR Communication Consortium (C2C-CC) are responsible for defining standards, use cases, and requirements for vehicular communication technologies. The set of requirements includes (1) strategic, (2) economic, (3) system capabilities, (4) performance, (5) organizational, (6) legal, and (7) standardization requirements. This subsection concentrates on performance requirements for ITS units and examines the conditions that must be met in the design of vehicular network architectures.

The ETSI EN 302 637-3 specification [10] identifies the definition and requirements for Decentralized Environmental Notification Messages (DENMs) to inform road users about detected events in a decentralized manner. These messages can be forwarded to other units (multi-hop transmission) to spread the information in a broader area. DENMs include action ID, detection time, termination of the event, event location, etc. Any ITS unit may start the transmission, forward, and receive DENM messages. Also, ITS units should terminate the transmission of the DENM when the expiration time is reached (it is predefined before initiating the transmission) or when another ITS unit has sent the information about earlier event termination.

TABLE 1. Characteristics of CAM messages for communication between ITS units according to ETSI EN 302 637-2 document [11].

Requirement	Value
CAM generation rate	1-10 Hz
CAM generation rate (ITS infrastructure to vehicle)	1 Hz

TABLE 2. Requirements for sending Cooperative Awareness Messages according to ETSI TR 102 698 requirements [12].

Requirement	Value
Min. transmission frequency	1 Hz
Max. latency	500 ms
Min. transmission frequency for motorcycles	5 Hz

TABLE 3. Processing of Decentralized Environmental Notification Messages according to ETSI TR 102 698 requirements [12].

Requirement	Value
Minimal transmission frequency	1 Hz
Coverage range	600 m
Time to start	3 s
Expiration time of information	3000 s

The ETSI EN 302 637-2 specification [11] identifies Cooperative Awareness Messages (CAMs) properties that contain information on the location of ITS units as well as vicinity of other ITS units that are within a single-hop distance. The generation rates of CAM messages by ITS units are provided in Table 1. The use of CAMs may improve not only the road awareness, but also the traffic efficiency on the roads. For instance, the intersection controllers can manage the traffic flows better by knowing exact locations of vehicles.

Also, ETSI TR 102 698 document [12] explores the use cases and requirements in vehicular communication areas. It includes the requirements for sending CAMs, which enable V2V and V2I cooperation. Such messages should be sent in a single-hop manner and not be forwarded further to other ITS units. The generation frequency should be 1 Hz, and the transmission latency should not exceed 500 ms. The CAM include information on, e.g., passing the emergency vehicles, sudden accidents or breakdown of a vehicle in the vicinity. Table 2 presents the requirements for sending the CAM messages. Additionally, this specification also explores the requirements for transmission of DENMs, which are provided in Table 3. When comparing CAMs and DENMs, it should be noted that CAMs contain less information and are sent in a single-hop manner. As a result, the size of CAMs

MOST WIEDZY Downloaded from mostwiedzy.pl

is smaller, and they are disseminated within a shorter range. Such an approach enables ITS units to constantly update the most essential information to the nearest ITS units. DENMs, on the other side, contain more detailed information and are transmitted in a multi-hop manner. Additionally, these messages are disseminated only if one of the ITS units initiated transmission (when that station has detected an unexpected event or wants to evaluate a current event).

The ETSI TR 103 562 document [13] distinguishes Collective Perception Messages (CPM), which include sensory data (i.e., camera images) to analyze the environmental perception better. These messages should consist of data such as details of disseminating ITS units, their sensory parameters and detected objects. CPMs are meant to provide Collective Perception Service (CPS) – information about detected obstacles and road users shared by ITS units. CPS messages aim to improve safety on the roads by disseminating the data about the location of objects and road units which otherwise would not be able to exchange with others (i.e., bicycles, pedestrians, vehicles, etc.).

The ETSI TS 103 301 specification [14] analyzes infrastructure services and introduces the Traffic Light Maneuver service, which needs MAP (topology) Extended Messages (MAPEM) and Signal Phase And Timing Extended Message (SPATEM). Each MAPEM message consists of a topology/geometry set of lanes and provides topological details. Dissemination of MAPEM messages by vehicles to other ITS units (i.e., to intersection controllers) should help to improve the road traffic management at crossroads. It is because the intersection controllers can obtain detailed data about the positions of all vehicles (i.e., referring to lanes in which the vehicles are positioned, which is hard to deduce by having the geographical coordinates only).

A SPATEM message, in turn, includes data about the location and reception of specific users. Also, such messages may be sent by the intersection controller to the vehicles to inform the drivers about the signalling phases schedule. Using SPATEM messages by the roadside infrastructure enables the deployment of virtual traffic lights at intersections. MAPEM and SPATEM messages are further described in detail in the ISO/TS 19091:2017 standard [15].

Standardization of Platooning Control Messages (PCMs) is not available yet, and is expected as a future result. It is planned to be available in the ETSI TR 103 298 document. Such messages are awaited by cooperative driving applications responsible for coordinating the movements of vehicles in a group. Additionally, there is a need to define the Maneuver Coordination Message (MCM) parameters for coordination maneuvers between vehicles (i.e., during vehicle overtaking). More information is expected to be provided in the ETSI TR 103 578 document.

The C2C-CC Manifesto [16] presents the analysis of performance requirements for V2X communications based on use cases such as Cooperative Forward Collision Warning, Pre-crash Sensing and Warning, V2V Cooperative Awareness, V2V Decentralized Environmental Notification, I2V One Way Communication, and Local RSU Communication. Specific requirements for these use cases are presented in Table 4.

TABLE 4. Requirements included in the C2C-CC Manifesto 2007 [16].

Use Case	Communication Range	Communication Type
Cooperative Forward Collision Warning	20-200 m	-
Pre-crash sensing and warning	20-100 m	-
V2V Cooperative Awareness	300-1000 m	Broadcast, Geocast
V2V Decentralized Environmental Notification	300-20000 m	Broadcast, Geocast
I2V One Way Communication	500-20000 m	Broadcast, Geocast
Local RSU Communication	0-1000 m	Unicast

Road safety and efficiency for Cooperative Intelligent Transportation Systems (C-ITSs) and Cooperative Automated Driving is analyzed by C2C-CC in [17]. The types of messages transmitted in the ITS may differ depending on the continent. For driving awareness (i.e., traffic awareness, traffic management at intersections and emergency vehicle prioritization use cases), ITS units should support the use of (1) CAMs, DENMs and VRUs in Europe, (2) BSMs (Base Safety Messages) and PSMs (Personal Safety Messages) in the USA as well as (3) MAPEM, SPATEM in both regions. For driving sensing (i.e., road awareness, lane change assistance use cases), the CPMs are defined to be interchanged between the ITS units in both regions. For cooperative driving (i.e., lane change assistance and platooning use cases), MCMs and PCMs are defined to be interchanged between the ITS units both in the USA and Europe. This specification also defines packet sizes for each message type to support differentiated safety applications as follows:

- CAM messages – 400 bytes
- DENM messages – 1000 bytes
- VRU messages – 350 bytes
- CPM messages – 1000 bytes
- SPATEM and MAPEM messages – 1200 bytes
- PCM messages – 400 bytes
- MCM messages – 1000 bytes
- BSM messages – 380 bytes
- PSM messages – 350 bytes

The ETSI TS 101 539-1 specification [18] presents requirements for Road Hazard Signalling based on Cooperative Awareness and Decentralized Environmental Notification (V2V communications). Road Hazard Signalling enables avoiding collisions on roads by sending warning messages to drivers. From that document perspective, an essential requirement is that an ITS unit should start disseminating the warning notifications to drivers at most 2-5 seconds before an accident. It is assumed that end-to-end communication delay is not greater than 300 ms and the packet loss ratio is not greater than 5%. Additionally, it is required that ITS devices (such as OBUs) are capable of processing 5000 messages per second and send the up-to-date CAM and DENM messages every 100 ms–1 s.

MOST WIEDZY Downloaded from mostwiedzy.pl

The ETSI TS 101 539-2 specification [19] explores requirements for Collision Risk Warning at Intersections (ICRW). Whenever an accident is detected by a vehicle or another ITS unit (such as an RSU), that unit starts transmitting CAM and DENM messages to all endangered vehicles. The communication range should be at least 300 m, while the update frequency should be 10 Hz. All ITS units (both OBUs and RSUs), should be able to process 1000 messages per second.

The ETSI TS 101 538-3 document [20] identifies requirements for Longitudinal Collision Risk Warning (LCRW). The goal of LCWR applications is to warn drivers about vehicles in the same lane, i.e., during overtaking maneuvers, road works, accidents ahead, etc. Such applications should use CAM and DENM messages. It is also required that the communication range is at least 300 m and that the units should be able to process 1000 messages per second.

The ETSI TS 103 300-2 specification [21] explores requirements for Vulnerable Road Users (VRU) awareness. Road users analyzed in [21] are broadly defined as pedestrians, vehicle drivers, scooter drivers, bicycle drivers, etc., and equipped with such ITS devices as OBUs or mobile smartphones. An important requirement for any ITS is that it should be capable of operating effectively with up to 5000 active users within 300 m circular radius of communication range. Every ITS device should be able to process 1000 messages per second, and, if a danger is detected, transmission should be started at most after 100 ms. Additionally, the communication delay is required to be lower than 5 ms for perfect communication conditions (no obstacles, good weather).

To conclude the requirements described in this section, there are differentiated types of messages defined in the respective documents. Also, specific message types should be used for particular services. The two main types of requirements are (1) transmission requirements and (2) performance requirements. The first group refers to the communication range, intervals of updating data and communication types, while the second one specifies the size of transmitted messages and the processing capabilities of ITS units. These requirements may also vary for different services, which must be considered when designing vehicular communication architectures.

C. REQUIREMENTS FOR MESSAGE DISSEMINATION CONCERNING MAJOR USE CASES

The ITS units, which are forming the roadside infrastructure designed to support cooperative awareness applications, should be capable of processing 5000-7000 messages per second. In a pessimistic scenario, according to our estimations based on data provided in this section, each of these units should process such messages at 2.8 Mb/s. Also, ITS units should support CAMs and DENMs simultaneously. Supporting DENMs requires an additional 7 Mb/s per station, while supporting both CAM and DENM messages requires 9.8 Mb/s per station. Vulnerable Road Users awareness application assumes that any ITS unit can operate at 0.35 Mb/s.

To summarize these assumptions, if the ITS design includes supporting CAM, DENM and VRU messages, the roadside infrastructure units should be able to operate at 10.15 Mb/s. Actually, there is no information about the performance requirements for other types of messages, i.e., PCMs, MCMs, MAPEMs and SPATEMs. Therefore, it is necessary to assume that these units must be prepared to operate at rates even higher than 10.15 Mb/s.

Concerning the traffic management use cases highlighted in Section II-A, each use case may refer to different properties, system performance, messages interchanged between vehicles, etc. Therefore, Section II-B presented only general requirements related to message dissemination. ETSI and C2C-CC provided several types of messages to enable traffic management applications in an ITS. Because of that, disseminating intervals, communication coverage, and communication delay may vary.

As identified in Table 5, the ITS communication infrastructure has to meet a large set of expectations. Although supporting differentiated types of messages is not a challenge, designing the system so that the transmission delay values do not exceed the assumed upper limits can be a huge problem. The highest delay in message transmission allowed to occur is 600 ms, which is fully justified regarding the security use cases. This assumption makes moving part of the infrastructure closer to users reasonable. On the contrary, a lot of computing power must be provided in the infrastructure segment to ensure fast processing of tasks. For this reason, simply moving the infrastructure to the edge of the network may not solve all the problems. Therefore, designing an efficient system may imply trade-offs between reducing communication delay and computing power.

A wide range of application deployment scenarios, a large number of users and stringent (often real-time) requirements make it hard to design vehicular network architectures offering road traffic management functionalities. However, as discussed in several research papers by the community, solutions based on cloudlets, MEC, cloud and fog can help mitigate this problem. Furthermore, as the literature shows, several system concepts for VANETs are already available. They are analyzed in detail in the following four sections of this paper.

