Blockchain-Enabled Resource Trading and Deep Reinforcement Learning-Based Autonomous RAN Slicing in 5G

Gordon Owusu Boateng, Daniel Ayepah-Mensah, Daniel Mawunyo Doe, Abegaz Mohammed, Guolin Sun, *Member, IEEE*, and Guisong Liu

***Abstract*—The advent of radio access network (RAN) slicing is envisioned as a new paradigm for accommodating different virtualized networks on a single infrastructure in 5G and beyond. Consequently, infrastructure providers (InPs) desire virtualized networks to share their subleased resources for effective resource management. Nonetheless, security and privacy challenges in the wireless network deter operators from collaborating with one another for resource trading. Lately, blockchain technology has received overwhelming attention for secure resource trading thanks to its security features. This paper proposes a novel hierarchical framework for blockchain-based resource trading among peer-to-peer (P2P) mobile virtual network operators (MVNOs), for autonomous resource slicing in 5G RAN. Specifically, a consortium blockchain network that supports hyperledger smart contract (SC) is deployed to set up secure resource trading among seller and buyer MVNOs. With the aim of designing a fair incentive mechanism, we model the pricing and demand problem of the seller and buyers as a two-stage Stackelberg game, where the seller MVNO is the leader and buyer MVNOs are followers. To achieve a Stackelberg equilibrium (SE) for the formulated game, a dueling deep Q-network (Dueling DQN) scheme is designed to achieve optimal pricing and demand policies for autonomous resource allocation at negotiation interval. Comprehensive simulation results analysis prove that the proposed scheme reduces double spending attacks by 12% in resource trading settings, and maximizes the utilities of players. The proposed scheme also outperforms deep Q-Network (DQN), Q-learning (QL) and greedy algorithm (GA), in terms of slice and system level satisfaction and resource utilization.**

***Index Terms*—Blockchain, deep reinforcement learning, network slicing, resource trading, stackelberg game, 5G.**

I. Introduction

THE EMERGING fifth generation (5G) and beyond (B5G) technologies have gained immense attention in recent years [1]. 5G in particular, is envisioned to improve on the legacy fourth generation (4G) technology in terms of network performance, service interoperability, network customization, and flexibility in resource management. Wireless network virtualization (WNV) is tasked with the abstraction, slicing, and isolation of network resources (and infrastructure in some cases), and presenting each partition of the resources (and infrastructure) to users as differentiated services to satisfy their varied quality of service (QoS) requirements. In network slicing, logical partitions of the substrate network infrastructure and resources are entrusted to different service providers (SPs), who independently manage the utilization of such resources in an efficient manner. Software defined networking (SDN) and network function virtualization (NFV) paradigms are the main enablers of WNV [2].

In spite of the overarching merits of network slicing, there are some critical challenges that need to be addressed in radio access network (RAN) research, the most obvious being resource allocation and isolation. Firstly, customization of the virtual networks to provide differentiated services introduces trust and secure interoperability concerns due to possible data leakages [3]. This may result in network interference, and overall performance degradation. Secondly, network partitioning and dynamic spectrum sharing (DSS) in wireless networks have security concerns emanating from the non-transparent and untrusted wireless network environment. Network operators would be unwilling to share their resources with other operators since there is no proper auditability of spectrum usage. Thirdly, resource providers in an attempt to maximize profit are compelled to over-commit the same resources to more than one buyer, assuming all buyers could not utilize the same resource at the same time. This is coupled with double spending attack, which results in deteriorated network performance. Finally, in conventional network slicing procedure, a centralized controller aggregates sensitive information of operators, e.g., slice identifier, location, etc., for resource slicing and trading control. Here, data security and privacy of the operators in the network are undermined. Also, the centrality of the controller increases communication overhead, escalating network latency.

