IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 8 June 2024, accepted 23 June 2024, date of publication 28 June 2024, date of current version 8 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3420720

RESEARCH ARTICLE

Blockchain-Empowered Resource Trading for Optimizing Bandwidth Reservation in Vehicular Networks

ABDULLAH AL-KHATIB[1], HALEEMAH HADI[2], HOLGER TIMINGER[1], (Senior Member, IEEE), AND KLAUS MOESSNER[3], (Senior Member, IEEE)

[1]Institute for Data and Process Science, Landshut University of Applied Sciences, 84036 Landshut, Germany
[2]Faculty of Computer Science, Technical University of Applied Sciences Augsburg, 86161 Augsburg, Germany
[3]Professorship for Communications Engineering, Technical University Chemnitz, 09111 Chemnitz, Germany

Corresponding author: Abdullah Al-Khatib (Abdullah.Al-Khatib@haw-landshut.de)

ABSTRACT Resource trading between vehicles, which involves the buying and selling of computing and bandwidth resources, is a promising approach for cost-effectively provisioning services in safety-critical applications such as autonomous driving. These applications require a guarantee and the timely receipt of resources through efficient advance reservations. However, due to uncertainties in future reservation duration and resource costs, vehicles exhibit two distinct patterns: some may have reserved insufficient resources and need to purchase more (acting as vehicle requesters), while others may have overbooked resources and need to sell (acting as vehicle providers). In this paper, we formulate the resource trading problem from both the requester and provider perspectives and propose a resource trading architecture to optimize bandwidth reservation. It utilizes blockchain smart contracts for secure and efficient resource exchange within a mobile network operator (MNO) environment. Two algorithms are introduced: a provider selection algorithm to enhance system efficiency by selecting cost-effective providers, and a decision-making algorithm to assist providers in choosing between selling overbooked bandwidth or canceling it. Through simulations, the results show that these algorithms lead to significant cost reductions for requesters and profit gains for providers, up to 59% and 19%, respectively, compared to reservation schemes without resource trading in such a dynamic environment.

INDEX TERMS Networked vehicular application, time-sensitive networking, network reservation, blockchain, smart contract.

I. INTRODUCTION

The emergence of fifth generation (5G) and beyond represents a significant advance in vehicle technology, particularly in collaborative, assisted, and connected autonomous driving. While these advances are promising, they also present critical challenges, especially in ensuring ultra-reliable and ultra-low latency communications. To enable seamless access to fog/edge computing resources for such safety-critical applications, sufficient network resources (i.e., bandwidth) are required. Meeting these strict requirements is essential for ensuring the safety and effectiveness of these applications [1].

In recent years, significant efforts have been made by academia and industry to develop efficient computation offloading solutions for in-vehicle networks [2], [3]. A prominent challenge is the execution of computation-intensive tasks within strict time constraints, often with a maximum latency threshold of 100ms∼1s [4]. Reservation approaches, provide guaranteed and timely access to scarce bandwidth resources. The conventional approach, network-side reservation [5], [6], involves the mobile network operator (MNO) allocating bandwidth for various quality of service (QoS) classes. However, this approach provides probabilistic rather than

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

deterministic guarantees for individual vehicles accessing network bandwidth.

A more efficient alternative is the vehicle-side reservation approach [7], [8], [9], [10], [11]. This approach prioritizes the resource consumer perspective and focuses on minimizing user expenditure from an economic standpoint. It allows resource consumers (i.e., vehicles) to reserve necessary resources based on their specific requirements through individual reservation requests. These requests aim to meet the real-time processing needs of time-sensitive vehicular applications. After receiving these requests, an MNO allocates the resources accordingly. However, these reservation schemes face challenges, including uncertainties from unpredictable mobility that affect reservation duration, leading to scenarios like underutilization and overutilization of dynamic resources. This research develops a resource trading and exchange mechanism within the vehicle network. This mechanism empowers both requesting and providing vehicles to adjust and optimize their bandwidth reservations cost-effectively along their driving paths. However, introducing resource trading in such networks brings about new challenges, such as unexpected latency in decision-making. For example, requesters have to spend excessive time analyzing current providers. Meanwhile, providers need to make quick decisions about selling or canceling within the coverage areas of access points (e.g., base stations or roadside units). These decisions could directly impact the actual usable time for practical bandwidth resource delivery during trading.

The resource trading approach has gained significant attention in academic studies [12], [13]. However, many of these solutions overlook dynamic pricing, which introduces an additional challenge to trading due to its impact on market conditions such as resource supply and demand. Recently, dynamic pricing has emerged as a promising solution that is receiving attention from both academia and industry in the context of resource management in edge computing [14], [15]. Its adoption is leading MNOs to revise their purchasing programs for increased revenue [16]. Real-world companies, for example, AWS, offer spot pricing for virtual instances, and MTN, China Telecom, and Uninor, have offered time-dependent pricing for bandwidth resources, Where the price dynamically changes after one hour or even one minute to achieve a balance between supply and demand [17], [18]. Dynamic pricing is a key topic in revenue management, with successful applications in industries like cloud computing [16], smart grid [19], and spectrum trading [20].

Moreover, the absence of security mechanisms results in unreliable resource requirements from providers, posing challenges in guaranteeing resources [21]. Additionally, there is the risk of fraud, where providers may sell resources without receiving payment, which further complicates the reliability of resource trading. Motivated by these issues, addressing how to trade and establish trust in resources with dynamic pricing within the context of bandwidth reservation scenarios is still an interesting issue.

Blockchain has attracted tremendous attention to the research community and is being extensively applied in diverse fields, including healthcare, the internet of things (IoT), and energy trading, among numerous others, due to its salient features, including decentralization, nontampering, security, and anonymity [22]. Smart contracts, a concept originally introduced by Nick Szabo, provide a decentralized mechanism for facilitating distributed operations, thereby removing the need for trusted middlemen [21]. These contracts, essentially segments of computer code that are stored within the blockchain network, establish conditions and rules mutually agreed upon by all parties involved in the contract [22]. Moreover, a smart contract offers advantages in the context of resource trading for bandwidth reservation. It assists vehicle requesters in selecting and matching with suitable resource providers. Additionally, it ensures transparency, automation, and trust. The key contributions of this article are summarized as follows:

- We formulate a mathematical model for resource trading in scenarios of bandwidth reservation. This model captures the perspectives of both requesters and providers, enabling cost minimization for requesters and profit maximization for providers, while ensuring efficient trading decisions in a dynamic environment.
- We utilize blockchain smart contracts to implement secure and efficient resource trading for bandwidth reservations. In addition, we aim to create a trusted environment for participating vehicles, ultimately reducing the costs associated with optimizing reservations.
- Two algorithms are proposed for resource trading. The provider selection algorithm is integrated into the smart contract to match and efficiently select the most cost-effective providers. The second algorithm involves decision making for vehicle providers, enabling them to choose between selling or canceling surplus bandwidth based on factors such as potential profit, cancellation fees, and the probability of finding requesters.
- We evaluate the efficiency and cost-effectiveness of our proposed blockchain-based resource trading for bandwidth reservation using dynamic pricing on a historical spot price dataset from Amazon [23].

The structure of the rest of the paper is as follows: Section II discusses related papers. Section III introduces the system model, provides a scenario description, and formulates a mathematical model for the resource trading. Section IV presents blockchain-enabled trading mechanism in bandwidth reservation process. In Section V, we present the performance evaluation of our algorithms. Finally, a conclusion is drawn in Section VI.

II. RELATED WORK

In this section, we review the related work concerning resource computation and bandwidth reservation in vehicular networks, as well as blockchain-enabled resource trading.

TABLE 1. Comparison of related studies.

	Platforms of Blockchain	Blockchain-enabled Resource Trading	Vehicular Reservation Scenario	Buy	Sell	Cancel	Decision Sell/Cancel	Dynamic Pricing
[7], [8], [11]	X	X	√	√	X	X	X	X
[10], [24], [25]	X	X	√	√	X	X	X	√
[21]	PoW-Ethereum	√	X	√	X	X	X	X
[26]–[28]	PoW-Ethereum	√	X	√	√	X	X	√
[29]	PoW-Ethereum	√	X	√	√	√	X	X
This study	DPoS-EOS	√	√	√	√	√	√	√

A. RESOURCE COMPUTATION/BANDWIDTH RESERVATION

Many studies have explored problems in resource reservation, focusing on network-side resource reservation in mobile networks [5], [6]. However, few studies have considered the economic implications of vehicle-side reservations with a focus on minimizing resource consumer expenditure. This is becoming an increasingly interesting area in Fog/Edge computing [7], [8], [9], [10], [11].