III. VEHICULAR CLOUD COMPUTING

Cloud systems consist of data centers that offer power resources on demand. A variety of devices can connect to the cloud system to run the application on the cloud side, as presented in Fig. 1. The main feature of cloud systems is providing high computing power and storage capacity. Such possibilities allow recording massive historical and real-time data for use by predictive methods, machine learning, statistical schemes, Big Data methods and many more. Therefore, cloud infrastructure is considered as a supporting system of the vehicular network (as presented in Fig. 2). However, there exist significant disadvantages to using cloud technology. For instance, the location of cloud servers is often unknown and distant from end-users. Additionally, a considerable distance

MOST WIEDZY Downloaded from mostwiedzy.pl

TABLE 5. Correlation between the use cases and requirements.

Use Case	Type of Message	Max. Latency	Min. Frequency	Coverage Range	Messages Processed per Second
Traffic Management at Intersections	CAM, DENM, CPM	500-600ms	1-10Hz	300m	5000
Start/Stop Reduction	MAPEM, SPATEM	1000ms*	1-10Hz*	0-1000m	5000*
Route Planning	MAPEM	1000ms*	1-10Hz*	0-1000m	5000*
Global Traffic Management	MAPEM	1000ms*	1-10Hz*	0-1000m	5000*
Traffic and Road Awareness	CAM, DENM, VRU, CPM	500-600ms	1-10Hz	300m	5000
Congestion Avoidance	CAM, DENM	500-600ms	1-10Hz	300m	5000
Reduction of the Average Waiting Time	MAPEM, SPATEM	1000ms*	1-10Hz*	0-1000m	5000*
Platooning	PCM, MCM	500ms*	1-10Hz*	300m*	5000*
Lane Change Assistance	CPM	500ms*	1-10Hz*	300m*	5000*
Parking Management	MCM	1000ms*	1-10Hz*	300m*	5000*
Emergency Vehicles Prioritization	CAM, DENM, CPM	500-600ms	1-10Hz	300m	5000
Green Traffic Management	–	–	–	–	–

* Predicted values are marked with an asterisk and are not included in the standardization documents.

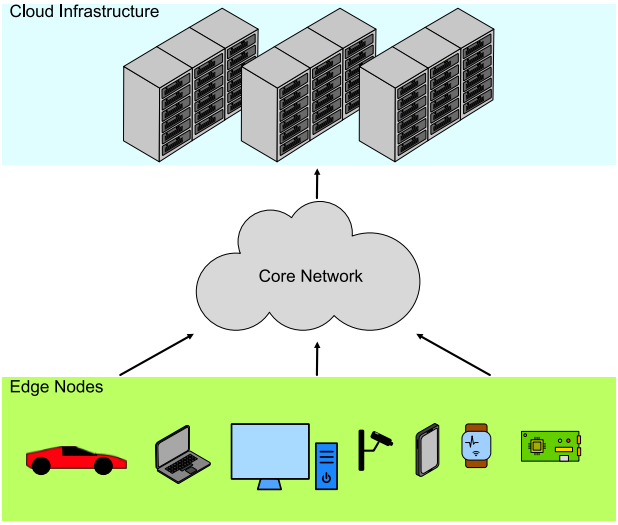

FIGURE 1. An example scheme of a cloud network.

between the user and the cloud infrastructure can cause long transmission delays. Applications related to the road safety of users (i.e., pedestrians, drivers) are a particular case in which the execution time (including the time of requesting, processing and sending the result) is critical. The advantages of cloud systems allow the implementation of use cases such as route planning at intersections, global traffic management, congestion avoidance, average waiting time reduction, parking management and emergency vehicles prioritization.

In [22], Kong *et al.* propose an auction-based method for parking space sharing. The main goal of the solution is to create a platform in which drivers can hire public parking spaces or exchange private parking spaces from the owner for a specified price. This auction system additionally optimizes benefits for parking space owners. It requires a cloud platform responsible for registering parking spaces by owners who want to share the parking space for specified time slots and manage the available parking spaces for drivers. The cloud platform uses the price-compatible top trading cycles and chain mechanisms to enable users to exchange their parking spaces. If parking space assignment fails, a one-sided Vickrey-Clarke-Groves auction is executed to assign parking spaces for drivers. Also, exchanging parking spaces mechanism should motivate drivers to share their parking spaces with others. It is fully justified to use the cloud infrastructure as the number of requests for parking lots is often huge considering the need to handle requests from vehicles across the city.

In [23], Lin *et al.* propose the Smart Parking Allocation algorithm (SPA) to maximize the utilization of parking lots. The SPA system needs to store the data about all parking lots and the data about the behavior of all drivers (i.e., parking time value) considering each parking lot. Therefore, cloud infrastructure is needed to store all the required information and assure the power computing resources to be able to operate. The SPA system includes three assignment policies: worst-fit, best-fit, and parking behavior forecast. The worst-fit policy selects the vehicle predicted to remain the longest time at the parking lot. The best-fit strategy finds the vehicle that will fully utilize the parking lot availability. In other words, it is about subtracting the predicted parking time of the vehicle and the availability time of the parking lot – the vehicle with the lowest subtraction output is selected. The parking behavior forecast analyzes the parking history and the parking traffic. The system tries to assign the vehicle to the parking lot by adjusting the vehicle's behaviour to the parking traffic.

In [24], Noreikis *et al.* introduce a guiding system to select a parking slot for the driver and transfer the driver to the public communication area. This proposition requires the global traffic awareness capability from the unit that will process the data. Considering the massive amount of vehicles on city roads, cloud infrastructure is needed as it can store and process a huge amount of data. The system consists of (1) cloud API to get information about public transport and plan the route for drivers to minimize the travel time, (2) in-vehicle Head-Up Display (HUD) to show the driver how to get to public communications from the parking space, and (3) Android application for smartphones that gives additional guidance about transferring to the public transport

MOST WIEDZY Downloaded from mostwiedzy.pl

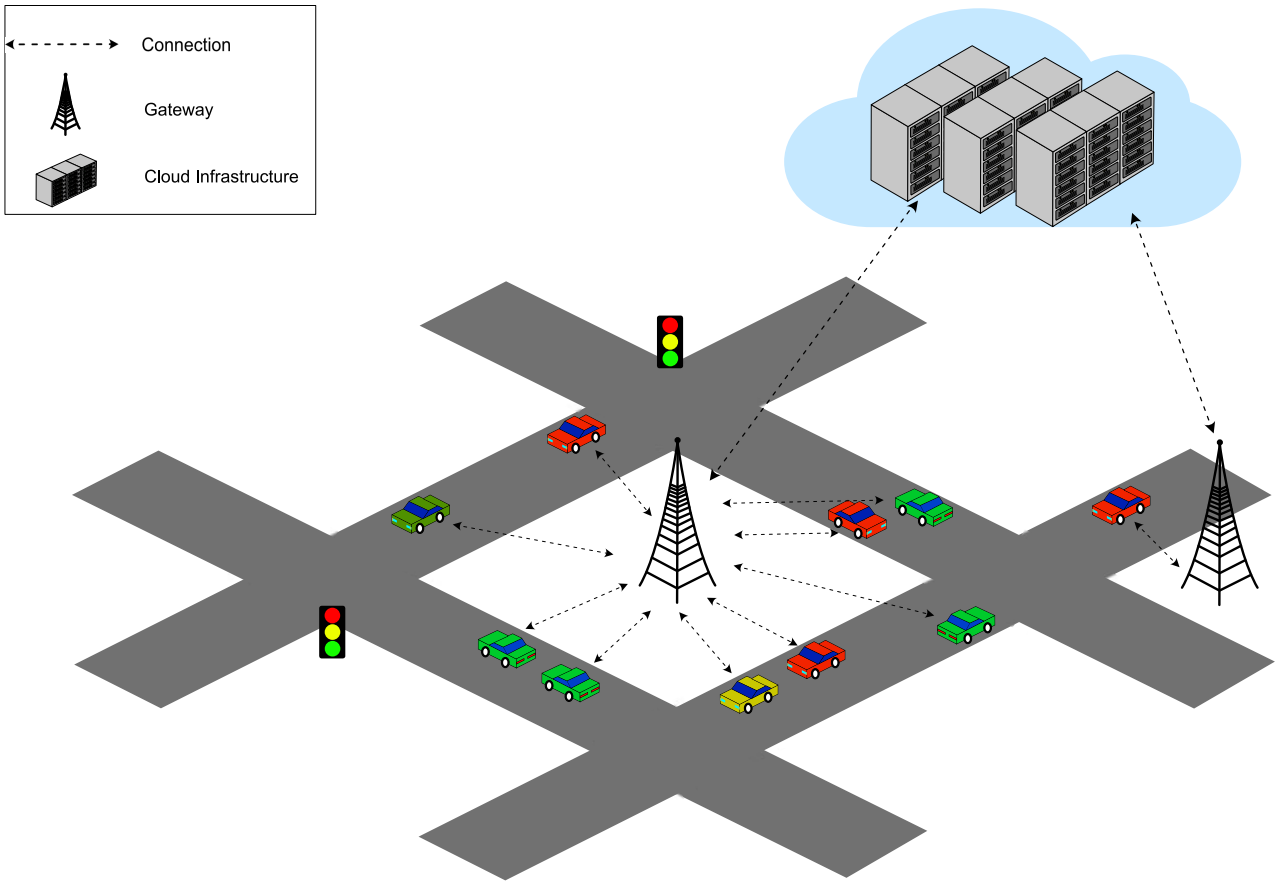

FIGURE 2. An example diagram of the network operation for the cloud architecture.

system. The work also provides a solution to reduce air pollution by splitting the driver's journey into a private vehicle trip and public transportation travel.

In [25], Khalid *et al.* propose a cloud-based system for computing routes for emergency vehicles and evacuees. The system requires RSUs to gather information about the volume of vehicles on the roads, their average speed, road conditions, etc., from the road sensors. These data should be transmitted from RSUs to the cloud platform, which next computes the route costs in real time and sends the suggested routes to evacuees if a disaster happens (i.e., a tornado, flooding, etc.). First, route costs are calculated for every road segment depending on the presence of emergency vehicles (as this type of vehicles has the highest priority) and by using the parameters such as road capacity, congestion ratio, top speed of vehicles and the maximum allowed speed on the road. Then, the path (to the nearest shelter) with the lowest sum of the total route cost is selected and sent to the evacuees to suggest the best evacuation route.

In [26], Li *et al.* introduce a cloud-based route planning approach to reduce the risk of accidents during the journey. The authors first define a neural network algorithm to forecast the crash probability for each road based on a multitude of road properties such as pavement roughness, speed limit, segment length, number of lanes, average daily traffic, the width of road lanes, the curvature of the road, the grade of the road and weather conditions. Next, an analysis of the paths for the vehicle is executed where the path with the minimal sum of road weights (based on travel time and risk of crash) is selected. Applying the cloud infrastructure is entirely understandable as this system uses a neural network algorithm (which is computationally consuming) and operates on many differentiated data (the historical data and at least nine types of real-time data).

In [27], Fanti *et al.* describe a route planning system designed to reduce fuel consumption and air pollution of heavy-duty vehicles (i.e., trucks). It needs the cloud infrastructure to store all the urban and rural roads data. The system consists of (1) an OBU in a truck that sends the data (i.e., payload, waypoints) to the cloud and (2) the cloud system that stores additional information about roads (i.e., traffic, speed limit). As soon as path selection is requested (by a truck driver or a logistic company), depending on the considered parameters, the cloud system determines three options, namely (1) the fastest route, (2) the alternative one that is limited to the maximum arrival time and (3) all possible routes between the actual location of the truck and the destination. Every road is next analyzed based on the estimated fuel consumption, air

MOST WIEDZY Downloaded from mostwiedzy.pl

TABLE 6. A summary of vehicular cloud computing (VCC) properties for the state-of-the-art solutions.