Manuscript received March 25, 2021; revised July 14, 2021; accepted September 16, 2021. Date of publication October 29, 2021; date of current version March 11, 2022. This work is supported by the National Natural Science Research Foundation of China, Grant No. 61771098, by the Fundamental Research Funds for the Central Universities under Grant No. ZYGX2018J068, by the fund from Intelligent Terminal Key Laboratory of Sichuan Province under Grant No. SCITLAB-200025 and by the fund from the Department of Science and Technology of Sichuan Province, Grant No. 2020YFQ0025. The associate editor coordinating the review of this article and approving it for publication was K. Xue. *(Corresponding author: Guolin Sun.)*

Gordon Owusu Boateng, Daniel Ayepah-Mensah, Daniel Mawunyo Doe, and Guolin Sun are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: guolin.sun@uestc.edu.cn).

Abegaz Mohammed is with the School of Mathematics, Zhejiang Normal University, Jinhua 321004, China.

Guisong Liu is with the School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 610074, China.

Digital Object Identifier 10.1109/TNSM.2021.3124046

Traditional security solutions in 4G such as automatic repeat request (ARQ), cannot stand the test of time in 5G since the security issues surpass data integrity. Blockchain technology with key characteristics of immutability, decentralization, transparency, security, and privacy, has been discovered as a potential solution to the above-mentioned challenges. Blockchain is a distributed public ledger, which was first implemented in Bitcoin to transfer cryptocurrency from one entity to another [4]. The concept of blockchain is based on a peer-to-peer (P2P) exchange network where transaction records are stored in tamper-resistant blocks, broadcast through the whole network and each node has a copy of the transaction in its database. Blockchain can be tailored for 5G via its deployment on the wireless network to securely manage virtual networks. To this end, blockchain has been considered for 5G network management in guaranteeing secure resource trading, efficient dynamic resource allocation, and improved network performance [5].

Many researchers have attempted to propose solutions for secure resource trading in 5G RAN slicing via blockchain technology. Some authors designed a brokering mechanism where a network slice broker (NSB) was deployed as an intermediary between vertical SPs and resource providers for secure sub-slice creation in 5G network [6]. In this work, all information about the entities involved is known only by the NSB (centralization). The work in [7] presented a blockchain-based mobile virtual network operators (MVNOs) creation via secure and transparent WNV technique. The proposed method solves the double spending attack problem and reduces business friction. However, there is no proper blockchain-related techno-economic autonomous network slicing procedures involved. Taking fluctuating network traffic into account, reinforcement learning (RL) [8] is preferred to traditional optimization schemes for solving resource allocation problems in RAN. The works in [9], [10] considered blockchain and DRL for secure and flexible content caching and computation offloading, respectively. However, their research interest was prioritized for physical (user-level) resource allocation, without considering slice-based dynamic resource allocation for changing network traffic.

Although these literatures have considered blockchain and DRL for secure resource sharing and RAN slicing separately, there still remain the following research gaps to be addressed: (1) How fair, trusted and secure is the centralized NSB between buyers and sellers? (2) Beyond the economic aspect of resource trading, what technical dynamic resource sharing procedures can be applied for efficient resource management of the limited network resource? (3) What network performance gains will the joint integration of blockchain, WNV, and DRL offer 5G? To the best of our knowledge, this is the first work that well bridges these three technologies in RAN.

In this paper, we leverage blockchain technology, WNV, and DRL to propose a techno-economic hierarchical framework for secure resource trading and intelligent dynamic resource allocation in 5G RAN. We present a permissioned blockchain with hyperledger smart contract (SC) for resource negotiation and renegotiation involving the resource owner and buyers. We model the interaction among buyers and the seller as a two-stage Stackelberg game [11] and formulate a pricing and demand problem for the seller and buyers, respectively. In order to arrive at a Stackelberg equilibrium (SE), we propose a model-free advanced DRL method with a multi-objective reward to update the rule of the stochastic policy without prior knowledge of the environment. The proposed DRL-based scheme seeks to maximize the trading utilities of the players while autonomously adjusting resources of customized SPs to ensure efficient resource utilization and acceptable QoS satisfaction levels. Then, we implement a base station (BS)-level resource update to reflect slice-level allocation. The main contributions of this paper are summarized as follows.