Generally, most studies related to reservation resource requests and provisioning mainly put emphasis on on-site competition [30], [31] or immediate request mode. The main difference between those two types of requests is that in competition requests, users compete for the resource through various game theoretic ways, such as auctions, Stackelberg games, etc. [30], [31], [32]. This results in only a limited number of winners acquiring the resources, leading to a risk of failure for some users to provision resources due to the stochastic nature of resource availability and demand. In contrast, immediate requests, as discussed in [7], focus on addressing the challenge of sharing the available spectrum between multiple secondary users and a primary user. In this scenario, primary users offer pricing information to secondary users, allowing them to reserve spectrum and optimize their utility. Chen et al. [8] developed an approach based on meta-learning to assist in reserving resources for computing with the goal of minimizing the cost of using edge services. Zang et al. worked on proposing a smart online reservation framework to minimize the cost of reserving resources for an individual user [9] or multiple users [11]. As a result of limited resources, the corresponding vehicles need to carry out the schedule reservation well in advance in order to ensure they are able to acquire the necessary resources on time.

The advanced reservation request offers a promising solution by providing guarantees, as discussed in [10]. In this research, a solution is proposed that enables advanced reservation at specific time intervals, achieving commendable cost-effectiveness and time efficiency. Despite the benefits of advanced reservation request approaches, challenges persist in dealing with uncertainties present in real-world situations, such as unpredictable mobility that may affect reservation times and locations. This uncertainty may result in the over- or underutilization of restricted resources. For instance, [24] investigated the issues associated with adjusting reservations, focusing on minimizing the initial reservations and on-demand provisioning costs under uncertainty in demand and price. However, existing reservation schemes typically lack trust mechanisms, resulting in unreliable and unguaranteed resources, which are insufficient for safety-critical applications. Without trust, there is the risk of fraud, where resource providers may sell resources without receiving payment.

B. BLOCKCHAIN-ENABLED RESOURCE TRADING

Several studies have explored the application of blockchain technology in various domains. The following studies highlight studies that investigate the use of blockchain and related technologies in enabling efficient and secure trading systems.

In study [26], the authors propose a secure and efficient framework for vehicle-to-grid energy trading within the context of the energy internet by incorporating consortium blockchain and edge computing technologies. They develop an edge computing-based task offloading mechanism to improve the probability of successful block creation. Another study by the authors in [21] introduces a trusted resource allocation mechanism for blockchain-enabled IoT environments, with a specific focus on the cooperation between edge servers and end users. The mechanism utilizes smart contracts and allows end users to select pricing schemes based on their specific delay and price requirements, while the reputation mechanism facilitates the evaluation of edge server performance and reputation. Furthermore, in [27], the authors present a unified energy blockchain system that utilizes a consortium blockchain for secure energy trading in diverse industrial IoT scenarios. They also propose a credit-based payment scheme to address transaction limitations caused by confirmation delays, thereby facilitating rapid and frequent energy trading. Moreover, the authors in [28] propose a system for secure and decentralized bandwidth trading using blockchain technology. Building on this concept, the paper in [29] presents a framework for bandwidth trading that leverages both software-defined networking (SDN) and blockchain technology. However, these approaches suffer from scalability limitations, as frequent transactions can lead to processing delays, and the inherent latency associated with blockchain verification poses challenges for real-time applications in dynamic vehicular networks.

A recent comprehensive review [33] provided an extensive overview of existing research on data sharing and trading in the blockchain domain in vehicle applications. The majority of prior studies in blockchain-enabled trading have

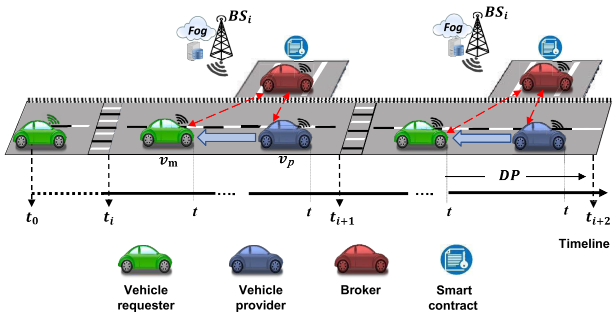

FIGURE 1. An illustration of our scenario and procedure description.

predominantly focused on platforms utilizing proof-of-work (PoW) consensus mechanisms [34] or proof-of-authority (PoA) [35]. In this study, we introduce the application of the EOS blockchain, a platform based on the delegated proof-of-stake (DPoS) consensus mechanism [36]. EOS, known for its scalability, rapid transaction confirmation times, and equitable resource distribution—features lacking in most other consensus mechanisms—serves as a novel approach in our exploration of blockchain-enabled trading. In addition, the current resource trading approach faces more challenges in the context of reservation scenarios in vehicle networks, requiring fast and informed decisions because vehicles need to leave the location of the trading. Another challenge is dynamic pricing, which prompts requesters to seek providers offering lower costs. The main attributes of related studies are summarized in Table 1.

III. BLOCKCHAIN-EMPOWERED RESOURCE TRADING FOR OPTIMIZING BANDWIDTH RESERVATION

This section begins by describing our reservation scenarios and procedures. We then formulate the resource trading problem from both the requester and provider perspectives. Finally, we introduce a blockchain-empowered resource trading architecture. The important notations are listed in Table 2.

A. SYSTEM MODEL AND SCENARIO DESCRIPTION

As shown in Fig. 1, we propose blockchain-enabled resource trading to optimize the reservation scenarios. In this model, each base station (BS) is equipped with a fog/edge server to provide computing offloading services to vehicles. These BSs, including roadside units (RSUs) or Macro Base Stations (MBSs), are strategically positioned along consecutive road segments RS. Each $RS_i; i \in 1 \ldots N$ is covered by a BS_i, playing a pivotal role in establishing wireless connections between fog/edge servers and vehicles connected to the core network. The driving path (DP) is divided into N road segments RS based on the network range of these BSs.[1]

[1]Our assumption is that the path refers to the RS sequence leading to the destination [37].

TABLE 2. List of important notations.

Notations	Definition
v_m	Vehicle requester
v_p	Vehicle provider
DP	The planned driving path
RS_i	i^{th} road segment
BS_i	BS of the i^{th} RS
t_0	Initial reservation request
T	Required time for the vehicle to go through the DP
t_i, t_{i+1}	The reservation start and end at the current RS_i
$\overline{r}_i$	Reservation duration between t_i and t_{i+1}
Requester Perspective	
$D_{v_m^i}(t)$	The bandwidth demand of requester v_m at time t in the BS_i
$S_{v_p^i}(t)$	The surplus bandwidth offered by provider v_p at time t in the BS_i
$C_{v_m^i, v_p^i}(t)$	The bandwidth cost offered by v_p at time t in the BS_i.
$A_{v_m^i, v_p^i}(t)$	The total bandwidth allocated to v_m at time t in a specific BS_i
Provider Perspective	
C_f	The fee charged by the MNO for canceling the reserved bandwidth
P_o	The price of the bandwidth the provider v_p originally bought or reserved from the MNO at t_0
P_b	Probability of v_p finding a requester v_m
A_{nb}	The amount of bandwidth a new requesters needs
N_{nb}	Number of new requesters in a given time period
N_{ns}	Number of providers in a given time period
N_{nb_req}	The number of new requesters in a given time period required for provider v_p to sell its surplus bandwidth at the current market price and offered price
P_s	Offer price at which the provider v_p offer its price
p_{sel}	The selling market: the prices other providers offer
A_{ch}	Cheaper bandwidth of other providers if v_p offer at P_s

Within this network, vehicles function as requesters and providers. The vehicle requester, denoted as v_m, initiates a reservation request targeting to the lowest cost MNOs for each road segment along the DP at time t_0. This request outlines the bandwidth time period T necessary for v_m to complete its intended DP. The time period T is further divided into $\overline{r_i} = [t_i, t_{i+1}]$, representing the time v_m and traversing the coverage area[2] of a RS_i covered by BS_i. The upcoming DP and route details, such as BS coverage area and $\overline{r_i}$, are deduced and relayed based on navigation system data.

In our scenario, vehicle v_m may find itself underbooked and require additional resources, for example, at time t. Vehicle v_m actively seeks to fulfill their resource needs by making requests or tapping into the resources offered by vehicle provider v_p. In this context, the v_p, also having initiated a reservation at t_0, may find itself at time t with excess reserved duration or available surplus bandwidth resources (i.e., overbooking). This could result in an early arrival at the handover point, providing an opportunity to either cancel the surplus reserved resource (subject to cancellation fees charged by the MNO) or sell it to a v_m at the time t. Therefore, the v_p must strategically find a suitable strategy to trade their surplus bandwidth within the same BS_i location area before reaching the handover point to BS_{i+1}. Within the

[2]The diameter of the coverage areas of a BS is approximately 900m, similar to [37], and is typically visible in downtown areas of metropolises.

network, there are vehicles that function as brokers. These vehicles can be stationary, positioned in each segment on the DP.[3] Broker vehicles play a crucial role in facilitating the resource trading process and communicating with MNOs after any trade to mutually allocate bandwidth between the v_m and v_p. Also, the broker vehicle role involves acting as full nodes that are responsible for storing a copy of the blockchain, executing smart contracts, and completing mining work. Our reservation system model focuses only on time-based bandwidth reservations. Additionally, it assumes a consistent level of requested bandwidth (in MB/s) across different BS_i locations. This assumption is based on that vehicles can determine their bandwidth needs by considering data usage, applications, and task models. This allows for the prediction of the required bandwidth using methods similar to those described in [8].