Ref.	Importance of Low Delay	Computational Complexity	Type of Data	Algorithms	Use Case
[22]	Low	Low	Dynamic	Auction-based	Parking management
[23]	Low	Low	Historical	Greedy algorithm	Parking management
[24]	Low	Low	Dynamic	Custom	Parking management; global traffic management
[25]	Critical	Medium	Dynamic	Custom	Route planning for vehicles; emergency vehicles prioritization; global traffic management
[26]	High/Medium	High	Historical	Neural networks	Route planning for vehicles
[27]	Low	High	Dynamic/Historical	Big Data methods	Route planning for vehicles; green traffic management
[28]	Medium	Low/Medium	Dynamic/Historical	Dijkstra's	Route planning for vehicles; green traffic management
[29]	High	High	Dynamic/Historical	Temporal Convolutional Network; Dynamic Bureau of Public Road; Dijkstra's	Route planning; emergency vehicles prioritization
[30]	Medium/Low	High	Dynamic/Historical	Custom	Route planning for vehicles; global traffic management; congestion avoidance
[31]	High	High	Dynamic	Deep Learning and Extreme Machine Learning	Congestion avoidance; global traffic management

pollution, and speed profile. Finally, the most eco-friendly route is selected and sent along with information about the speed to the truck driver.

In [28], Boriboonsomsin *et al.* propose a navigation system for planning the journey in an environmentally-friendly manner. The authors developed a dedicated database to store the digital map with historical and real-time data about traffic and road measurements. The energy emission factors are generated based on information from road measurements. Then, the EOPS (Energy/Emissions Operational Parameter Set) vector of values is calculated for every route, which includes such parameters as (1) vehicle characteristics, (2) road characteristics, (3) traffic characteristics, and (4) other variables (i.e., driver characteristics). Next, route calculation is executed based on Dijkstra's algorithm. Finally, a path with the lowest emission cost is searched, and the result is sent to the driver. Operating on historical and real-time data demands a lot of storage resources. Therefore, the system requires cloud infrastructure to assure reliability.

In [29], Jin *et al.* propose a route planning method for emergency vehicles on a highway. This technique requires historical data about the travelling time for road segments and the road traffic density to predict the future traffic volume using the Temporal Convolutional Network (TCN). The road impedance values (function of travelling time, the volume of traffic and the capacities of each road segment) are generated by running the Dynamic Bureau of Public Roads function on the output of TCN and the real-time data about the actual traffic volume. Dijkstra's algorithm is used to find the best route by selecting the one with the minimal value of road impedance. Cloud infrastructure is fundamental for this system because storing the historical data is memory-exhaustive, and it must accessible at any time.

In [30], Guidoni *et al.* present a traffic management system named Re-RouTE. The main goal of Re-RouTE is to detect the congested routes and next offload them. It is required that vehicles send the data (i.e., their actual position, destination) to the cloud. The cloud is expected to analyze a massive amount of data. The classification of the routes is based on the road congestion ratio. The Re-RouTE service should be executed in the cloud to classify the routes based on their congestion ratio, considering the traffic engineering theory approach. The system finds alternative routes for vehicles already in traffic jams and for vehicles about to reach the road traffic congestion areas.

In [31], Shengdong *et al.* present a cloud-based system for minimizing congestion on urban roads. The authors use deep learning algorithms combined with the Extreme Learning Machine (ELM) algorithm to forecast the future density of vehicles on roads. The respective algorithm dispatches vehicles among roads using Floyd's shortest path algorithm combined with the previously predicted traffic volume. The main goal of such an approach is to balance the traffic on roads evenly by using future predictions of congestion on each road and selecting different paths for every vehicle. The system operates on a high volume of data (taken in real time) to forecast the density of vehicles on roads. Therefore, it requires cloud infrastructure to ensure computing power and storage resources.

As mentioned in this section, the cloud infrastructure offers enormous computing power and huge memory space. It is due to many storage servers, high-end CPUs and GPUs unit deployment. In the context of road traffic management support, the only disadvantage of cloud infrastructures is a theoretical long communication delay between the cloud and the end-user.

As summarized in Table 6, solutions proposed in [22]–[24], [27], [30] are less sensitive to latency. They are more flexible concerning the execution of services at the cloud infrastructure side, especially if high computational power is required. In our opinion, other applications such as those proposed in [25], [26], [29], [31] are more sensitive to

MOST WIEDZY Downloaded from mostwiedzy.pl

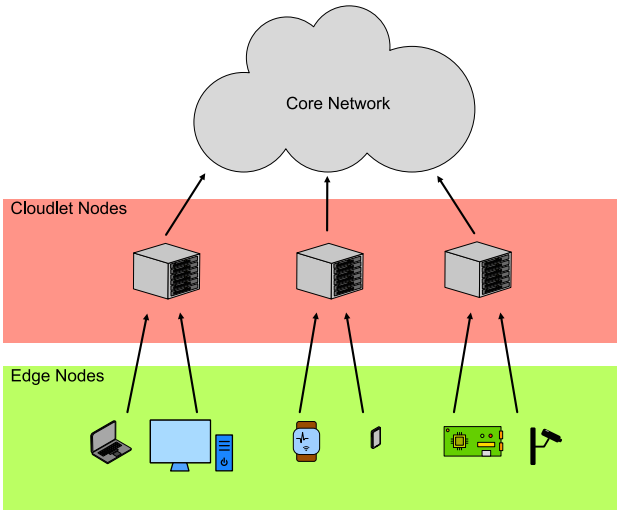

FIGURE 3. An example scheme of a cloudlet network.

communication delays, as they require fast decision-making during service provisioning. Therefore, it is necessary to assess whether it is possible to implement intermediate servers (e.g., MEC servers), which would support the cloud infrastructure to reduce communication delays. Applications from [29], [31] require special attention because they use dynamic data (which need to be processed in real time) and are characterized by high computational complexity. Large communication delays can also make it difficult to deliver these services in real time. Again, the support of intermediate servers could have a positive impact but would require the respective re-evaluation and redesign of applications.

IV. CLOUDLETS

For the first time, the idea of cloudlets was presented in 2009 by Satyanarayanan *et al.* in [32]. The concept of cloudlets is understood as the deployment of servers at the nearest location to the end-users. The main objective is to shorten the distance between the servers and users and to improve application execution at the edge of the network, as presented in Fig. 3. Although the computing power of cloudlets is lower than that of the cloud, a short distance between servers and end-users enables to reduce the communication delay significantly, making the execution of applications considerably faster. The next crucial aspect is that each vehicle has limited resources that can be used during the journey to support driving assistance, communication with other vehicles or networks (Internet), etc. Therefore, the more applications are processed outside the vehicle (e.g., offloaded to the cloudlet), the more energy will be preserved and used for other functions of a vehicle. Additionally, what is even more important, the cloudlet and its proximity to vehicles (as in Fig. 4) reduce communication delay [33], which is critical for road safety and traffic management.

When designing an application, it is often assumed that its execution will be on the client's side. Applications that require high computing power should also consider offloading their execution (or part of it) to the cloudlet in the vicinity to support, e.g., image processing, multi-path analysis, and prediction methods. As these operations should be performed in real time, it is suggested to offload their execution to the neighboring servers, characterized by higher computing power and connected to reliable power sources. When redesigning any application, special attention should be paid to the communication requirements to meet the criteria outlined in Section II, emphasizing the importance of low communication delay and frequent updates.

At the time of preparations of this paper, no traffic management method based on a cloudlet system was available. However, many driving assistance applications consider their execution at vehicle OBUs. Therefore, it is necessary to reevaluate such applications to check if it is possible to offload the application execution to the nearest cloudlet server. As an example, Keivani *et al.* [34] evaluate several vision-based driver assistance systems that process images from the front camera to detect and notify the drivers about any hazardous conditions. As such methods require high computational power for processing the images, offloading them to edge servers would have a considerable impact on the vehicle's energy. Therefore this section aims to take a closer look at the existing driving assistance methods executed at the vehicle's side and verify if the execution of those applications could be offloaded to the cloudlets successfully. In other words, in this section, we discuss OBU applications that, in our opinion, are suitable to be redesigned in a way to become supported by a cloudlet system.

In [35], Zeng *et al.* introduce a scheme of eco-routing of vehicles to find a path that minimizes the fuel consumption and, as a result, implies a much-reduced amount of air pollution. The algorithm develops a fuel consumption model for each path. The respective Lagrangian relaxation heuristic is proposed to find the most eco-friendly paths for vehicles. The algorithm is executed by OBUs of vehicles, but it may require high computational power to analyze many paths. Therefore, it would be good to run such a service outside the vehicle. Offloading it to the cloudlet infrastructure may improve the overall execution time.

In [36], Chakraborty and Datta propose two methods: (1) for finding the shortest path of the journey for vehicles and (2) for supporting the drivers in collision avoidance. To find the shortest travel route, the ant colony algorithm is used by examining the two paths: (1) from the source to the destination and (2) the reverse path (from the destination to the source). Then, the repulsive vector field around the objects is created by analyzing the data from sensors (i.e., a camera or a LiDAR sensor), which can be another vehicle or an obstacle on the road. In our opinion, the efficiency of this scheme could be increased by pushing the execution of these two methods to the cloudlet server. Then, the vehicles would simply send the requests (with the required data) for the service.

MOST WIEDZY Downloaded from mostwiedzy.pl

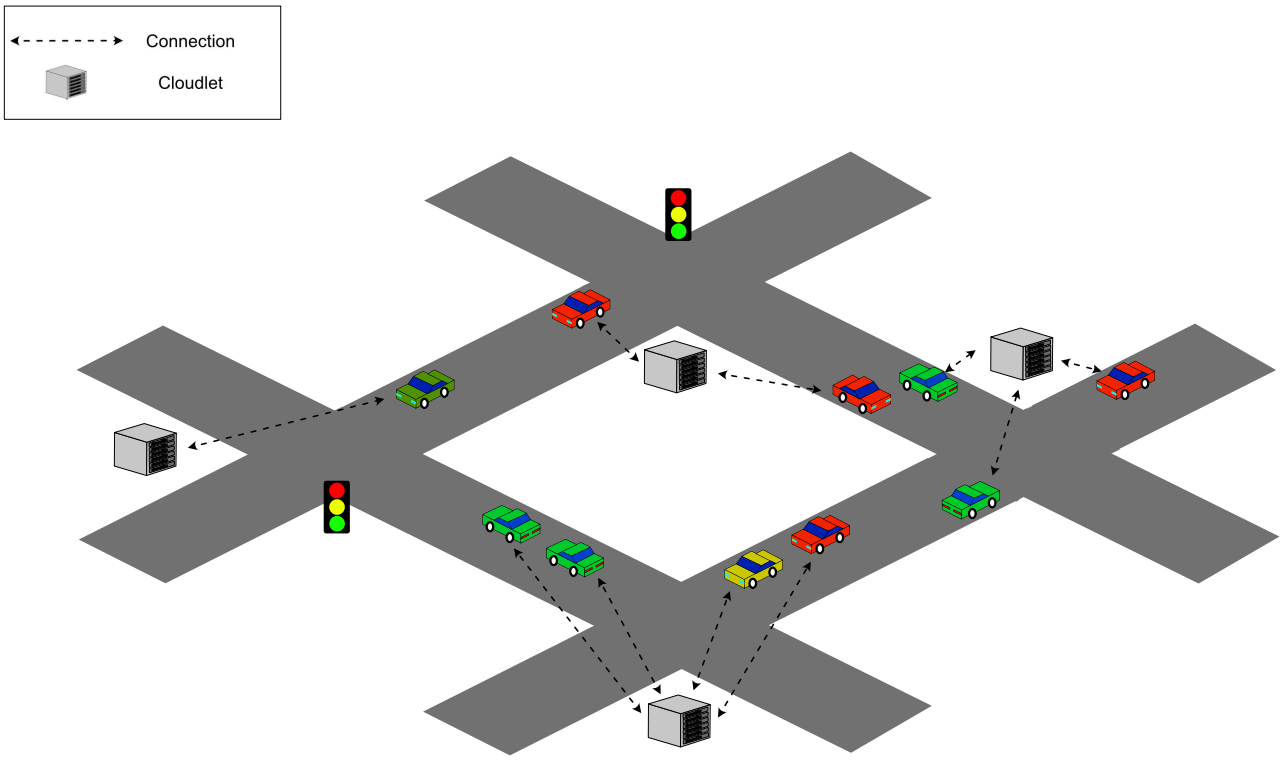

FIGURE 4. Example of a vehicular network operation for a cloudlet architecture.