1) We propose a hierarchical blockchain-DRL framework for secure resource trading and autonomous resource slicing in 5G RAN.
2) We deploy a consortium blockchain network that supports hyperledger trading SC to ensure security and transparency in resource trading between a single buyer and multiple seller SPs.
3) We formulate a two-stage Stackelberg game among the multiple buyers and single seller based on demand and pricing strategies. The seller serves as the leader to determine its resource price first, and the buyers serve as followers to determine their individual demands in the game.
4) We design a novel improved DRL technique with a multi-objective reward function to achieve optimal pricing and demand policies for dynamic resource management. We seek to maximize the utilities of resource buyers and the seller, while balancing QoS satisfaction and resource utilization with respect to slices' constraints.

The remainder of the paper is structured as follows: Section II presents the related works, and Section III covers the system model. In Section IV, we discuss the blockchain-enabled resource trading, Stackelberg game formulation, and DRL-based optimal pricing and demand policies for resource management. Experimental results and analysis are presented in Section V. Finally, we conclude this work in Section VI.

II. Related Works

Recently, WNV technique has been proposed to orchestrate the partitioning of network infrastructure and resource in 5G RAN [2], [12]. Vlachos *et al.* [13] improved on conventional RAN slicing schemes by investigating inter-slice device-to-device (D2D) resource sharing. A centralized cross-tenant controller was deployed in the network to ensure appropriate allocation of resource to D2D users of different operators. There emerges a limitation of whether the centralized controller is a trusted entity in the network, which was beyond their scope. An extensive survey on integrating blockchain technology in 5G and beyond networks have been presented in [5].

The integration of blockchain and network slicing is expected to unleash the full potential of 5G. Zanzi *et al.* [14] proposed an NSB method that capitalizes on an intermediate broker and SC to sublease resources from infrastructure providers (InPs) to tenants in a secure and automated manner. The authors in [6] deployed an NSB as an intermediary between SPs and resource providers for sub-slice creation, in a 5G network. However, the centrality of the NSB increases communication overhead, and questions fairness. A multi-operators spectrum sharing (MOSS) SC built on consortium blockchain was designed for resource trading, eliminating the need for a spectrum broker [15]. Rawat investigated the gains of combining three emerging technologies; SDN, edge computing (EC), and blockchain on WNV [16]. The same authors capitalized on the benefits of blockchain to create MVNOs using secure and transparent WNV technique [7]. Based on dynamic configuration, primary wireless resource owners (PWROs) were forbidden to over-commit their resources (stops double spending), and the QoS requirements of users in MVNOs were achieved. In the face of techno-economic resource slicing procedures, the authors fail to provide vivid details on autonomous resource allocation in a secure wireless environment.

Game theory has emerged as a tool for thoroughly modeling the interactions among competing resource providers and requesters in a trading environment, presenting an avenue to complement current research. Qiu *et al.* [17] jointly optimized spectrum price and spectrum amount in a blockchain-enabled spectrum trading framework between primary mobile network operators (MNOs) and unmanned aerial vehicle (UAV)-assisted cellular networks by solving a Stackelberg game. The work mainly focused on the business aspect of spectrum trading involving cost minimization and SC. The authors in [11] proposed a two-stage Stackelberg game model for dynamic pricing-based resource allocation. Each player in the game seeks to optimize their strategies to achieve high utility. In order to maximize users' utilities and MVNOs' revenue, the work in [18] formulated a two-stage Stackelberg game.

Recent advances in RL show its ability to learn the stochastic policy of a dynamic environment without prior knowledge. In [19], resource trading in blockchain-based industrial Internet of Things (IIoT) was discussed. A Stackelberg game was formulated for the interaction between cloud providers and miners. Using DRL, a Nash equilibrium (NE) was established for the resource management and pricing problem. The work in [20] applied RL technique to solve the incentive mechanism problem for multiple providers and multiple Internet of Things (IoT) devices' computing service trading. In [9], blockchain and AI were jointly considered to propose a secure and intelligent architecture for 5G and beyond wireless networks. Specifically, blockchain-based caching problem was formulated to maximize system utility using DRL for D2D cache matching and dynamic bandwidth allocation. This literature was limited to user-level resource allocation, without considering the diverse QoS requirements of different services.