B. RESOURCE TRADING FROM THE REQUESTER PERSPECTIVE

This section focuses on optimizing bandwidth reservation in vehicular networks from the perspective of a vehicle acting as a requester (v_m). The objective is to efficiently allocate bandwidth for v_m within each road segment (RS_i) covered by a specific BS_i. This optimization takes into account the specific bandwidth needs (demands) of v_m at time t, the available supply of bandwidth surplus from the vehicle provider v_p, and the bandwidth cost. The goal is to strike a balance between minimizing costs and satisfying both demand and supply constraints within the dynamic nature of vehicular communication networks. The components of the formulation involving variables, constraints, and objective function are as follows:

1) VARIABLES

- Number of road segments (N): The DP is divided into N segments, each with a specific BS_i providing coverage.
- Requester bandwidth demand ($D_{v_m^i}(t)$): The amount of bandwidth requested by v_m at time t within the BS_i.
- Provider bandwidth supply ($S_{v_p^i}(t)$): The surplus bandwidth available from provider vehicle v_p at time t within the BS_i.
- Bandwidth cost ($C_{v_m^i,v_p^i}(t)$): The cost for v_m to purchase bandwidth from v_p at time t within the BS_i.
- Bandwidth allocation ($A_{v_m^i,v_p^i}(t)$): The amount of bandwidth allocated from v_p to v_m at time t within the BS_i (decision variable).

2) CONSTRAINTS

- Supply constraint: The total bandwidth allocated by v_p in BS_i ($A_{v_m^i,v_p^i}(t)$) cannot exceed its available surplus at t.

$$\sum_{v_m} A_{v_m^i,v_p^i}(t) \leq S_{v_p^i}, \quad \forall v_p^i \tag{1}$$

- Demand constraint: The total bandwidth allocated to a requester vehicle v_m in a specific segment BS_i must satisfy its demand.

$$\sum_{v_p} A_{v_m^i,v_p^i}(t) = D_{v_m^i}(t), \quad \forall v_m^i \tag{2}$$

- Non-negative allocation: The allocation of bandwidth must be non-negative.

$$A_{v_m^i,v_p^i}(t) \geq 0, \quad \forall v_m^i, v_p^i \tag{3}$$

3) OBJECTIVE FUNCTION

The objective is to minimize the total cost of bandwidth allocation while satisfying all demand and supply constraints:

$$\begin{aligned} \text{Minimize} \quad & \sum_{v_m}\sum_{v_p}\sum_{i} C_{v_m^i,v_p^i}(t) \cdot A_{v_m^i,v_p^i}(t) \\ \text{subject to} \quad & \sum_{v_m} A_{v_m^i,v_p^i}(t) \leq S_{v_p^i}(t), \forall v_p^i \\ & \sum_{v_p} A_{v_m^i,v_p^i}(t) = D_{v_m^i}(t), \forall v_m^i \\ & A_{v_m^i,v_p^i}(t) \geq 0, \forall v_m^i, v_p^i \end{aligned} \tag{4}$$

C. RESOURCE TRADING FROM THE PROVIDER PERSPECTIVE

The provider v_p aims to trade this surplus bandwidth to v_m who need demand bandwidth to fulfill their requirements. This section formulates a decision-making model from the provider vehicle (v_p) perspective. The model helps v_p choose between selling its surplus reserved bandwidth or canceling it. This decision involves considering several factors, including potential profit from selling the surplus, cancellation fees, and the likelihood of finding a requester vehicle v_m. The model aims to assist the v_p maximize their utility, specifically focusing on profit and cost-saving. The components of this model are as follows:

1) VARIABLES

- Profit from selling (P_s): Represents the potential profit the v_p can gain by selling surplus bandwidth.
- Cancellation fee (C_f): The fee charged by the MNO for canceling the reserved bandwidth (e.g., 12% following Microsoft Azure strategy [39]).
- Probability of finding a requester (P_b): The likelihood of finding a vehicle that is underbooking and needs the excess bandwidth.

In the next section, we delve into the calculation of the probability of finding a requester (P_b), a crucial factor that influences a providers decision to sell or cancel.

[3]We assume the broker vehicle is parked, according to the AAA Foundation for Traffic Safety survey in 2016, which reported an average driving time per day in the U.S. of only 50.6 minutes [38]. Thus, we could incentivize owners of these vehicles to allow their vehicles to be used for processing blockchain trading and exchange bandwidth tasks.

2) FINDING A REQUESTER (P_B) PROBABILITY

The probability of finding a requester (P_b) within the remaining time in the current BS_i (denoted by $T_{remaining}$) depends on several factors:

- Arrival rate of requester vehicles ($\overline{N_{nb}} = \mathbb{E}[N_{nb}]$): A higher average arrival rate of vehicles at road segment RS_i seeking bandwidth increases the chance of finding a requester v_m.
- Departure rate of provider vehicles ($\overline{N_{ns}} = \mathbb{E}[N_{ns}]$): A lower departure rate of provider vehicles with surplus bandwidth (i.e., more providers selling in the market) reduces the chance of a specific provider v_p offer being chosen.
- Selling price (p_{sel}): Prices offered by other providers (market price) at a lower price are more likely to attract requester vehicles. This assumes that the requesters are greedy and prioritize buying bandwidth with lower prices first.
- Provider v_p selling price (P_s): Considering the price that provider v_p offers (P_s) along with the prices offered by other providers p_{sel}, we can calculate the amount of bandwidth available in the market at a lower cost than what provider v_p is offering. We assume that the other providers have to advertise their resources and prices to the broker, and the provider v_p knows the market book information from the broker. Otherwise, the provider v_p obtains this information from other markets where the resources and prices are public, which is the same as implemented in [38].

The following equation estimates the probability of finding a requester ($P_b(t)$), as a function of the remaining time ($T_{remaining}$), considering these factors:

$$P_b(t) = P_b(T_{remaining}, \overline{N_{nb}}, \overline{N_{ns}}, p_{sel}) \tag{5}$$

a: EXPECTED NUMBER OF REQUIRED REQUESTERS

If we assume requesters follow a greedy strategy, they likely purchase from the cheapest available providers first. They only consider a provider v_p offer (P_s) if no cheaper options exist from other providers. Under this assumption, let $\overline{A_{nb}} = \mathbb{E}[A_{nb}]$ represents the expected amount of bandwidth a new requester demands in the current BS_i. We can calculate the expected number of new requesters needed to sell the surplus bandwidth based on the amount of cheaper bandwidth available (A_{ch}). This refers to the total bandwidth offered by other providers at a lower price P_{sel} than provider v_p offered price P_s (i.e. $P_{sel} < P_s$):

$$N_{nb\_req} = \frac{A_{ch}}{\overline{A_{nb}}} \tag{6}$$

This represents the minimum number of requesters needed to absorb the entire surplus.

b: PROBABILITY OF SELLING WITH POISSON DISTRIBUTION

The average rate of new requesters per minute, $\overline{N_{nb}} = \mathbb{E}[N_{nb}]$ and the remaining time $T_{remaining}$ to sell can be multiplied to get the average new requesters during the remaining time. Using these we can calculate the probability of selling (assuming no new providers with cheaper offers enter the market) using a Poisson distribution:

$$P_b(t) = Poisson(k = N_{nb\_req}, \lambda = \overline{N_{nb}} \cdot T_{remaining}) \tag{7}$$

We assume that the expected value of exiting vehicles is the same as the expected number of entering vehicles.

c: CONSIDERING WORST-CASE SCENARIO

In the worst-case scenario, all new providers sell at lower prices, making the at least necessary number of requesters $2 \cdot N_{nb\_req}$ to be able to sell from provider v_p. Incorporating this into equation (7), we get:

$$P_b(t) = Poisson(k = 2 \cdot N_{nb\_req}, \lambda = \overline{N_{nb}} \cdot T_{remaining}) \tag{8}$$

3) DECISION MODEL

Formulates the decision-making optimization problem by choosing the decision with the largest expected profit. The expected profit from selling is:

$$SellProfit = P_b \cdot P_s \tag{9}$$

And the expected profit from canceling is:

$$CancelProfit = P_o - C_f \tag{10}$$

The decision criteria involve selecting between selling and canceling based on the calculated utility as follows:

- If $SellProfit > CancelProfit$, the provider v_p should choose to sell the excess bandwidth.
- If $SellProfit < CancelProfit$, the provider v_p should choose to cancel the surplus reservation.

The model can be further refined by considering the remaining time $T_{remaining}$ before reaching the handover point to the next BS_{i+1}. This factor can impact the probability of finding a requester from provider v_p surplus bandwidth. Specifically, a longer $T_{remaining}$ may influence the likelihood of encountering potential requester vehicles. With more time left before the next BS_{i+1}, there is a higher chance of encountering new requesters for the surplus bandwidth, thus increasing the probability of a successful sale.

In Algorithm 1, the provider computes the expected utility from selling and the cost of canceling. The decision is then made based on which option offers the highest utility. If the expected profit from selling (considering the probability of finding a requester) is greater than the cancellation fee, the provider v_p chooses to sell. Otherwise, the provider v_p may opt to cancel. If both actions result in equal utility, the decision remains, indicating a need for further analysis or data.