In [37], Shashua *et al.* propose a single-frame classifier for pedestrian detection to support driver awareness. The architecture of this system is divided into three modules. The first module is to generate candidate regions of interest by selecting 75 windows per frame to feed the classifier. The next one – single-frame detection classification – analyzes those frames to detect pedestrians. Finally, the third one proceeds with the multi-frame approval process analyzing the images chosen by the single-frame detection classifier to confirm the pedestrian location. As this detection system requires high computational power due to constant video analysis, offloading such a service to a cloudlet server could save the vehicle's energy.

In [38], Liu *et al.* propose a night-time pedestrian detection method. This solution requires a monocular far-infrared camera that records images of the environment. The Pyramid Entropy Weighted Histograms of Oriented Gradient (PEWHOG) is used to capture the shape distribution of pedestrians and the spatial layout. During the last stage, the Support Vector Machine modelling is used to detect pedestrians using the PEWHOG result as an input. An interesting feature would be to stream the camera recordings to the nearest cloudlet server to execute the PEWHOG method and perform the Support Vector Machine operation. Then, the server would send back just the output (i.e., information on detected pedestrians) to the vehicle.

In [39], Arbabzadeh and Jafari propose a real-time risk prediction and classification method for improving safety on the roads. The algorithm uses information about the characteristics of drivers (i.e., their sleep habits, driving history, risk taking, driver behavior in the past, etc.), journey parameters and the current localization. Those parameters can be modified by adding, changing, or removing input parameters as the scenarios on the road may vary and, because of that, it requires elasticity from prediction methods. The method deploys a regularized multinomial logistic regression model to detect and classify the state of risk into normal driving, near collision and collision. There are many parameters to analyze, so it might be power-exhausting for the vehicle to run such a service. Pushing the service execution to the cloudlet server could thus save the energy of vehicles.

In [40], Song *et al.* present the method of detection and classification of objects on the roads to disseminate collision warnings to drivers. This method uses an odometer algorithm to process the frames (captured by the front camera) to find lateral objects and then combines the output with information from the millimeter-wave radar to detect longitudinal objects. Such an attitude helps to identify the dynamics of objects and classify the risk. In our opinion, the vehicle could stream the data to the nearest server with the request to analyze and detect the objects. Furthermore, we believe that if more than one vehicle requested the service, the cloudlet could execute the detection method once and send the suggestions to all relevant vehicles.

In [41], Duan *et al.* propose a framework to improve the vehicle environmental awareness at intersections.

MOST WIEDZY Downloaded from mostwiedzy.pl

TABLE 7. A summary of cloudlet properties for the state-of-the-art solutions.

Ref.	Importance of Low Delay	Computational Complexity	Communication Type	Energy Saved	Use Case
[35]	Medium	Medium	Unicast/Broadcast	Low	Route planning for vehicles; green traffic management
[36]	Medium/High	High	Unicast	High	Route planning for vehicles; traffic and road awareness
[37]	Critical	High	Unicast	High	Traffic and road awareness
[38]	High	Medium	Broadcast	Medium	Traffic and road awareness
[39]	High	Very High	Unicast	High	Traffic and road awareness
[40]	High	High	Unicast/Broadcast	High	Traffic and road awareness
[41]	High	High	Unicast	High/Medium	Traffic and road awareness
[42]	High	Low/Medium	Unicast	Low	Lane change assistance
[43]	High	Heigh/Medium	Unicast	Medium	Lane change assistance
[44]	Low	Low	Unicast	Low	Start/stop reduction; green traffic management

An autonomous vehicle that intends to pass an intersection is expected to connect to an RSU terminal to receive a map of objects at the intersection. Then, the vehicle's OBU should run the Normal Distribution Transform (NDT) algorithm to determine its location (in the received map of objects) and send its position to the RSU. The vehicles have information about their surroundings from three sources, including (1) built-in cameras, (2) LiDAR sensors and (3) RSU (a received map of objects). The 3D Point Cloud Object Detection Algorithm (RGB-PVRCNN) is proposed to unify the output from all three sources and return the 3D target detection results. However, the authors report that such an algorithm requires a lot of computing power and computation at the roadside edge server should be applied. In this case, vehicles could request from the cloudlet server to run the RGB-PVRCNN algorithm to get a unified 3D map with detected targets.

In [42], Hou *et al.* present a method for assisting in lane-change maneuvers. The authors use the Bayes classifier and a decision tree algorithm to detect if a change lane maneuver is needed or not. Additionally, the method that combines both algorithms mentioned above is presented. These approaches require information about the difference in the speed between (1) a lead vehicle and a merging vehicle (the one that changes the lane) and (2) a lag vehicle (the one that is behind the merging vehicle) and a merging vehicle, as well as (3) the distance between the beginning of a merge lane and a merging vehicle. In this case, the cloudlet server could have up-to-date data (i.e., about the position and speed) from all the vehicles in the area. Running the Bayes classifier and decision tree algorithm on the cloudlet server side and sending the suggestions to the vehicles and/or drivers could save the energy of vehicles.

In [43], Sun *et al.* present an adaptive algorithm to warn the drivers during the lane change maneuvers. The algorithm requires information about the distance between vehicles, their acceleration and road characteristics to disseminate warning messages. Additionally, the method includes observing the driver's characteristics and reactions for proper construction of lane change warnings to the driver as every driver may react to the signalling differently. We suggest redesigning the method so that a vehicle could send the driver's characteristics, vehicle speed and acceleration to the cloudlet server and ask for lane change maneuver suggestions.

In [44], Lin *et al.* propose a macroscopic traffic model for an individual vehicle with the main goal to predict the traffic pattern using the information on the speed and acceleration of vehicles to reduce air pollution and travel time. The traffic prediction flow algorithm is run at OBUs. However, we think that the execution of service could be offloaded to the nearest cloudlet server.

Following the above-mentioned properties summarized in Table 7, it is necessary to investigate in greater detail the requirements for OBUs to provide road traffic management services. The solutions described in [36]–[40], [42]–[44] require low communication delay as they also impact road safety. Solutions [36], [37], [39]–[41], [43] require relatively high computing power. Therefore, the execution of these services creates a demand for a large amount of energy. Redesigning the above-mentioned methods to redirect the execution to the cloudlet infrastructure would support fast communication and save the energy of vehicles while providing the services, the computational complexity of which is non-trivial. In our opinion, the proposal from [35] does not require high computing power and is relatively resistant to high communication delays. This provides flexibility when implementing such a service because, depending on the needs of end-users, it can be implemented at the OBU, a cloudlet side, or even in the cloud.

V. MOBILE EDGE COMPUTING

Mobile Edge Computing (MEC) was introduced by the European Telecommunications Standards Institute (ETSI) and Industry Specification Group (ISG) [45]. The architecture defines the services at the edge of a mobile network by deploying mobile edge stations near the end-users as presented in Fig. 5. The main goal is to create a network with ultra-low latency, real-time access and high bandwidth. Regarding the network architecture, the mobile edge stations are in between the cloud infrastructure and end-users located at the edge of the network. The reason for such a solution is to shorten the distance between end-users and the MEC station to minimize the communication latency. If more computing power is needed, or there is another reason, e.g., lack of information in the cache about a specific application, the MEC station redirects requests of the end-user to the cloud in order to provide the required service.

MOST WIEDZY Downloaded from mostwiedzy.pl

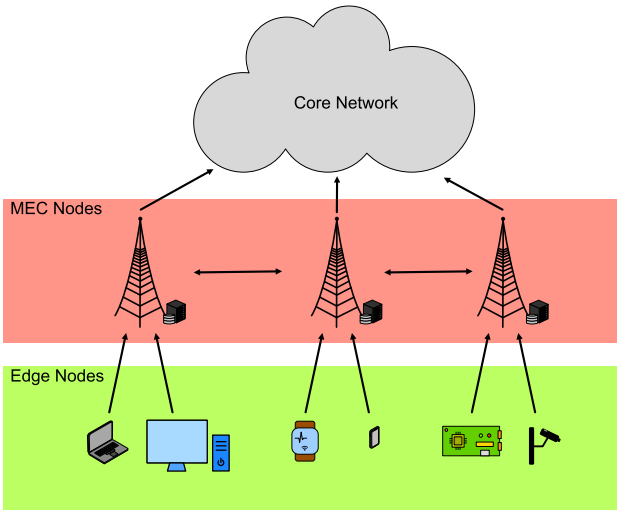

FIGURE 5. An example scheme of a MEC network.

In the context of vehicular communications and road traffic management, it is assumed that the RSUs are configured as the access points for vehicles (as in Fig. 6). Additionally, RSUs can have the role of a root of trust. Therefore, they may be treated as the main traffic coordinators since MEC systems may assume the use of RSUs, gateways and MEC servers to support drivers during their journey. Unlike the cloudlet infrastructure, the MEC infrastructure can participate in managing the traffic on the roads (by deploying the support of the cloud system). An important value is the introduction of one main traffic coordinator by ensuring the aggregation of data about the state of roads and the use of information for management in a safe and efficient manner. Also, the MEC system can be flexibly configured to have additional communication with the cloud and/or cloudlet infrastructure. Mixing the architectures could broaden the capabilities of the whole system, and, as a result, more use cases could be realized.

In [46], Bento *et al.* proposed a road traffic management model to coordinate vehicles at intersections based on the vehicle type. The first one is called the legacy vehicle that is not equipped with any V2V/V2I communication system. The second type can communicate wirelessly with other vehicles and the infrastructure (i.e., RSU, intersection controller). One algorithm out of three is selected by the intersection controller depending on the vehicle type and intersection situation. The first two algorithms are Waiting Method for Intelligent Traffic Management (WMITM) and the Early Method for Intelligent Traffic Management (EMITM) respectively. They both require that all vehicles can communicate wirelessly with other vehicles and the infrastructure, and assume that vehicles send information about their speed, position, destination and vehicle size to the intersection controller. The speed profile is next generated and sent back to vehicles. EMITM adjusts the speed profile of the vehicle based on the fastest path, while WMITM schedules the vehicle's path after the latest reserved path of other vehicles. The third algorithm is the Legacy Early Method for Intelligent Traffic Management (LEMITM). It uses sensors on the roads to detect incoming and outgoing vehicles. In this case, vehicles do not need to be equipped with any communication technology. The LEMITM algorithm detects the incoming vehicles, computes all possible trajectories of vehicles, and schedules the green/red phases. The intersection controller manipulates traffic signals based on LEMITM analysis. Although it is not stated directly, this method needs an RSU (as an intersection controller) to command the vehicles when they enter the intersection.

In [47], Shi *et al.* define a real-time algorithm to orchestrate the vehicles at intersections. At first, vehicles send requests to the intersection controller. The request should include vehicle position, speed, acceleration, etc. Then, based on the data about all vehicles trying to pass the intersection, the intersection controller calculates the passing priorities of vehicles (by checking the road lane class), the required intersection passing time, and the waiting time of the first vehicle in the lane. As a result, signalling at the intersection is adjusted to the priority of passing vehicles (previously calculated). This method assumes that the intersection controller can command the vehicles entering the intersection. Therefore, the controller should be treated as an RSU server which guarantees that vehicles (or drivers) will obey the commands of a controller.