Motivated by the above-mentioned limitations, our work seeks to exploit the properties of blockchain, WNV and DRL for secure, trust-based resource trading and dynamic resource allocation in 5G RAN. We present a consortium blockchain-based secure resource block (RB) trading between the resource owner and buyers for real-time autonomous resource slicing according to the changing traffic conditions.

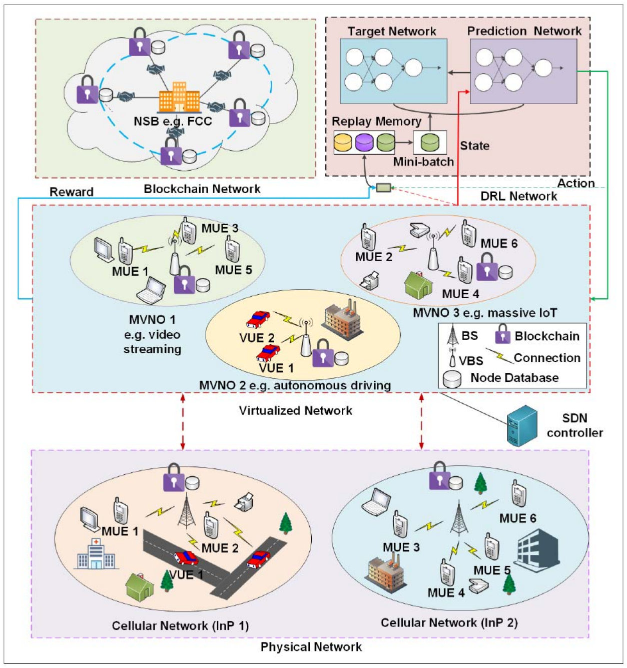


Fig. 1. System architecture.

III. System Model

The system architecture is made up of four constituents namely; physical network, virtualized network, blockchain network, and DRL network. The following entities co-exist in the system; users, virtualized operators (slices), BSs, SDN controller, NSB, radio resources, and DRL agent. Each user reports its channel and state information to the BSs through the SDN controller for proper admission control, user association, and efficient resource allocation. The NSB serves as a trusted authority that sits on top of the network and regulates the blockchain platform. Here, the NSB could be a regulatory body such as the government, or the federal communications commission (FCC). Radio resources are defined as bandwidth with granular units of RBs. We decentralize the SDN controller by linking it to all entities in the blockchain network. Thus, every information on the controller is copied to all the other entities. In this way, external attacks on or single point of failure of the SDN controller may not result in data loss [21]. Since the DRL agent is deployed on the SDN controller, they are both automatically decentralized. The DRL agent selects the best actions from network observations to achieve efficient pricing and demand policies for autonomous resource management.

A complete description of the system architecture as shown in Fig. 1 is presented as follows: We assume the substrate physical infrastructure, who are cellular networks have been abstracted and partitioned into virtualized networks called slices, to be managed independently by MVNOs. The MVNOs provide differentiated services to users associated with their respective slices. For autonomous resource management, the DRL agent observes the fluctuations in network traffic, and suggests an effective resource balancing among the various virtual operators. To realize this balance, a decentralized blockchain network for secure and transparent resource trading among the virtualized operators is deployed in the system. All the nodes with their public keys, private keys and transaction databases are linked to the NSB for certification, authentication and regulation. In this way, the centrality of the NSB is annulled, unlike in [6]. In case of an attack on the NSB, keys cannot be forged because entities participating in resource trading use changeable public keys in order to hide their identity or private information from other participants.