The time complexity of Algorithm 1 is primarily determined by the calculation of utility values and the decision-making process. The calculation of utility values involves basic arithmetic operations, which have a time complexity of $O(1)$. The decision-making process involves a comparison

Algorithm 1 Decision Making for Bandwidth Providers

Input:
- P_o: Original price of the bandwidth
- C_f: Cancellation fee charged by the MNO
- P_b: Probability of finding a buyer
- P_s: Selling price for the bandwidth

Output:
- *Decision*: "Sell" or "Cancel"

Initialize Decision Variables:
- *SellProfit*: Utility from selling (initialized to 0), the expected profit of selling the surplus bandwidth
- *CancelProfit*: Utility from canceling (initialized to 0), the profit received back upon cancellation

Calculate Utility Values:
- $SellProfit = P_b(t) \cdot P_s$
- $CancelProfit = P_o - C_f$

Evaluate Decision:

if $SellProfit > CancelProfit$ **then**
  Set *Decision* to "Sell"
else
  Set *Decision* to "Cancel"
end if
Return $Decision = 0$

operation, which also has a time complexity of $O(1)$. Therefore, the overall time complexity of Algorithm 1 is $O(1)$, indicating that it executes in constant time, regardless of the size of the input.

IV. BLOCKCHAIN DESIGN IN TRADING RESERVED BANDWIDTH

Our proposed architecture for resource trading, which is based on blockchain technology, is depicted in Fig. 2. Blockchain technology provides a secure and transparent way to record transactions, which enhances the reliability and trustworthiness of our system. Upon registration with a recognized authority, each vehicle attains a legitimate status, enabling it to participate securely and efficiently in resource trading. This process operates on a transparent and effective protocol facilitated by the blockchain infrastructure. The resource trading process within our blockchain framework is as follows:

1) **Vehicle provider:** Advertise and register the available bandwidth in data pools managed by brokers.
2) **Vehicle requester:** broadcast their requirements to the network and request a list of suitable provider v_p from the broker.
3) **Broker selection:** Brokers search for the requester v_m needs in the data pools, selecting the provider v_p using a cost-effectiveness algorithm (detailed in Algorithm 2).
4) **Order placement:** The broker informs the requester about the chosen provider v_p. The requester v_m then places an order for the bandwidth with v_p.

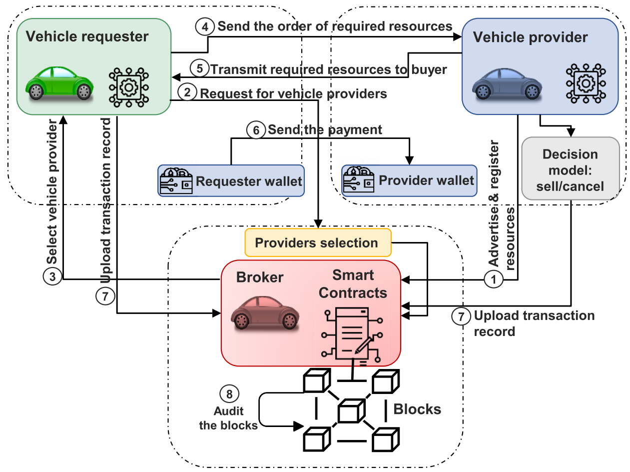

FIGURE 2. Blockchain-based resource trading process.

5) **Bandwidth delivery:** The v_p verifies the order and v_m identity before delivering the bandwidth. Delivery can happen directly after informing the MNO, or via the broker, who is responsible for communicating with the MNO.
6) **Transaction verification:** Upon completion of data transmission, the v_m verifies the transaction success by assessing the validity and completeness of the received data. If satisfied, the v_m sends a token to the v_p public wallet address.
7) **Payment and validation:** Following payment, both parties submit transaction data to the broker for validation and auditing. Brokers who significantly contribute to data sharing are rewarded, incentivizing their participation in solving the DPoS challenge.
8) **Blockchain integration:** Brokers collect, encrypt, and digitally sign transaction data. They then organize it into blocks, each cryptographically linked to the previous one via a hash value. Brokers can act as block producers, adding validated blocks to the blockchain.

A. PROVIDER SELECTION ALGORITHM

In scenarios where dynamic pricing by MNOs and vehicle mobility are key factors, the pursuit for the most cost-effective bandwidth reservation becomes crucial. This necessitates a swift and efficient method for identifying the minimum bandwidth reservation costs. Our provider selection algorithm is specifically designed for this purpose and plays a pivotal role in minimizing these costs. The procedure aligns with the steps laid out in Algorithm 2.

The process begins when a requester v_m, upon entering a BS_i and noticing underbooking, sends a request to the smart contract. Assuming there are providers who have declared their surplus bandwidth, the smart contract then initiates a search for the provider offering the lowest cost.

The smart contract evaluates two potential scenarios:

- **Single provider selection:** If it identifies a solitary provider who can meet the requester demand ($D_{v_m^i}$) at

Algorithm 2 Provider Selection

Let $v_p = \{v_{p1}, v_{p2}, \ldots v_{pP}\}$ be the set of registered providers.
for each provider v_{pk} in the registered providers **do**
  if provider V_{pk} can satisfy the request at a low cost **then**
    Notify the requester v_m with the selected provider: $f(v_{pk}) = 1$
  else if multiple providers $\{v_{pk1}, v_{pk2}, \ldots, v_{pkj}\}$ can fulfill the request at a low cost where $f(v_{pkj})$ = True for all $(j = 1, 2, \ldots)$ **then**
    Divide the request among the multiple providers.
    Notify the requester v_m with the selected providers: $\sum_{v_{pkj}} f(v_{pkj}) = 1$
  end if
end for

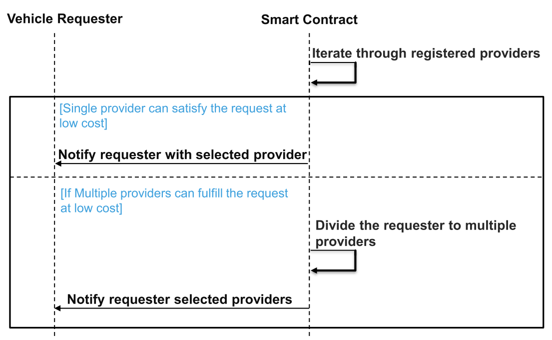

FIGURE 3. Workflow of provider selection in resource trading process.

the most affordable rate, this provider is chosen for the bandwidth reservation, which we refer to as the full scenario.

- **Multiple provider selection:** Conversely, if there are several providers capable of satisfying the request at a low cost, the smart contract distributes the required bandwidth among these providers, which we refer to as the partial scenario. The workflow of provider selection is shown in Fig. 3.

Following the selection process, the smart contract communicates with the requester v_m, informing them of the selected providers and the total cost. This enables the v_m to move forward with the bandwidth reservation, assured of having secured the most cost-effective option. The time complexity of Algorithm 2 is primarily determined by the iteration over all providers and the selection process. The iteration over all providers has a time complexity of $O(n)$, where n is the number of providers. The selection process involves a comparison operation, that has a time complexity of $O(1)$. Therefore, the overall time complexity of Algorithm 2 is $O(n)$, indicating that its execution time increases linearly with the number of providers.

B. EOS BLOCKCHAIN SECURITY AND PRIVACY

Choosing a blockchain for trading reserved bandwidth depends on several factors. These include security, scalability, and transaction speed. Below are some aspects that advantage the use of blockchain in our system.

1) UNRELIABLE RESOURCE NEEDS

Through the use of smart contracts, blockchain technology optimally manages the resources and avoids some of the risks associated with the distribution of the resources [40]. Through reputation systems, it can name providers who offer unreliable resources. Smart contracts are responsible for handling the provider reputation scores derived from feedback for each reservation. Confirmation transactions, which take place on the blockchain to signify agreement, are a confirmation transaction that occurs when a provider agrees to a reservation. Non-delivery penalties are defined in the smart contract to guarantee the reliability of the resources.

2) FRAUDULENT PROVIDERS

A risk that a provider sells a resource without getting paid for it, can be managed with a robust authentication and authorization mechanism. Secure payment and resource delivery can be guaranteed with secure escrow services or blockchain-based smart contracts [40]. Blockchain inherent attributes, such as transparency and immutability, can help minimize the potential level of fraud wherever it is applied. As every transaction performed on the blockchain is recorded on a public ledger, the tires cannot be changed or deleted once a transaction has been approved. Such transparency will discourage fraudulent activities since every transaction conducted in the system can be accounted for.

3) IMPROVING PRIVACY

Although being transparent, blockchain can also be highly private. For example, [40] explained that vehicles can participate in transactions through pseudonymous identities. In this way, they can maintain high privacy, as their real identities will not be exposed. Meanwhile, these entities may continue interacting with the network. Moreover, the high level of privacy can be achieved through the use of advanced cryptographic techniques. For instance, zero-knowledge proofs can help the vehicles prove that their transactions are valid without necessarily disclosing any additional information about the transaction.