In [48], Steinmetz *et al.* propose the Collision-Aware Resource Allocation (CARA) algorithm to manage vehicles crossing the intersection while using the reduced amount of communication resources. The method uses the intersection manager unit to support vehicles at intersections and assign communication resources to every vehicle. Managing the vehicles at intersections includes computing the Collision Possibility Indicator (CPI) – the probability of collisions between pairs of vehicles. The CPI is used to calculate the recommended speed and deceleration values at each time, which are sent to vehicles. Also, the CPI is calculated for each timestamp during the prediction horizon to assign communication resources. The result is used as an input to the integer program, which next calculates the latest possible time interval to communicate with each vehicle to update the suggestions.

In [49], Wuthishuwong *et al.* introduce a traffic management model at intersections to manage the incoming vehicles which are about to cross the intersections without traffic lights. The model assumes that vehicles send request messages to the intersection controller and this intersection controller calculates and reserves a proper time slot to cross the intersection for every vehicle. Based on vehicles speed, destination and location, the intersection controller discretizes the distance to the intersection, trajectory and time in order to calculate the time slot for every vehicle. Considering the importance of the intersection controller in this method, the

MOST WIEDZY Downloaded from mostwiedzy.pl

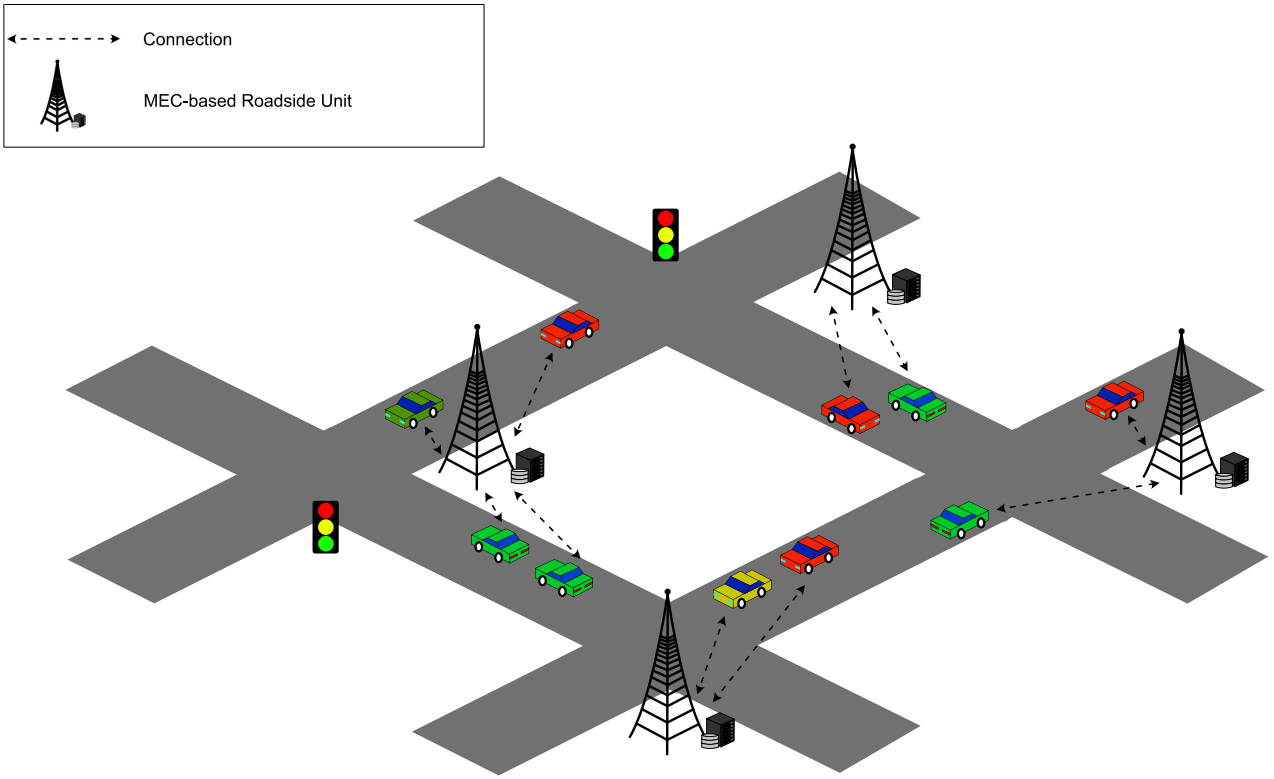

FIGURE 6. Example of the network operation for a MEC architecture.

MEC RSU is the only edge server able to process these tasks (i.e., aggregating the traffic data, processing the vehicles' requests and assigning the intersection entrance time) in real time.

In [50], Altché *et al.* introduced a supervised coordination system for semi-autonomous vehicles to minimize deadlocks and collisions. The system requires a controller (the supervisor) that monitors (by the cameras) the vehicle dynamics at crossroads, roundabouts and/or highway merging. A mixed-integer quadratic program is deployed to predict the collisions and deadlocks between vehicles. If a collision between the vehicles is foreseen, the controller signals a possibly dangerous situation to the respective vehicles. Concerning the proposal from [50], we consider the controllers to be MEC servers because the system requires constant intersection monitoring and reacting to the actual dynamics (by sending the suggestions to the vehicles) in real time.

In [51], Liao *et al.* present an accident detection system to notify drivers about the nearest accidents on the road. The system requires smartphones to take pictures of the road and send those pictures jointly with other data such as the speed and acceleration of the vehicle to the MEC server. The MEC server's role is to gather and process all the data from vehicles in the neighborhood to recognize accidents and broadcast the alerts to vehicles if an accident is detected. In order to recognize the accident, the MEC server executes an algorithm of deceleration detection and decides about a potential severity of the scenario based on the data about the speed and acceleration.

In [52], Asadi and Vahidi present a predictive cruise control scheme to reduce stops of vehicles at red lights. The method adjusts the speed of each vehicle based on the timing of changing the traffic lights. It is assumed that the vehicle's distance to the next traffic light is known in advance, and the traffic light controller broadcasts the information about timings of switching the lights. Therefore, vehicles can estimate their velocities by using a set of proposed rules and functions to meet the green phase only. We suggest to deploy the MEC RSU server as a supervisor of the intersection, as it possess the general knowledge of the traffic at the roads (i.e., the density of the vehicles, and their speed).

In [53], Huang *et al.* present the Cooperative Adaptive Driving (CAD) scheme to support driving in a platooning mode. The CAD selects the Platoon Leader (PL) with the lowest MAC number or the first vehicle in the platoon group. The platoon leader is responsible for connecting to the MEC servers to gather the data about the traffic. The MEC server is required to create a virtual machine instance for every Platoon Leader, so that this instance may be transferred between MEC servers (as the vehicle may communicate to a multitude of MEC servers during its journey). Based on information from MEC servers, the Platoon Leader decides on the maximum

MOST WIEDZY Downloaded from mostwiedzy.pl

TABLE 8. A summary of MEC properties for the state-of-the-art solutions.

Ref.	Importance of Low Delay	Legacy Vehicle Support	Network Type	Cloud Support	Use Case
[46]	High	+	WiFi	–	Traffic management at intersections
[47]	High	–	WiFi	–	Traffic management at intersections
[48]	High	–	WiFi	–	Traffic management at intersections
[49]	High	–	WiFi	–	Traffic management at intersections
[50]	High	+	WiFi	+	Traffic management at intersections
[51]	High	+	Cellular	–	Traffic and road awareness
[52]	Medium	–	WiFi	–	Start/stop reduction; green traffic management
[53]	High	–	WiFi	+	Platooning
[54]	Medium	–	Cellular	–	Platooning; green traffic management; Route planning for vehicles
[55]	Medium	–	WiFi	–	Eco-friendly driving

length and the speed of the platoon group. Then, the Platoon Leader updates the information (about the platoon length and speed) to all platoon members periodically or on-demand. All Platoon Members are required to adjust their speed based on the decision of the Platoon Leader and update their state (their current speed, location, physical inter-distance, etc.) to the Platoon Leader.

In [54], Chen *et al.* present a mobile edge computing-based platoon system to group the autonomous vehicles. The goal is to minimize fuel consumption by improving the creation process of platoon groups. The system applies a Q-learning algorithm to select the shortest path between starting and ending points of journeys. Vehicles are expected to form platoons according to their paths. In other words, vehicles with similar paths create a platoon group and move together. As the vehicle OBUs are resource-limited and reinforcement learning requires a large amount of data to be processed, the authors suggest offloading the Q-learning algorithm execution to the nearest edge server. For this case, the use of the MEC server is further analyzed in that paper in the computation offloading context.

In [55], Griggs *et al.* present two speed advisory schemes to minimize fuel consumption. It is required that the base station communicates with vehicles in a certain geographic area. Every vehicle is obliged to send information about its speed to the base station. In the first scheme, the base station computes the average speed of vehicles and sends them the speed advice. In the second one, the base station calculates the recommended speed of each vehicle, and sends the optimized speed advice to every vehicle. The additional goal of algorithms from [55] is to send speed recommendations in a way that vehicles do not know the recommended speed of the other vehicles. We consider the base stations in [55] as MEC servers because these units are required to coordinate the vehicles by sending them the instructions.

To summarize, the use of systems based on the MEC architecture adds many features that are fundamental when designing new road traffic management systems. Regarding Table 8, it is also possible to communicate with vehicles and smartphones so that future applications are capable of working with vehicles that are not equipped with modern communication devices. An important aspect of the MEC architecture is giving controllers an additional role, which is the ability to manage the traffic directly. This means that inspectors could send orders or prohibitions to vehicles. Applications that manage road traffic at intersections (see [46]–[50]), require low communication delay and the ability to influence the flow of vehicles directly. Therefore, the MEC architecture may play an important role in implementing safety-based applications. Additionally, edge servers can play an informative role because they often can have up-to-date information about the condition of roads (from other vehicles in the vicinity of the cloud infrastructure). It is an important feature for traffic light refresh services [52], [55], and traffic awareness applications [51]. Such applications require a connection to a trusted server that is able to present the up-to-date data to implement the services effectively. MEC systems can also support platooning systems [51], [52] to better group the vehicles that are far away from each other or if computationally intensive decision algorithms are needed.

VI. FOG COMPUTING

The fog computing architecture assumes communication between all edge devices for cooperation and joint implementation of services (as presented in Fig. 7). All devices that are able to connect to the network (such as smartphones, laptops, PCs, servers, cloud infrastructure) can be part of one coherent fog network. As fog networks accept all types of devices that are able to connect to the network, a large number of units are expected to cooperate in executing applications. Therefore, with a high probability, such devices will often be close to each other. This is an additional advantage because short distances reduce communication delays. In fog computing, all devices form one coherent platform with high total computing power. By combining fog and VANET technologies, vehicles equipped with OBUs can communicate with other units in a neighboring location (as presented in Fig. 8). Considering the problem of efficient support of vehicles at intersections, devices such as traffic cameras, pedestrian smartphones, and vehicle OBUs can exchange data for increasing the road awareness at intersections. Moreover, vehicles can manage the traffic on their own without the involvement of a centralized roadside infrastructure. Although the fog architecture requires solutions to problems such as orchestration or

MOST WIEDZY Downloaded from mostwiedzy.pl

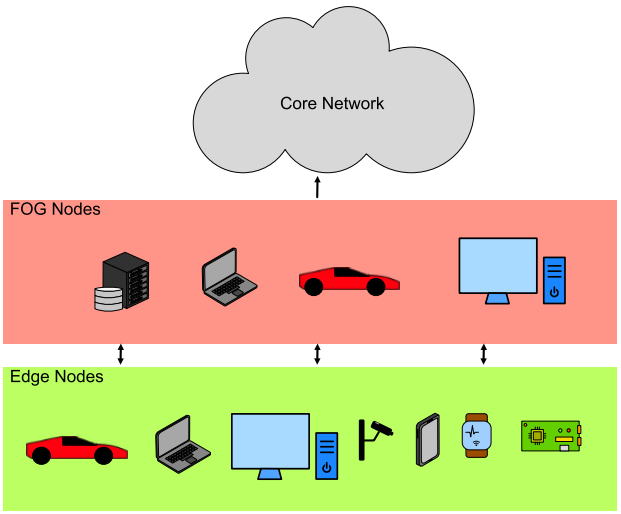

FIGURE 7. An example scheme of a fog network.

network security, fog networks are suitable for many services, including Collision Warning, Virtual Signalization at intersections, Start/Stop Reduction, Traffic and Road Awareness, Congestion Avoidance, Average Waiting Time Reduction, Platooning, Traffic Management at non-signalized intersections, Lane Change Assistance, Parking Management and Emergency Vehicles Prioritization.