A. Business Model

We design a two-layer hierarchical business model of which InPs sublease resources to MVNOs, who manage their own networks. Let $\mathcal{J} = \{1, 2, \ldots, J\}$ represent a set of InPs, each owning a BS. We assume a set of MVNOs $\mathcal{M} = \{1, 2, \ldots, M\}$ as slices. In this sequel, we refer to MVNOs who provide resources in exchange for revenue as sellers, and MVNOs who buy resources as buyers. We represent a set of seller MVNOs as M^{sell} and a set of buyer MVNOs as M^{buy}, i.e., $M^{buy} + M^{sell} = \mathcal{M}$. Due to the untrusted transmission and broadcast features in 5G wireless environment, real-time efficient resource management among MVNOs via resource trading may suffer privacy leaks, double-spending attacks, or overall system insecurity. For instance, MVNOs are entreated to share their private information with the regulator for resource trading activities [16]. To resolve this situation, we deploy a blockchain network to provide a decentralized, transparent and trusted resource trading environment among MVNOs who want to trade resources. Although resource trading is among MVNOs, the InPs are concerned with the efficient utilization of their scarce physical resources and would want to participate in resource trading activities. Therefore, the blockchain network is made up of an NSB, and block managers/BSs of InPs and MVNOs.

We deploy a consortium blockchain platform, where each MVNO m requires a unique identification address ID_m for identity authentication to be able to participate legitimately in resource trading. An MVNO creates a blockchain account for resource trading and acquires a private key SK_m, a public key PK_m, and a wallet address ADD_m [22]. The private key will be used to digitally sign transactions and the public key is for node identification, when other nodes desire to verify the transaction coming from a specific MVNO. The wallet address will be used to realize resource trading transactions. Also, the MVNO account should contain an account balance BAL_m. Therefore, the transaction details of MVNO m is expressed as a tuple $\{ID_m, SK_m, PK_m, ADD_m, BAL_m\}$. Likewise, the transaction details of InP j is $\{ID_j, SK_j, PK_j, ADD_j, BAL_j\}$.

B. Virtualization Model

The physical network is abstracted and partitioned into multiple virtualized networks as slices, by WNV technique. Based on 5G specifications, we assume three virtualized networks as SPs with versatile QoS requirements, as enhanced mobile broadband (eMBB), ultra reliable low latency communications (uRLLC), and massive machine type communications (mMTC) application services. In each slice, an association manager assigns users among the BSs of InPs based on aggregated demand, and traffic distribution of slices. The classification of flows into slices occurs by mapping the QoS classes of individual flows to the QoS classifier index (QCI) table [23]. A DRL agent deployed on the SDN controller monitors the states of the slices and has an overview knowledge of the existing slices, their available resources, and which flow belongs to which slice. The DRL agent gathers the resource utilization and QoS satisfaction information of the slices at decision steps to ensure dynamic effective resource management via secure resource trading. In an attempt to ensure performance isolation, the virtualized resources of slices are mapped to physical RBs to be allocated to users in the various slices.

C. Network Model

We consider a number of cellular networks with a set of users $\mathcal{I} = \{1, 2, \ldots, I\}$ randomly distributed in their coverage areas, and are connected to the BSs. The total number of users in a specific slice is denoted by $\mathcal{I}_m$ whiles i_m represents a single user in a slice.The overall physical resource pool of the InPs is in bandwidth of $\mathcal{W}$ MHz, which comprises $\mathcal{B}$ RBs each having a bandwidth w_b kHz in time and frequency domains. At each transmission time interval (TTI), an assignment decision is made and each user is assigned a certain amount of RBs. The transmit power of BS *j* is $\mathcal{P}_j$ watts, and the number of RBs allocated to user *i* is b_i. We assume a channel state information (CSI)-aware RB allocation, where co-channel interference can be controlled by applying powerful inter-cell interference cancelation (ICIC) techniques [24]. The average achievable data rate of user *i* in slice *m* associated with BS *j* is calculated as;

$$r_{i_m,j} = \mathcal{W} \cdot \log_2(1 + \phi(i_m, j)) \tag{1}$$

where $\phi(i_m, j)$ denotes the signal-to-interference-noise ratio (SINR) of user *i* in slice *m* associated with BS *j*. For fractional allocation of resource, the achieved data rate of user *i* in slice *m* is expressed as;