4) EFFICIENCY AND SPEED

Blockchain has the potential to simplify and/or/ automate the transaction process, making it fast and efficient. This is highly useful in the case of resource trading and exchange reservation. EOS is a blockchain operating platform with excellent scalability and an original consensus mechanism known as DPoS. Originally designed for more than a million of transactions per second, it outperforms many blockchains in terms of transaction speed and the time it takes to execute each transaction [40]. Such massive scale is largely due to its DPoS consensus, where token holders elect delegates to help decide transactions, as well as defend the network.

For applications involving the trading of bandwidth resources, the EOS blockchain provides features make it attractive. For example, the fee of transaction is eliminated through a resource allocation model. The model allows to the vehicles to use resources such as network bandwidth and CPU through EOS tokens. This approach ensures an equitable for distribution resources, which is a common challenge in blockchain bandwidth trading. EOS also provides a versatile platform for use cases, and enhancing the decentralized applications (dApps). For example, in the case of resource reservation and trading.

C. VULNERABILITIES AND MITIGATIONS IN EOS BLOCKCHAIN

While blockchain technology offers significant security advantages for resource trading in our reservation exchange scenario. It is important to be aware of potential vulnerabilities and corresponding mitigation strategies. EOS is one of the most representative Blockchain 3.0 platforms, which involves many new features like the DPoS consensus model and smart contracts. These features enable a massive transactions and a prosperous dApps ecosystem. However, there are vulnerabilities like any technology which are described in the [41].

1) POTENTIAL VULNERABILITIES

a: SMART CONTRACT VULNERABILITIES

Attacking smart contracts allows the funds to be obtained, and the reservation exchange logic can be manipulated by malicious code.

b: SYBIL ATTACKS

The ability to create large numbers of fake identities in order to disrupt or control the network, or to vote in ways which are unfair to the rest of the network.

c: DOS ATTACKS

The ability of a malicious participant to flood the network with transactions that negatively impact the existing reservation exchange.

2) MITIGATION STRATEGIES

Here we have discussed some of the security features of EOS, which may serve to suggest ways reduce vulnerabilities [41]:

Role-based access control (RBAC): Provides a fine-grained control level over who can to interact with smart contracts.

Byzantine fault tolerance (BFT): Resilience in EOS is maintained by consensus for dealing with malicious nodes affecting the network.

V. PERFORMANCE EVALUATION

Results of extensive experimentation have been presented in this section. To demonstrate the efficiency of proposed method, we compare our algorithms with various existing bandwidth reservation schemes in vehicular settings.

A. EXPERIMENTAL SETUP

This section details the design and evaluation of our bandwidth trading system for vehicular networks. The system leverages the EOS blockchain framework for secure resource exchange.

1) BLOCKCHAIN CONFIGURATION

The EOS blockchain is configured through a file called config.ini. This file controls how an instance of nodeos operates. It allows users to specify the nodes to which the nodeos instance establishes connections (EOS mainnet), define the plugins to use, and customize the behavior of the node through plugin-specific options [40]. The EOS blockchain virtual machine is configured to recognize and execute instructions in WASM format, allowing the use of various programming languages, including C++, C, and Rust, for smart contract development. The connection between the EOS library and the WASM binary code was established through web assembly modules, which ensured seamless interaction and execution of smart contracts [40].

The implementation includes the development of smart contracts using the C++ programming language. In addition, to develop smart contracts for the EOS blockchain, it is necessary to install the contract development toolkit (CDT), which includes the eosio-cpp command. This command is used for various purposes, primarily to compile the C++ files of the contract into the WebAssembly (WASM) format, in addition to generating the necessary ABI files. These ABI files are crucial for facilitating the conversion of data between JSON and its binary representation.

Node management in EOS is handled through a core EOS node daemon called nodeos [40]. Nodeos, using plugins, is configured to run a node in the EOS network. Nodes can be connected to an existing blockchain network and synchronize existing history or used to create a new blockchain with custom parameters [40]. An API node is a critical component of the EOS blockchain network that serves as an interface between users, including dApps, and an EOS blockchain. API nodes serve one of the following roles when handling incoming client requests received through one of the chain api plugin endpoints: Push API node, Chain API node, and Pull API node.

2) SYSTEM SETUP AND CONFIGURATION

In the system setup, Ubuntu 20.04 was installed on a laptop equipped with an Intel Core i7 CPU, 16 GB of RAM and a 1000 GB SSD. Subsequently, EOS version 1.8.1 was installed, and the instructions described in [40] were followed to configure the Testnet environment. It should be noted that all comparisons to the Ethereum blockchain were done before its switch [42] from PoW to proof-of-stake (PoS) consensus. In the EOS blockchain, the execution of smart contract transactions is contingent upon two distinct resources: computing power (CPU) and network bandwidth (NET). For a sender to engage effectively with a smart

TABLE 3. CPU and net bandwidth resource consumption of the trading process.

Transactions	CPU Usage (µs)	NET Usage (Bytes)
Bandwidth registrations	282	128
Request for bandwidth	372	120
Place order	211	128
Accept order & transmit token	414	184

contract on EOS, it is necessary to have sufficient resources. In scenarios where the sender lacks the required resources, EOS prompts them to acquire resources by using EOS tokens. The EOS infrastructure introduces two critical mechanisms to alleviate resource scarcity for senders. The first mechanism allows senders to stake EOS tokens, which consequently provides them with the necessary bandwidth and storage on the blockchain. This staking process plays a pivotal role in enabling the execution of transactions. Additionally, the EOS ecosystem permits token holders to distribute their surplus CPU and NET resources to other entities within the system [40]. This approach of resource delegation or rental significantly augments the efficiency of resource utilization across the network. A notable feature distinguishing EOS from platforms like Ethereum is the absence of transaction fees for smart contracts and dApps. Token ownership in the EOS network grants vehicles a proportionate share of the network computational power. Registered vehicles can benefit from free usage on the EOS platform, but EOS tokens remain necessary for transactions and operations on the blockchain. When vehicles invest tokens to obtain RAM, EOS intelligently assigns the resources of block producers (BPs) to the contract owners. This system optimizes the allocation of BP resources. EOS ensures a fair distribution of resources by allocating them proportionally based on the number of EOS tokens staked. A crucial aspect to highlight is the way the EOS system facilitates the reversion of allocated CPU and NET resources back to the available pool for redistribution when vehicles opt to unstake their tokens. This dynamic process of resource allocation and reallocation is essential to maintaining a balanced distribution of resources within the EOS network.

3) METRICS

The metrics are utilized to evaluate the following: 1) CPU and bandwidth resource consumption costs associated with various processes of our proposed bandwidth-trading architecture; 2) Throughput, which is defined as the number of transactions processed per second (TPS) over a given time interval; 3) Block generation time (BGT), which is the duration required to create and add a new block to the EOS blockchain; 4) Smart contract execution time, which refers to the time taken to process and validate all transactions associated with a smart contract; 5) Cost-effectiveness of exchange reservation through resource trading, which measures how much cost is saved in our methodology; 6) The average return payment for four different strategies.

4) BENCHMARK APPROACHES

The following benchmark approaches are selected for comparison with our algorithms. The first baseline is the immediate reservation request (IRR) scenario, where the vehicle v_m or v_p requesting reservation prices at time t_0. This request is immediate or on-site, meaning the vehicle doesn't have a pre-defined schedule for updating its reservation in the future, potentially leading to the need for updates based on cost trends and available resources. This approach is similar to existing methodologies employed in [8], [9], and [11]. The second baseline is the advanced reservation request (ARR) scenario involves the vehicle placing a smart reservation request in advance. This request optimizes reservation timing for cost-efficiency within a specific timeframe and schedules future updates. However, uncertainties remain regarding future resource availability and costs. This approach is similar to the method used in [10]. The third baseline is optimization of reservations from MNO, as in [24] and [43], this approach involves vehicles placing smart reservation requests with MNOs in advance. The vehicle continuously monitors and updates its resource needs to avoid uncertainties. This can occur when a vehicle finds itself underbooked (having extra bandwidth) but lacks information about other vehicles that might need to sell their unused resources. Here, the MNO doesn't facilitate resource exchange between vehicles, leading to potential inefficiencies. The always cancel strategy presented in [10], [24], and [25] imply that the excess or overbooked bandwidth is always canceled. Another strategy is never canceling, which implies that if there is no cancellation, all the bandwidth is lost without any return. In addition, we compare our EOS blockchain with the Ethereum blockchain baseline demonstrated in [21], [26], [27], [28], and [29].

5) DATASET DESCRIPTION

To assess the effectiveness of our resource trading, we utilized a historical dataset of Amazon spot prices, which are subject to fluctuations influenced by factors such as capacity, demand, geographic location, and specific instance types [23]. Given the time-sensitive nature of various applications, vehicles require both computing instances and communication links, i.e., bandwidth. Our assumptions are that the pricing for setting up computing and communication resources aligns with Amazon's spot pricing model, as previously referenced in [9] and [11]. For this study, we collected pricing data from all available instances and two specific regions, namely us-west-1b and us-west-1c.