In [56] and [57], Brennand *et al.* propose a fog-based system for traffic management to minimize traffic congestion. In [56], the fog system includes RSUs and other components (sensors and vehicles) which are in the communication range of RSUs. Each RSU has to collect the traffic data from vehicles and other RSUs to properly suggest vehicles travel paths in its communication range (also called the area of interest). Each RSU creates a graph where vertices denote crossroads and edges represent roads connected to the crossroads. The edges have weights representing the relation between the average speed and the maximum speed on the road (the closer the vehicle speed to the maximum value, the higher the respective weight). Such a scheme enables RSUs to detect congestion, analyze other routes by calculating k alternative paths to select the most appropriate one for each vehicle, as well as perform periodic re-routing of vehicles in the communication range. Additionally, during path analysis, the method uses Boltzmann probability distribution to avoid simultaneously selecting the same path for a group of vehicles. The approach from [57] extends the previous scheme by adding a new definition of the Area of Knowledge (AoK) which represents the area of responsibility of a single RSU. The AoK might be greater than the communication range of RSU. Therefore, the fog system is used to support the RSU by implementing the data exchange between vehicles (outside the RSU communication range). The vehicles should send the traffic information to an RSU as soon as any vehicle has entered the RSU coverage area.

In [58], Gomides presents a distributed multi-hop traffic management system for traffic congestion minimization. The method assumes that vehicles proactively disseminate data to each other about the traffic conditions and react based on the obtained information. Each vehicle stores the environmental data represented by a directed graph where vertices denote intersections of roads and edges represent road segments connected to the roads. The weights of edges refer to the congestion factor impacting the expected travel time. Each vehicle analyzes the data and selects the best route using the graph with data from other vehicles. If there is no information about a specific road, the vehicle uses the Reactive Data Dissemination protocol to discover missing information about the congestion factor of that road. The design of this method doesn't involve any infrastructure unit (i.e., RSU or cloud server). The service execution is located at the edge of the network and processing the service is done by edge nodes only (i.e., vehicles), which is a good example of a fog-based methodology.

In [59], Brandão *et al.* present a system that aims to reduce congestion on roads caused by accidents. The system includes three layers, namely (1) Accident Layer, (2) Traffic Management Layer, and (3) Smart City layer. In the Accident Layer, vehicles are required to monitor the embedded sensor's surroundings to detect accidents. If an accident is detected, a multi-hop transmission is started to inform vehicles in the vicinity about the event. In the Traffic Management Layer, the vehicles analyze the data from embedded sensors to predict the road conditions. If any anomaly is detected, a vehicle triggers the transmission to the vehicles located one hop away. The Smart City layer is a sum of the previous two layers and adds the communication with the infrastructure to give more comprehensive information about the situation on the road. This example shows the cooperation between various devices at the edge of the network (vehicles, embedded sensors). If more profound traffic knowledge is required, the vehicles can exchange the data with RSUs. Although RSUs are considered MEC system units, they can also be found in a fog system.

In [60], Makarem *et al.* propose a decentralized self-coordination of autonomous vehicles at intersections. The goal is to implement cooperation between vehicles to avoid accidents and ensure smooth driving without using any additional infrastructure unit (i.e., an RSU). Vehicles disseminate data to other vehicles about their position, speed, path and expected arrival time at the intersection. The interchanged data enables the vehicles to calculate their speed by deploying a dedicated decentralized navigation function. Moreover, the authors introduce the functions to (1) calculate the acceleration or deceleration of a vehicle to avoid collisions and (2) weigh the beta-function to assign higher priority to heavier vehicles.

In [61], Wu *et al.* introduce an intelligent control system for traffic lights aimed at improving the coordination of vehicles at intersections. Authors consider fog systems

MOST WIEDZY Downloaded from mostwiedzy.pl

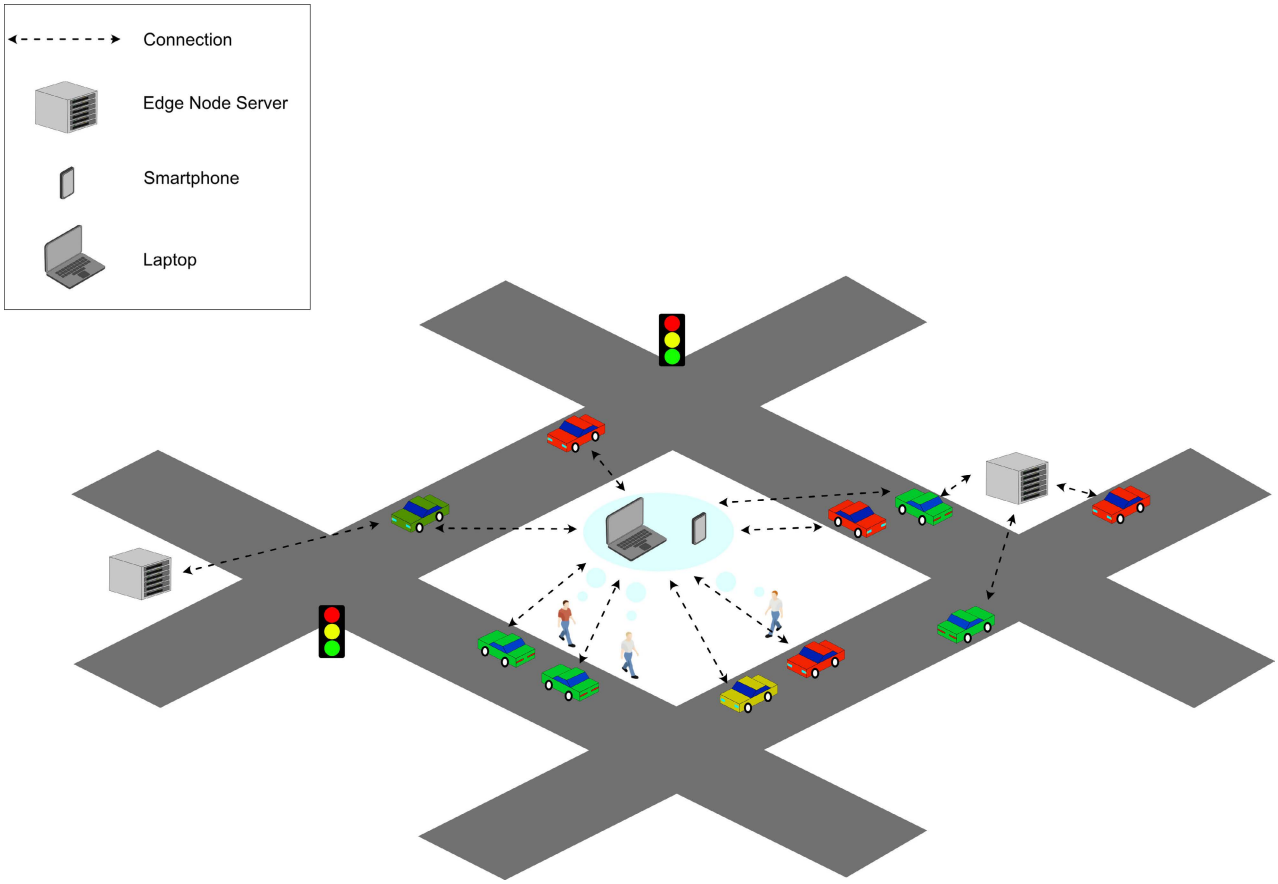

FIGURE 8. Example diagram of a vehicular network operation according to the fog architecture.

at intersections that include traffic light sensors monitoring the traffic volume on roads and fog servers (referred to as Edge Node Servers in Fig. 8) which analyze the data obtained from sensors and manipulate the traffic lights. The fog server is the main controller that manages traffic lights, calculates the coordination function that requires information about the number of vehicles on roads, verifies if it is influencing the adjacent intersections and sets priorities for the roads. It is worth noting that MEC RSUs can also offer the same functionalities. Therefore, these devices may also be used to run such a service.

In [62], Belyaev *et al.* present the overtaking assistance system based on wireless video streaming. The proposed architecture assumes that the overtaken vehicles (i.e., trucks) transmit the front camera views to the vehicles behind that are willing to start the overtaking maneuvers. This way, the vehicles that obtain the frames can provide the drivers with the actual frontal view of the vehicle in front of them. Such support should increase road safety by providing the drivers with relevant information to decide if the road conditions are acceptable to start the overtaking maneuver. Further, the authors consider image distortions and data transmission collisions between two vehicles (the transmitting vehicle and the gathering one) moving in opposite directions. The algorithm for selecting the best transmission power is introduced to solve that problem. As service provisioning doesn't require any infrastructure unit, this methodology can be considered a fog-based system.

In [63], Tsugawa *et al.* propose a method of platooning the automated trucks. It is assumed that trucks are controlled independently and cooperate with each other to drive in a group. The method requires vision units, a steering actuator and a communication system. Every 20 ms, trucks exchange data about their speed, acceleration, braking signal, platoon ID and truck position. Then, the shared information between trucks is used in the Longitudinal Control system that deploys the Lyupanov stability method to maintain the speed of trucks and keep the desired gap between the trucks. As this solution doesn't require any infrastructure, it can be considered as a fog-based system. Comparing this method with the one from [53] (which is a MEC-based platooning methodology), this proposition is also suitable for rural roads because of its infrastructureless communication.

In [64], Huang *et al.* propose a distributed cooperative control system so that each autonomous vehicle in a platooning group can decide about its maneuvers by itself to enhance the cruise flow of the whole platooning group. Vehicles in a platoon are required to broadcast the information about

MOST WIEDZY Downloaded from mostwiedzy.pl

TABLE 9. A summary of fog properties for the state-of-the-art solutions.

Ref.	Importance of Low Delay	Communication Design	Computational Complexity	Scenario	Use Case
[56]	Medium	V2I	Medium/Low	Urban	Congestion avoidance; average waiting time reduction
[57]	Medium	V2I	Medium/Low	Urban	Congestion avoidance; average waiting time reduction
[58]	Medium	V2V	Medium	Urban	Congestion avoidance; average waiting time reduction
[59]	Low	V2I/V2V	–	Urban	Congestion avoidance; global traffic management
[60]	High	V2V	Medium	Urban	Traffic management at intersections
[61]	High	V2I	Medium	Urban	Traffic management at intersections
[62]	Critical/High	V2I	Low	Rural	Traffic and road awareness
[63]	High	V2V	Low	Highway	Platooning
[64]	Medium/High	V2V	Medium/High	Highway	Platooning

their motion states (the time and the location of maneuver) to other vehicles in the group. No infrastructure at the edge of the network is needed to support the transmission. Each vehicle implements the predictive control model with an artificial potential field algorithm to select the best path and optimize the motion control based on information from the neighborhood (the vehicle's sensors and the data transmitted from other vehicles). We consider this method a fog-based scheme because it is a distributed control system without any infrastructure involvement.

Similar to MEC and cloudlet architectures, fog architecture offers low communication latency. The difference is that in fog systems vehicles can cooperate frequently to provide a specific service or even manage traffic. The platooning service is most often implemented in a decentralized manner, i.e., only vehicles are obliged to decide on the movement and the behavior of the entire platooning group. Therefore, as outlined in [63] and [64], fog architecture is ideal for delivering services of this type, as it inherently implies strong cooperation between the end nodes (in this case, the vehicles). Furthermore, fog architecture is very flexible because it may utilize similar solutions as in other architectures, e.g., using the cloud infrastructure for complex computations or deploying RSUs as decision-making units in road traffic management. For instance, the schemes from [56] and [57] use RSUs to gather traffic situation information, analyze it and select the best paths to avoid congestion. Vehicles can further disseminate such information to other vehicles. In [58] and [59], vehicles, in turn, collect and send information about the traffic situation so that each vehicle can independently choose the path to avoid congested roads.

The end nodes in a fog network can play a variety of roles. On the one hand, the end-nodes can gather information about the traffic and send this data to the central server (i.e., RSU) to analyze the data and provide guidance to the vehicles on how to behave on the road [61]. On the other hand, it is possible to implement a system immune to the lack of a central server in such a way that the vehicles exchange information with each other and, at the same time, the vehicle or the driver is directly responsible for traffic management on the road [60], [62]. As summarized in Table 9, we highly recommend fog systems for rural roads as they are able to provide low communication delay even on roads outside the city. It is a universal architecture for many use cases without the requirement of infrastructure deployment.

VII. DISCUSSION

In this paper, we overviewed the available methods of supporting the drivers and managing the traffic on roads. Each method has been assigned by us to one of four architectures: VCC, cloudlet, MEC, or fog. In this section, we take a closer look at the pros and cons of the considered architectures to summarize their usefulness in VANETs and specifically for road traffic management.

A. CLOUD

The cloud approach highlighted in Section III is the only one that offers enormous storage capacity and computing power. As a result, only this system is capable of running applications related to global traffic management. Therefore, a cloud-based system is often considered in the VANET environment.

However, the cloud architecture is the only one most limited by the distance between the server and the end-user (in this case, the vehicle). Such a long distance impacts the communication delay, so it often cannot meet the time requirements (mentioned in Section II) for the implementation of traffic management services. Therefore, the cloud system should be considered at most a support for the ITS.

Building a new cloud infrastructure is very expensive. Therefore, we assume that cities would find it easier to rent cloud servers from corporations that have already deployed such infrastructure. We are, however, concerned about a potentially high dependence of cities on a cloud infrastructure provider who could monopolize the service prices when drivers become dependent on a road traffic management service or autonomous vehicles that would strongly cooperate with the ITS.

B. CLOUDLET

The main goal of the cloudlet architecture discussed in Section IV is to place servers close to end-users. In this way, devices in the vicinity (e.g., telephones, OBUs, tablets) are expected to easily use the computing power and memory capacity of the available servers. The cloudlet approach enables to push the execution of the application to the server in the vicinity, but these servers cannot manage the traffic

MOST WIEDZY Downloaded from mostwiedzy.pl

on the roads. Since ITS applications and services are not designed to be supported by the cloudlet system only, in this survey, we highlighted the methods that should be executed on the vehicle (OBU) side, but could be offloaded to the servers in the proximity. In our opinion, such a modification would improve the overall application performance to meet all the ITS requirements. However, some important limitations of the use of cloudlets should be noted.

Firstly, connectivity with cloudlets would have to occur via a WiFi network. Compared to the cellular network, WiFi has a short range. Considering the short range of a WiFi network and high mobility of vehicles, it is required to implement an efficient session transfer method between cloudlets.

In an urban environment, cloudlet servers would have a permanent source of energy. The low cost of a single server allows setting up many servers in many different places. However, it may be not easy to provide a permanent source of energy in rural areas. Alternatively, it is possible to equip the servers with renewable energy sources, but it would dramatically increase the production costs of a single unit. Additionally, in the non-urban area, the speed of vehicles is much higher. At the same time, bearing in mind the short range of any server, cloudlet servers would have to transfer sessions between themselves often during a single journey of a vehicle.

Cloudlet servers would increase the quality of infotainment services, i.e., gaming, video streaming, and advertisements. In the case of road traffic management, the most significant help would be the provision of image processing services. We believe that the cloudlet architecture cannot form a standalone ITS supporting the traffic management service. However, it would be an excellent element supporting the ITS infrastructure, e.g., for a MEC RSU when such a unit is heavily loaded.

C. MEC

The MEC system discussed in Section V implements the features proposed not only to support drivers but also to manage the road traffic with an authorization to send the orders to the drivers/vehicles. In such a case, communication between an RSU and vehicles can occur via a cellular network, significantly increasing the range. Furthermore, the cellular network enables monetizing the communication services between the RSU and the vehicle, making the maintenance of the entire MEC system easier.

However, we believe that the MEC system itself would not provide an effective traffic management service. We are concerned that if thousands of requests were sent to a single RSU, it would disrupt the unit's flawless operation. A natural solution would be to combine the MEC system with a cloud system (or a cloudlet system) to unlock the possibility of offloading the tasks to other servers. In our opinion, all respective use cases and requirements could be met in a system that combines the MEC architecture and the cloud architecture.

Also, we can see legal difficulties in introducing such a system. For example, in the scenario where a given RSU requests the driver to make a specific maneuver, if the driver fails to do so, and in consequence, an accident happens – who will be then to blame. The driver's explanation could be related, e.g., to not having seen the message.

How the problem of legacy vehicles would be solved if there is no modern OBU, but the driver uses a mobile phone only? Perhaps it is worthwhile to assign the information disseminator roles to RSUs in the first stages of introducing the MEC system. Additionally, only in the areas where only autonomous vehicles are allowed, RSUs would have the role of road managers.

D. FOG

In the fog architecture analyzed in Section VI, all edge nodes can cooperate in executing applications. Additionally, idle nodes can offer their computing power to other nodes in the vicinity. In an ITS, vehicles can cooperate if the infrastructure is unavailable, idle vehicles can be combined into local clouds to deliver services to nearby active vehicles. An interesting area to be explored is the monetization of fog device services, i.e., how a terminal device can benefit from providing services.

The execution of services constrained by V2V communications is an essential feature of fog architecture. The server infrastructure may not be available in the out-of-town area, or the connection could be burdened with additional communication delays. Therefore, the implementation of a fog system on rural roads seems to be inevitable.

We anticipate that the extension of the fog architecture by other end nodes (e.g., mobile phones, tablets, road sensors) may create difficulties in data aggregation and device orchestration. When many different devices are connected, this will increase the risk that the exchanged data might become non-understandable because of different standards. It may be not possible to standardize all devices. Such devices shall comply with all possible standards for providing services in the fog network while supporting infotainment, video streaming, or even medical applications.

We believe that it is necessary to properly determine specific devices that can be present in the fog architecture of a vehicular network and the implementation of road traffic management services. For example, cell phones have very little computing power. Even if a cloud with dozens of mobile phones is created, it is still challenging to orchestrate all the phones, and the energy in their batteries is limited. The road traffic management aspect is also burdened with high road safety requirements. Therefore, it is necessary to legally regulate what specific devices could be connected to the fog network (what parameters should characterize the devices, e.g., computing power, storage and energy resources).

VIII. SUMMARY

In this survey, we have reviewed the applications of VANET network architectures dedicated to road traffic

MOST WIEDZY Downloaded from mostwiedzy.pl

management and their potential implementation based on clouds, cloudlets, MEC, and fog systems. We looked at the requirements provided by institutions such as ETSI and C2C-CC. The main problem that future communication networks are likely to face in supporting transport systems is ensuring short delays during data transmission. In particular, applications that affect road safety (e.g., traffic management at intersections, traffic and road awareness, congestion avoidance, lane change assistance, emergency vehicles prioritization) require short transmission delays to react quickly to dynamic changes on roads. The considered network architectures (such as cloudlet, MEC, and fog) typically push application execution to the edge of the network to reduce this delay.

With many vehicles in cities, congestion on roads, high fuel consumption and air pollution are common problems. Considering this aspect, dedicated traffic management controllers seem to be the best solution. Such controllers should be responsible for a specific area (e.g., at intersections in a designated area with effective communication coverage) to be able to handle all vehicle requests. Therefore, we recommend the MEC architecture for traffic management applications in urban areas. Typically, the MEC architecture assumes the use of RSUs, which are dedicated units to support the ITS and have the role of the main manager.

A different scenario characterizes road traffic in a non-urban environment. The use of RSUs in all such areas would be an expensive solution. In addition, RSUs would still have to update their information on distant non-urban areas to be able to implement, e.g., vehicle re-routing or traffic awareness. Therefore, in this case, we are convinced that fog architecture is necessary, as it provides collaboration between edge nodes (in this case, vehicles) by default.

The cloudlet system involves using smaller servers to support the edge nodes in the network. This is a big advantage, especially when considering electric vehicles. However, it cannot be the main traffic management system but only serve as a support system for vehicles and ITSs. An interesting case is the use of a cloud infrastructure because a long distance between vehicles and cloud servers would cause high communication delays. This aspect should be an eliminating factor in the context of choosing a leading traffic management system.

Nevertheless, the cloud infrastructure offers enormous computing power, which is crucial when implementing applications with high computational complexity (e.g., predicting the volume of road traffic in a given area, taking hundreds of parameters into account). For this reason, we believe that future ITSs will use the cloud infrastructure in the event of system overloading. It is important to find a proper balance between performing computations close to users versus proceeding with the same operations in the cloud.

In Section IV related to cloudlets, we took into account the methods that would be executed at the vehicle side (in OBUs). Currently, the literature only very rarely takes into account the use of cloudlets in the context of supporting the drivers or road traffic management. Therefore, in Section IV, we provided our recommendations to redesign the available methods in such a way that their implementation could be pushed into the cloudlet. In addition, we took into account the methods that provide services based on one architecture. Consideration of hybrid architectures that combine variety of architectures (e.g., fog, cloud and MEC based architectures, etc.) remains an open area for further research.

REFERENCES

[1] F.-Y. Wang, D. Zeng, and L. Yang, "Smart cars on smart roads: An IEEE intelligent transportation systems society update," *IEEE Pervasive Comput.*, vol. 5, no. 4, pp. 68–69, Oct. 2006.

[2] L. Qi, "Research on intelligent transportation system technologies and applications," in *Proc. Workshop Power Electron. Intell. Transp. Syst.*, Aug. 2008, pp. 529–531.

[3] *Traffic England*. Accessed: Dec. 11, 2019. [Online]. Available: http://www.trafficengland.com

[4] *ITS Knowledge Resources*. Accessed: Dec. 11, 2019. [Online]. Available: https://www.itsknowledgeresources.its.dot.gov/ITS/benecost.nsf/ID/D9946096D9C68AFF85257C580056A448?OpenDocument&Query=Home

[5] *Intelligent Transportation System Joint Program*. Accessed: Dec. 11, 2019. [Online]. Available: https://www.its.dot.gov/research_archives/aeris/pdf/UCR_eco-approach_final2.pdf

[6] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[7] T. Lin, H. Rivano, and F. Le Mouël, "A survey of smart parking solutions," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3229–3253, Dec. 2017.

[8] A. B. De Souza, P. A. L. Rego, T. Carneiro, J. D. C. Rodrigues, P. P. R. Filho, J. N. De Souza, V. Chamola, V. H. C. De Albuquerque, and B. Sikdar, "Computation offloading for vehicular environments: A survey," *IEEE Access*, vol. 8, pp. 198214–198243, 2020.

[9] A. M. de Souza, C. A. Brennand, R. S. Yokoyama, E. A. Donato, E. R. Madeira, and L. A. Villas, "Traffic management systems: A classification, review, challenges, and future perspectives," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 4, pp. 1–14, 2017.

[10] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service*, document ETSI EN302 637-3, V1.3.1, Apr. 2019.

[11] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, document ETSI EN302 637-2, V1.4.1, Jan. 2019.

[12] *Intelligent Transport Systems (ITS); Vehicular Communications; C2C-CC Demonstrator 2008; Use Cases and Technical Specifications*, document ETSI TR102 698, V1.1.2, Jul. 2010.

[13] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2*, document ETSI TR103 562, V2.1.1, Dec. 2019.

[14] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities Layer Protocols and Communication Requirements for Infrastructure Services*, document ETSI TS103 301, V1.3.1, Feb. 2020.

[15] *Intelligent Transport Systems 'Cooperative ITS' Using V2I and I2V Communications for Applications Related to Signalized Intersections*, ISO, Geneva, Switzerland, Mar. 2017.