B. RESULTS

Table 3 displays the resource consumption costs associated with various processes of our proposed bandwidth trading architecture. Through our experiments, we observed the "Accept order and transmit token" process incurs higher resource consumption costs compared to other processes. This outcome is due to the complexity and number of

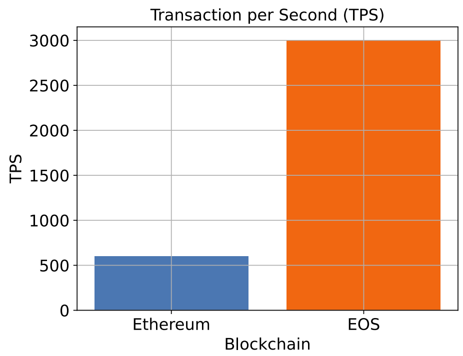

FIGURE 4. Throughput comparison between EOS and Ethereum.

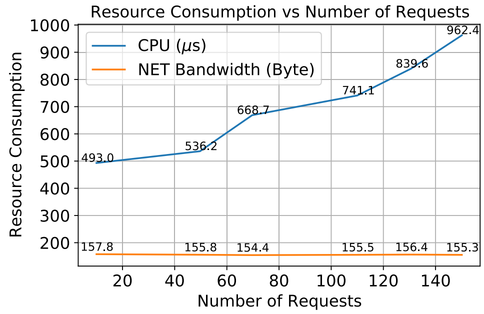

FIGURE 5. Number of requests vs. resource consumption.

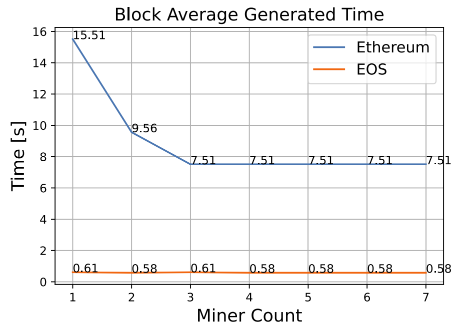

FIGURE 6. Block generation time.

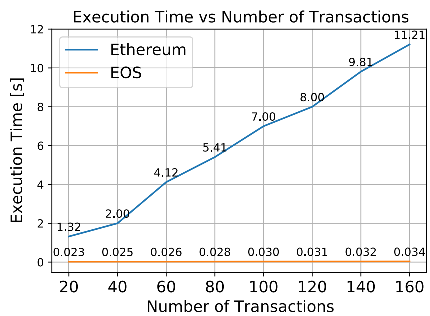

FIGURE 7. Execution time comparison between Ethereum and EOS.

operations involved in the transaction process. This process includes transferring tokens from the requester to the provider and updating the state of the smart contract, reflecting the new state of the order and the balances of the involved parties. These operations require more CPU and network usage compared to other processes.

1) TRANSACTIONS PER SECOND (TPS)

The EOS blockchain platform, in comparison to Ethereum (as shown in Fig. 4) and other earlier PoW-based platforms such as Bitcoin, exhibits superior performance in the deployment and execution of smart contracts, particularly in terms of throughput. Throughput, an essential performance metric, is defined as the number of transactions processed per second over a given time frame [44]. The throughput of TPS for a specific vehicle v in the interval from t_h to t_w, where Tx is a resource transaction, can be calculated using the equation:

$$\mathrm{TPS}_v = \frac{\mathrm{Count}(\mathrm{Tx\ in}(t_h, t_w))}{t_h - t_w}\ (\mathrm{txs/s}) \tag{11}$$

For assessing the average throughput across participating vehicles, the following equation is used:

$$\mathrm{TPS}_{\mathrm{avg}} = \frac{\sum(\mathrm{TPS}_v)}{N}\ (\mathrm{txs/s}) \tag{12}$$

This contrast in throughput is attributed to the fundamental differences in consensus mechanisms between EOS and PoW-based platforms. PoW, utilized by Ethereum and Bitcoin, involves intense computational efforts for hashing calculations, which inherently limits throughput. EOS, on the other hand, adopts a DPoS consensus algorithm. This algorithm is engineered to curtail the waste of computational power, thereby facilitating a significant increase in throughput.

$$B_{\mathrm{tt}}T_{\mathrm{v}} = \frac{\sum_{\mathrm{Tx}}(t_{\mathrm{TxConf}} - \mathrm{MAX}_{\mathrm{block}}(t_{\mathrm{TxDNE}}))}{\mathrm{Count}(\mathrm{Tx\ in}(t_h, t_w))}\ (\mathrm{tx/s}) \tag{13}$$

2) RESOURCE CONSUMPTION

Our study focused on assessing the impact of varying numbers of bandwidth reservation requests on the resource consumption of our smart contract deployed on the EOS blockchain network. Specifically, we aimed to understand how patterns of resource usage fluctuate in response to handling an increasing number of requests.

The smart contract executes a series of operations during bandwidth reservations. These operations include selecting cost-effective providers, verifying the requester financial balance, and confirming requested bandwidth availability with the chosen provider, ultimately finalizing the reservation. We recorded and analyzed the resource consumption associated with these operations, specifically focusing on CPU and NET bandwidth. Fig. 5 depicts the relationship between the number of reservation requests and the corresponding average resource consumption. This figure illustrates how resource usage within the smart contract changes as the volume of bandwidth reservation requests increases. Analyzing this data provides insights into the scalability and efficiency of the EOS smart contract in managing varying levels of demand, with light consumption of resources (CPU and bandwidth) as the request volume grows.

3) BLOCK GENERATION TIME (BGT)

In blockchain networks, BGT refers to the duration required to create and add a new block to the blockchain. The EOS blockchain stands out by generating blocks every 0.5 seconds, which is a significant advantage in terms of transaction processing speed. In this process, once a block producer (analogous to a miner) completes a transaction, there is a brief interval before the next half-second mark, at which point the processed block is disseminated to other block producers in the network. This streamlined mechanism offers users a rapid and fluid blockchain experience, similar to that of web applications or payment systems.

To quantify the time taken from the initial processing of a transaction to its confirmation in the blockchain, a metric called transaction confirmation time is used. This essentially measures the delay between a transaction being submitted and it being officially included in a block on the blockchain network. The following formula, adapted from [44], calculates the transaction confirmation time for a participating vehicle v:

Here, t_{TxConf} denotes the moment a transaction is initially confirmed, and $\text{MAX}_{\text{block}}(t_{\text{TxDNE}})$ represents the timestamp marking the completion of the final transaction within the same block that contains Tx.

To assess the overall system performance, we employ the equation below to determine the average BGT:

$$B_{\text{tt}}T_{\text{avg}} = \frac{\sum_{v}(B_{\text{tt}}T)}{N} \text{ (txs/s)} \tag{14}$$

Both Bitcoin and Ethereum operate by accumulating broadcast transactions into blocks, utilizing the PoW mining algorithm. In PoW, nodes solve a complex computational puzzle to validate transactions and earn rewards. Ethereum has notably decreased its block generation time from 10 minutes to about 15 seconds, enhancing the rate of transactions. However, EOS use of the DPoS mechanism, where stakeholders elect witnesses to validate transactions, offers a more efficient solution, resulting in shorter block generation times and faster transaction confirmations.

Fig. 6 shows the block generation times for various smart contract platforms. In the EOS blockchain, new blocks are produced consistently every 500 milliseconds, independent of the number of miners.

4) SMART CONTRACT EXECUTION TIME

Another crucial performance metric for blockchain platforms is contract execution time. This refers to the time taken, measured in seconds, to process and validate all transactions associated with a smart contract. Transaction processing and validation times can be a bottleneck for blockchain platforms, especially those using PoW like Ethereum. Limited block sizes and slower block generation times in PoW systems can lead to longer execution times for smart contracts. In contrast, EOS faster block generation times and efficient resource allocation mechanisms contribute to faster contract execution times, making it a strong candidate for real-time reservation applications.

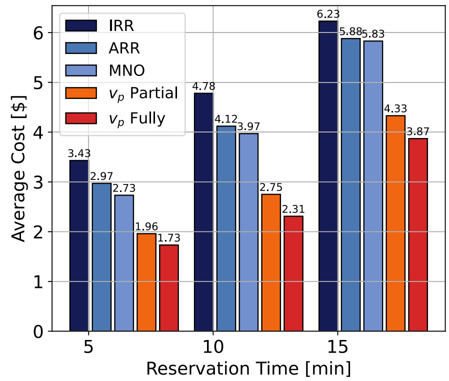

FIGURE 8. Average cost of bandwidth between 7:00-11:00.

For determining the transaction execution time for a participating vehicle v, within a blockchain network, the following equation is applied [44]:

$$CET_v = \frac{\sum_{\text{Tx}}(t_{\text{TxDNE}} - t_{\text{TxSRT}})}{\text{Count}(\text{Tx in}(t_w, t_h))} \text{ (tx/s)} \tag{15}$$

Here, t_{TxSRT} represents the commencement of a transaction's execution, while t_{TxDNE} marks its completion. The average execution time across participating vehicles can be calculated using the equation:

$$CET_{avg} = \frac{\sum v(CET_v)}{N}, (tx/s) \tag{16}$$

Fig. 7 illustrates the average execution time for two popular smart contract platforms across different transaction volumes. The average is computed by performing the experiment 10 times for each transaction volume. The findings demonstrate that the execution time for both blockchain platforms increases with the growing number of broadcasted transactions. However, EOS outperforms Ethereum in terms of performance at both low and high transaction rates, making it a suitable platform for real-time reservation applications.