[16] *C2C CC, Overview of the C2C-CC System, Version 1.1*, CAR 2 CAR Commun. Consortium, Braunschweig, Germany, Aug. 2007.

[17] *C2C CC, Position Paper on Road Safety and Road Efficiency Spectrum Needs in the 5.9 GHz for C-ITS and Cooperative Automated Driving, Version 2.0*, CAR 2 CAR Commun. Consortium, Braunschweig, Germany, Feb. 2020.

[18] *Intelligent Transport Systems (ITS); V2X Applications; Part 1: Road Hazard Signalling (RHS) Application Requirements Specification, V1.1.1*, document ETSI TS101 539-1, Aug. 2013.

[19] *Intelligent Transport Systems (ITS); V2X Applications; Part 2: Intersection Collision Risk Warning (ICRW) Application Requirements Specification*, document ETSI TS101 539-2, V1.1.1, Jun. 2018.

MOST WIEDZY Downloaded from mostwiedzy.pl

[20] *Intelligent Transport Systems (ITS); V2X Applications; Part 3: Longitudinal Collision Risk Warning (LCRW) Application Requirements Specification*, document ETSI TS101 539-3, V1.1.1, Nov. 2013.
[21] *Intelligent Transport System (ITS); Vulnerable Road Users (VRU) Awareness; Part 2: Functional Architecture and Requirements Definition; Release 2*, document ETSI TS103 300-2, V2.1.1, May 2020.
[22] X. T. R. Kong, S. X. Xu, M. Cheng, and G. Q. Huang, "IoT-enabled parking space sharing and allocation mechanisms," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1654–1664, Oct. 2018.
[23] J. Lin, S.-Y. Chen, C.-Y. Chang, and G. Chen, "SPA: Smart parking algorithm based on driver behavior and parking traffic predictions," *IEEE Access*, vol. 7, pp. 34275–34288, 2019.
[24] M. Noreikis, P. Butkus, and J. K. Nurminen, "In-vehicle application for multimodal route planning and analysis," in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2014, pp. 350–355.
[25] O. Khalid, M. U. S. Khan, Y. Huang, S. U. Khan, and A. Zomaya, "EvacSys: A cloud-based service for emergency evacuation," *IEEE Cloud Comput.*, vol. 3, no. 1, pp. 60–68, Jan. 2016.
[26] Z. Li, I. Kolmanovsky, E. Atkins, J. Lu, D. Filev, and J. Michelini, "Cloud aided safety-based route planning," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2014, pp. 2495–2500.
[27] M. P. Fanti, A. M. Mangini, G. Rotunno, G. Fiume, A. Favenza, and M. Gaetani, "A cloud computing architecture for eco route planning of heavy duty vehicles," in *Proc. IEEE 14th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2018, pp. 730–735.
[28] K. Boriboonsomsin, M. J. Barth, W. Zhu, and A. Vu, "Eco-routing navigation system based on multisource historical and real-time traffic information," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1694–1704, Dec. 2012.
[29] Y. Jin, W. Xu, P. Wang, and J. Wang, "Route planning of rescue vehicles in the process of dynamic change of traffic volume under emergency conditions," in *Proc. 3rd Int. Symp. Auton. Syst. (ISAS)*, May 2019, pp. 125–130.
[30] D. L. Guidoni, G. Maia, F. S. H. Souza, L. A. Villas, and A. A. F. Loureiro, "Vehicular traffic management based on traffic engineering for vehicular ad hoc networks," *IEEE Access*, vol. 8, pp. 45167–45183, 2020.
[31] M. Shengdong, X. Zhengxian, and T. Yixiang, "Intelligent traffic control system based on cloud computing and big data mining," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6583–6592, Dec. 2019.
[32] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.
[33] M. Satyanarayanan, "A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets," *GetMobile, Mobile Comput. Commun.*, vol. 18, no. 4, pp. 19–23, 2015.
[34] A. Keivani, F. Ghayoor, and J.-R. Tapamo, "A vision-based driver assistance system using collaborative edge computing," in *Proc. Global Wireless Summit (GWS)*, Oct. 2017, pp. 160–164.
[35] W. Zeng, T. Miwa, and T. Morikawa, "Eco-routing problem considering fuel consumption and probabilistic travel time budget," *Transp. Res. D, Transp. Environ.*, vol. 78, Jan. 2020, Art. no. 102219.
[36] T. Chakraborty and S. K. Datta, "SIIoT: A shortest path estimation and obstacle avoidance system for autonomous cars," in *Proc. Global Internet Things Summit (GIoTS)*, 2018, pp. 1–6.
[37] A. Shashua, Y. Gdalyahu, and G. Hayun, "Pedestrian detection for driving assistance systems: Single-frame classification and system level performance," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 1–6.
[38] Q. Liu, J. Zhuang, and J. Ma, "Robust and fast pedestrian detection method for far-infrared automotive driving assistance systems," *Infr. Phys. Technol.*, vol. 60, pp. 288–299, Sep. 2013.
[39] N. Arbabzadeh and M. Jafari, "A data-driven approach for driving safety risk prediction using driver behavior and roadway information data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 446–460, Feb. 2018.
[40] W. Song, Y. Yang, M. Fu, F. Qiu, and M. Wang, "Real-time obstacles detection and status classification for collision warning in a vehicle active safety system," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 758–773, Mar. 2018.
[41] X. Duan, H. Jiang, D. Tian, T. Zou, J. Zhou, and Y. Cao, "V2I based environment perception for autonomous vehicles at intersections," *China Commun.*, vol. 18, no. 7, pp. 1–12, Jul. 2021.
[42] Y. Hou, P. Edara, and C. Sun, "Modeling mandatory lane changing using Bayes classifier and decision trees," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 647–655, Apr. 2014.
[43] Q. Sun, H. Zhang, Z. Li, C. Wang, and K. Du, "ADAS acceptability improvement based on self-learning of individual driving characteristics: A case study of lane change warning system," *IEEE Access*, vol. 7, pp. 81370–81381, 2019.
[44] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Integrated urban traffic control for the reduction of travel delays and emissions," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1609–1619, Dec. 2013.
[45] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing a key technology towards 5G," Eur. Telecommun. Standards Inst., Sophia Antipolis, France, Tech. Rep., Sep. 2015.
[46] L. C. Bento, R. Parafita, S. Santos, and U. Nunes, "Intelligent traffic management at intersections: Legacy mode for vehicles not equipped with V2V and V2I communications," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 726–731.
[47] M.-K. Shi, H. Jiang, and S.-H. Li, "An intelligent traffic-flow-based real-time vehicles scheduling algorithm at intersection," in *Proc. 14th Int. Conf. Control, Automat., Robot. Vis. (ICARCV)*, Nov. 2016, pp. 1–5.
[48] E. Steinmetzl, R. Hult, Z. Zou, R. Emardson, F. Brännström, P. Falcone, and H. Wymeersch, "Collision-aware communication for intersection management of automated vehicles," *IEEE Access*, vol. 6, pp. 77359–77371, 2018.
[49] C. Wuthishuwong, A. Traechtler, and T. Bruns, "Safe trajectory planning for autonomous intersection management by using vehicle to infrastructure communication," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, pp. 1–12, Dec. 2015.
[50] F. Altché, X. Qian, and A. de La Fortelle, "An algorithm for supervised driving of cooperative semi-autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3527–3539, Dec. 2017.
[51] C. Liao, G. Shou, Y. Liu, Y. Hu, and Z. Guo, "Intelligent traffic accident detection system based on mobile edge computing," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2017, pp. 2110–2115.
[52] B. Asadi and A. Vahidi, "Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 3, pp. 707–714, May 2011.
[53] R.-H. Huang, B.-J. Chang, Y.-L. Tsai, and Y.-H. Liang, "Mobile edge computing-based vehicular cloud of cooperative adaptive driving for platooning autonomous self driving," in *Proc. IEEE 7th Int. Symp. Cloud Service Comput. (SC)*, Nov. 2017, pp. 32–39.
[54] C. Chen, J. Jiang, N. Lv, and S. Li, "An intelligent path planning scheme of autonomous vehicles platoon using deep reinforcement learning on network edge," *IEEE Access*, vol. 8, pp. 99059–99069, 2020.
[55] W. Griggs, G. Russo, and R. Shorten, "Leader and leaderless multilayer consensus with state obfuscation: An application to distributed speed advisory systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 711–721, Mar. 2018.
[56] C. A. R. L. Brennand, F. D. da Cunha, G. Maia, E. Cerqueira, A. A. F. Loureiro, and L. A. Villas, "FOX: A traffic management system of computer-based vehicles FOG," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 982–987.
[57] C. A. R. L. Brennand, A. Boukerche, R. Meneguette, and L. A. Villas, "A novel urban traffic management mechanism based on FOG," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2017, pp. 377–382.
[58] T. S. Gomides, R. E. De Grande, F. S. H. Souza, and D. L. Guidoni, "RIDER: Proactive and reactive approach for urban traffic management in vehicular networks," in *Proc. 16th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2020, pp. 51–58.
[59] Y. V. Brandao, L. M. De Souza, T. S. Gomides, R. E. De Grande, F. S. H. Souza, and D. L. Guidoni, "A multi-layer and Vanet-based approach to improve accident management in smart cities," in *Proc. 16th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2020, pp. 165–172.
[60] L. Makarem and D. Gillet, "Fluent coordination of autonomous vehicles at intersections," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2012, pp. 2557–2562.
[61] Q. Wu, F. He, and X. Fan, "The intelligent control system of traffic light based on fog computing," *Chin. J. Electron.*, vol. 27, no. 6, pp. 1265–1270, Nov. 2018.
[62] E. Belyaev, A. Vinel, K. Egiazarian, and Y. Koucheryavy, "Power control in see-through overtaking assistance system," *IEEE Commun. Lett.*, vol. 17, no. 3, pp. 612–615, Mar. 2013.
[63] S. Tsugawa, S. Kato, and K. Aoki, "An automated truck platoon for energy saving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 4109–4114.
[64] Z. Huang, D. Chu, C. Wu, and Y. He, "Path planning and cooperative control for automated vehicle platoon using hybrid automata," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 959–974, Mar. 2019.

MOST WIEDZY Downloaded from mostwiedzy.pl

KAROL JURCZENIA received the B.E. and master's degrees from the Gdańsk University of Technology, Gdańsk, Poland, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Communications, Faculty of Electronics, Telecommunications and Informatics.

He has been a Software Engineer at the Data Center Group, Intel Corporation. In his work, he deals with telemetry and energy management on servers. His main research interests include application of cloud-, cloudlet-, MEC-, and fog-based infrastructures to support vehicles, and intelligent transportation systems.

JACEK RAK (Senior Member, IEEE) received the M.Sc., Ph.D., and D.Sc. (Habilitation) degrees from the Gdańsk University of Technology, Gdańsk, Poland, in 2003, 2009, and 2016, respectively. He is currently an Associate Professor and the Head of the Department of Computer Communications, Gdańsk University of Technology. From 2016 to 2020, he was leading the COST CA15127 Action "Resilient Communication Services Protecting End-User Applications from Disaster-Based Failures" (RECODIS) involving over 170 members from 31 countries. He has authored over 100 publications, including the monograph *Resilient Routing in Communication Networks* (Springer, 2015) and co-edited the book *Guide to Disaster-Resilient Communication Networks* (Springer, 2020). His main research interests include resilience of communication networks and networked systems.

Prof. Rak has served as a TPC member of numerous conferences and journals. He is a member of the Editorial Board of *Optical Switching and Networking* (Elsevier) and the Founder of the workshop on Resilient Networks Design and Modeling (RNDM). Recently, he has been the General Chair of ITS-T'17 and MMM-ACNS'17, the General Co-Chair of NETWORKS'16, and the TPC Co-Chair of ONDM'17, ONDM'22 and IFIP Networking'19.

MOST WIEDZY Downloaded from mostwiedzy.pl