5) COST OF EXCHANGE RESERVATION

This section evaluates the cost-efficiency of bandwidth reservation exchange through vehicle provider (v_p) in both fully (single provider) and partially (multiple providers) scenarios, compared to IRR, ARR and MNOs. We analyze two scenarios: fully where a single provider bandwidth resource satisfy the reservation request, and partially where resources are pooled from multiple providers. We conduct experiments at different times of day (morning, afternoon, and night) to capture potential cost fluctuations, each represented in Fig. 8, 9 and 10, respectively. These figures depict not only varying bandwidth costs but also the cost

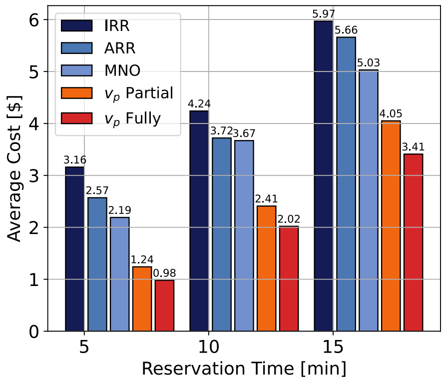

FIGURE 9. Average cost of bandwidth between 12:00–17:00.

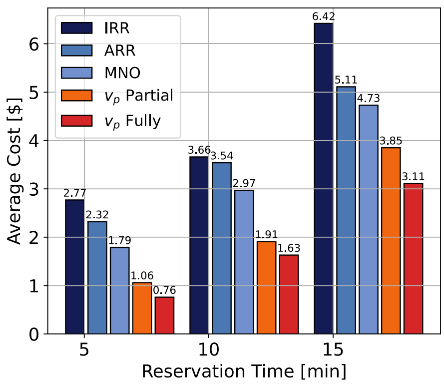

FIGURE 10. Average cost of bandwidth between 18:00–21:00.

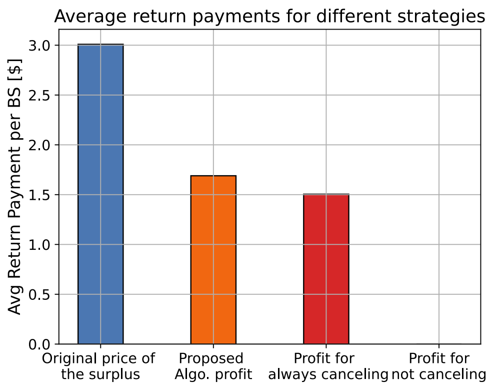

FIGURE 11. Average return payments for different strategies.

implications under different reservation approaches (fully and partially) across the three time periods.

The findings revealed a notable trend in bandwidth pricing across these phases. During the morning phase, bandwidth costs were at their peak, driven by a high volume of reservation requests. This trend shifted as the day advanced, with costs diminishing through the afternoon and further into the night. A key outcome of the study is the cost-effectiveness of resource trading in exchange reservations via resource providers (fully and partially), evident across all three phases. This efficiency stems from the fact that these vehicles had previously purchased bandwidth from MNOs. Therefore, when these vehicles offered their surplus bandwidth, their prices for reservations were comparatively lower than direct reservations from MNOs (usually, the price for an IRR is higher than when reserved in advance ARR mode [10], [24], [25]). This trend is depicted in the figures, showing the total cost implications for bandwidth exchange scenarios.

The findings from our research suggest that the EOS blockchain exhibits significant potential for decentralized applications, particularly those requiring real-time operations like reservation applications. This positions the EOS blockchain as a pivotal technology in the evolving landscape of decentralized application development.

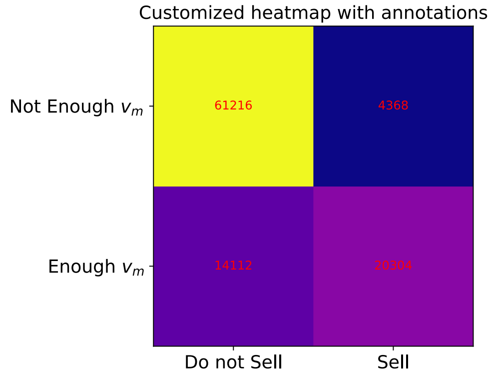

FIGURE 12. Unnormalized confusion matrix: frequency of different scenarios based on the strategy used and the availability of v_m.

6) PROVIDER DECISION (SELLING OR CANCELING)

Fig. 11 presents a comparison of the average return payment for four different strategies: the original price of the surplus (P_o), our proposed algorithm profit, the profit for always canceling (a naive strategy), and the profit for not canceling. The original price of the surplus is the price at which the provider v_p was reserved at time t_0. The always canceling strategy, represented in [10], [24], and [25], implies that the surplus or overbooked bandwidth are always canceled. Another strategy is never canceling, which implies that if cancellation does not occur, all the bandwidth is lost without any return. As can be seen, our proposed algorithm, which shows a slightly higher return payment, is better than the naive always canceling strategy, saving around 19%. This comparison provides valuable insights into the effectiveness of different strategies in terms of their return payments, which can be crucial for the decision-making model for vehicle provider (v_p) registered in our trading architecture.

Fig. 12 presents an unnormalized confusion matrix derived from 100,000 simulations. The matrix illustrates the frequency of different scenarios based on the strategy used and the availability of the vehicle requester (v_m). For instance, in 20,304 out of the 100,000 simulations, the strategy chose to sell surplus bandwidth when there was a sufficient number of v_m available. Conversely, in 4,368 simulations, the strategy aimed to sell, but there were not enough v_m to do so. When the strategy decided not to sell, there were two possible outcomes: In 61,216 simulations, there were enough vehicles (v_m), and in 14,112 simulations, there were enough v_m, but the strategy still decided not to sell.

We recognize the limitations of relying solely on simulations. Real-world environments involve unforeseen complexities such as dynamic traffic patterns, unpredictable network behavior, and potential human error. Simulations may not fully capture these nuances. Addressing these limitations is crucial for enhancing our work. In our future work, we plan to address these limitations by leveraging existing datasets. We intend to incorporate real-world traffic and network data into our simulations to enhance their realism and generalizability.

VI. CONCLUSION

This research presents significant insights into using blockchain for managing trading and bandwidth reservations in vehicular networks. It introduces the use of blockchain smart contracts for efficient and secure bandwidth trading between vehicles. For that purpose, we selected the efficient platform EOS as a key component of our proposed solution. EOS demonstrated impressive performance with low latency and high transaction processing capability, handling up to 1000 requests per second. This high throughput is essential for quick and smooth processing of transactions and updates, making EOS particularly suitable for scenarios like bandwidth reservation among vehicles. We developed two algorithms to enhance the system's efficiency. The first, a provider selection algorithm, prioritizes cost-effective service providers. The second, a decision-making model for providers, enables providers to choose between selling excess reserved bandwidth or canceling it. Our analysis indicated that trading bandwidth between vehicles is more economical than reserving bandwidth directly from MNOs. The decentralized nature of the EOS blockchain facilitates trust in trading resources between vehicles seamlessly. This setup enables vehicles with excess bandwidth to offer their resources and those needing additional bandwidth to acquire it efficiently.

REFERENCES

[1] Y. Wang, X. Tao, X. Zhang, P. Zhang, and Y. T. Hou, "Cooperative task offloading in three-tier mobile computing networks: An ADMM framework," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2763–2776, Mar. 2019.

[2] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward computation offloading in edge computing: A survey," *IEEE Access*, vol. 7, pp. 131543–131558, 2019.

[3] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[4] IEEE-Spectrum. *6 Key Connectivity Requirements of Autonomous Driving*. Accessed: Jun. 28, 2024. [Online]. Available: https://spectrum.ieee.org/6-key-connectivity-requirements-of-autonomous-driving

[5] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," in *Proc. IEEE Conf. Comput. Commun.*, May 2017, pp. 1–9.

[6] A. A. Al-Khatib and A. Khelil, "Priority- and reservation-based slicing for future vehicular networks," in *Proc. 6th IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2020, pp. 36–42.

[7] D. Niyato and E. Hossain, "Competitive spectrum sharing in cognitive radio networks: A dynamic game approach," *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2651–2660, Jul. 2008.

[8] D. Chen, Y.-C. Liu, B. Kim, J. Xie, C. S. Hong, and Z. Han, "Edge computing resources reservation in vehicular networks: A meta-learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5634–5646, May 2020.

[9] S. Zang, W. Bao, P. L. Yeoh, B. Vucetic, and Y. Li, "Filling two needs with one deed: Combo pricing plans for computing-intensive multimedia applications," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1518–1533, Jul. 2019.

[10] A. A. Al-Khatib, F. Al-Khateeb, A. Khelil, and K. Moessner, "Optimal timing for bandwidth reservation for time-sensitive vehicular applications," in *Proc. IEEE 6th Int. Conf. Fog Edge Comput. (ICFEC)*, May 2022, pp. 94–99.

[11] S. Zang, W. Bao, P. L. Yeoh, B. Vucetic, and Y. Li, "SOAR: Smart online aggregated reservation for mobile edge computing brokerage services," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 527–540, Jan. 2023.

[12] M. Liwang and X. Wang, "Overbooking-empowered computing resource provisioning in cloud-aided mobile edge networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 5, pp. 2289–2303, Oct. 2022.

[13] X. Ren, C. Qiu, Z. Chen, X. Wang, D. Niyato, and W. Wang, "CompCube: A space-time-request resource trading framework for edge-cloud service market," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 1–13, Sep. 2023.

[14] Y. Liao, X. Qiao, Q. Yu, and Q. Liu, "Intelligent dynamic service pricing strategy for multi-user vehicle-aided MEC networks," *Future Gener. Comput. Syst.*, vol. 114, pp. 15–22, Jan. 2021.

[15] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Data collection and wireless communication in Internet of Things (IoT) using economic analysis and pricing models: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2546–2590, 4th Quart., 2016.

[16] H. Xu and B. Li, "Dynamic cloud pricing for revenue maximization," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 158–171, Jul. 2013.

[17] S. Sen, C. Joe-Wong, S. Ha, and M. Chiang, "A survey of smart data pricing: Past proposals, current plans, and future trends," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 1–37, Nov. 2013.

[18] S. Ha, S. Sen, C. Joe-Wong, Y. Im, and M. Chiang, "TUBE: Time-dependent pricing for mobile data," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, Aug. 2012, pp. 247–258.

[19] J. Ferdous, M. P. Mollah, M. A. Razzaque, M. M. Hassan, A. Alamri, G. Fortino, and M. Zhou, "Optimal dynamic pricing for trading-off user utility and operator profit in smart grid," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 2, pp. 455–467, Feb. 2020.

[20] L. Yang, H. Kim, J. Zhang, M. Chiang, and C. W. Tan, "Pricing-based decentralized spectrum access control in cognitive radio networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 522–535, Apr. 2013.

[21] H. Cheng, Q. Hu, X. Zhang, Z. Yu, Y. Yang, and N. Xiong, "Trusted resource allocation based on smart contracts for blockchain-enabled Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 7904–7915, Jun. 2022.

[22] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng, "When Internet of Things meets blockchain: Challenges in distributed consensus," *IEEE Netw.*, vol. 33, no. 6, pp. 133–139, Nov. 2019.

[23] *Amazon EC2 On-Demand Pricing*. Accessed: Jun. 28, 2024. [Online]. Available: https://aws.amazon.com/ec2/spot/pricing/

[24] A. A. Al-Khatib, M. U. Hassan, and K. Moessner, "Heuristic optimization of bandwidth reservation cost for vehicular applications," in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 4909–4915.

[25] A. Al-khatib, K. Moessner, and H. Timinger, "Optimizing bandwidth reservation decision time in vehicular networks using batched LSTM," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 2, pp. 963–972, 2024.

[26] Z. Zhou, L. Tan, and G. Xu, "Blockchain and edge computing based vehicle-to-grid energy trading in energy internet," in *Proc. 2nd IEEE Conf. Energy Internet Energy Syst. Integr. (EI2)*, Oct. 2018, pp. 1–5.

[27] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018.

[28] B. M. Yakubu, M. M. Ahmad, A. B. Sulaiman, A. S. Kazaure, M. I. Khan, and N. Javaid, "Blockchain based smart marketplace for secure internet bandwidth trading," in *Proc. 1st Int. Conf. Multidisciplinary Eng. Appl. Sci. (ICMEAS)*, Jul. 2021, pp. 1–6.

[29] Y. E. Oktian, T.-T.-H. Le, U. Jo, and H. Kim, "Blockchain-powered bandwidth trading on SDN-enabled edge network," *IEEE Access*, vol. 10, pp. 114024–114039, 2022.

[30] Y. Cao, C. Long, T. Jiang, and S. Mao, "Share communication and computation resources on mobile devices: A social awareness perspective," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 52–59, Aug. 2016.

[31] I. Bajaj, Y. H. Lee, and Y. Gong, "A spectrum trading scheme for licensed user incentives," *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 4026–4036, Nov. 2015.

[32] Y. Chen, Z. Li, B. Yang, K. Nai, and K. Li, "A Stackelberg game approach to multiple resources allocation and pricing in mobile edge computing," *Future Gener. Comput. Syst.*, vol. 108, pp. 273–287, Jul. 2020.

[33] J. Gao, C. Peng, T. Yoshinaga, G. Han, S. Guleng, and C. Wu, "Blockchain-enabled Internet of Vehicles applications," *Electronics*, vol. 12, no. 6, p. 1335, Mar. 2023.

[34] A. Porat, A. Pratap, P. Shah, and V. Adkar, "Blockchain consensus: An analysis of proof-of-work and its applications," Stanford Univ., Stanford, CA, USA, Tech. Rep. 17au-cs244b, 2017. [Online]. Available: https://www.scs.stanford.edu/17au-cs244b/labs/projects/porat_pratap_shah_adkar.pdf

[35] K. Azbeg, O. Ouchetto, S. Jai Andaloussi, and L. Fetjah, "An overview of blockchain consensus algorithms: Comparison, challenges and future directions," in *Advances on Smart and Soft Computing* (Advances in Intelligent Systems and Computing). Singapore: Springer, May 2021, pp. 357–369.

[36] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2018, pp. 1545–1550.

[37] A. Nadembega, T. Taleb, and A. Hafid, "A destination prediction model based on historical data, contextual knowledge and spatial conceptual maps," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 1416–1420.

[38] D. Han, W. Chen, and Y. Fang, "A dynamic pricing strategy for vehicle assisted mobile edge computing systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 420–423, Apr. 2019.

[39] *Azure Reserved Virtual Machine Instances*. Accessed: Jun. 28, 2024. [Online]. Available: https://azure.microsoft.com/en-us/pricing/reserved-vm-instances/

[40] B. Xu, D. Luthra, Z. Cole, and N. Blakely, "EOS: An architectural, performance, and economic analysis," *Retrieved June*, vol. 11, pp. 41–66, 2018.

[41] N. He, H. Wang, L. Wu, X. Luo, Y. Guo, and X. Chen, "A survey on EOSIO systems security: Vulnerability, attack, and mitigation," 2022, *arXiv:2207.09227*.

[42] E. Kapengut and B. Mizrach, "An event study of the Ethereum transition to proof-of-stake," *Commodities*, vol. 2, no. 2, pp. 96–110, Mar. 2023.

[43] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Trans. Services Comput.*, vol. 5, no. 2, pp. 164–177, Apr. 2012.

[44] P. Zheng, Z. Zheng, X. Luo, X. Chen, and X. Liu, "A detailed and real-time performance monitoring framework for blockchain systems," in *Proc. IEEE/ACM 40th Int. Conf. Softw. Eng., Softw. Eng. Pract. Track*, May 2018, pp. 134–143.

ABDULLAH AL-KHATIB received the M.S. degree in network technology from the National University of Malaysia (UKM), Malaysia, in 2017. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Information Technology, specializing in communications engineering. Additionally, he works as a Research Associate with the Institute for Data and Process Science (IDP), Landshut University of Applied Sciences, Germany. His research interests include wireless communication systems, the Internet of Things, and fog/edge computing.

HALEEMAH HADI received the B.S. degree in computer science from Sana'a University, in 2019. She is currently pursuing the M.S. degree in computer science with the Technical University of Applied Sciences Augsburg, Germany, while also working as a Software Developer with Cadenas GmbH. Her research interests include machine learning and blockchain technologies.

HOLGER TIMINGER (Senior Member, IEEE) received the Dipl.-Ing. and Dr.-Ing. (Ph.D.) degrees in electrical engineering from Ulm University, Ulm, Germany, in 2002 and 2005, respectively. From 2002 to 2008, he was a Research Assistant and a Research Scientist with Philips Research, Hamburg, Germany, where he conducted research in the field of medical imaging, algorithms, and technology. From 2008 to 2011, he was the Project Manager for large-scale international imaging projects of Philips Healthcare. Since 2011, he has been a Professor of project management with Landshut University of Applied Sciences, Landshut, Germany, where he founded and heads the Research Institute for Data and Process Science. He is currently the author of several books related to modern project management.

KLAUS MOESSNER (Senior Member, IEEE) was the Founding Chair of the IEEE DYSPAN Working Group (WG6) on sensing interfaces for future and cognitive communication systems. He is currently a Professor of communications engineering with Technical University Chemnitz and also a Professor of cognitive networks with the Institute for Communication Systems, University of Surrey. He was involved in a large number of projects in cognitive communications, service provision, and the IoT areas. He was responsible for the work on cognitive decision-making mechanisms in the CR Project ORACLE. He led the work on radio awareness in the ICT FP7 Project QoSMOS and the H2020 Speed5G Project. He has also led several EU-funded ICT projects. His research interests include cognitive networks, the IoT deployments, sensor data based on knowledge generation, and reconfiguration and resource management for reliable wireless communication.