$$r_{i_m} = \sum_{j \in J} b_{i_m,j} \cdot r{*}_{i_m,j} \tag{2}$$

where $r{*}_{i_m,j}$ is the normalized average achievable data rate with respect to packet size $\mathcal{L}_i$, and $b_{i_m,j}$ is the amount of RBs allocated to user *i* belonging to slice *m* on BS *j*. Based on M/M/1 queuing theory [25], the average delay experienced by a QoS packet in the queue to user *i* in slice *m* at BS *j* is;

$$\tau_{i_m,j} = \frac{1}{r_{i_m} - \lambda_{i_m}} \tag{3}$$

where r_{i_m} is the normalized user achievable rate with respect to average packet length, and λ_{i_m} is the packet arriving rate of the user in packets per second.

D. Utility Model

1) QoS Satisfaction: We define QoS satisfaction on either data rate or delay, depending on the user satisfaction requirements on the QCI table. QoS satisfaction is defined in the range of [0, 1], where a value of 0.5 and above indicates that the user is satisfied, and a value below 0.5 indicates otherwise. For user *i* in slice *m* with satisfaction priority on data rate, the QoS satisfaction is modeled as a sigmoid function and is expressed as [26];

$$\xi(r_{i_m}) = \frac{1}{1 + e^{-\eta\left(r_{i_m} - r_{i_m}^{min}\right)}} \tag{4}$$

where η is a constant that determines the shape of the satisfactory curve, and $r_{i_m}^{min}$ is the minimum data rate requirement of user *i* in slice *m*. The QoS satisfaction on data rate of slice *m* is the aggregated data rates of users in the slice, which is calculated as;

$$\xi(r_m) = \sum_{i=1}^{I} \xi(r_{i_m}) \tag{5}$$

For user *i* in slice *m* with satisfaction priority on delay, the QoS satisfaction on delay is defined as;

$$\xi(\tau_{i_m}) = \frac{1}{1 + e^{-\eta\left(\tau_{i_m}^{max} - \tau_{i_m}\right)}} \tag{6}$$

where $\tau_{i_m}^{max}$ is the maximum tolerant delay requirement of user *i* in slice *m*, which is required to satisfy the upper bound delay for user *i*. The satisfaction on delay of slice *m* is the sum of the delay experienced by users in the slice, and is expressed as;

$$\xi(\tau_m) = \sum_{i=1}^{I} \xi(\tau_{i_m}). \tag{7}$$

2) Resource Utilization: We define resource utilization of a slice as a ratio of the number of RBs occupied by users in the slice, to the total number of RBs allocated to the slice. The resource utilization of slice *m* can be calculated as;

$$\psi_m = \frac{\varphi_m}{b_m} \tag{8}$$

where φ_m is the amount of RBs occupied by slice *m*, and b_m is the number of RBs allocated to slice *m*.

3) Trading Utility Function: In order to obtain extra RBs to serve its customers due to increase in user population, a buyer MVNO needs to pay for the extra RBs it occupies from the seller MVNO. This is the revenue the seller receives for granting the buyer access to utilize its idle RBs. Considering RB demand $d_{m^{buy}}$ of an arbitrary buyer MVNO, and the unit price δ_b of an RB, the utility of m^{buy} is calculated as;

$$\mathcal{U}_{m^{buy}}(d_{m^{buy}}, \delta_b) = \mathcal{R}(d_{m^{buy}}, \delta_b) - \mathcal{C}(d_{m^{buy}}, \delta_b) \tag{9}$$

where $\mathcal{R}(d_{m^{buy}}, \delta_b)$ is the revenue gained from the resale of purchased RBs, and $\mathcal{C}(d_{m^{buy}}, \delta_b)$ is the cost involved in purchasing the RBs from m^{sell}. Generally, the cost increases exponentially with increasing number of purchased RBs. The cost function is expressed as;

$$\mathcal{C}(d_{m^{buy}}, \delta_b) = d_{m^{buy}} \cdot \delta_b \tag{10}$$

Taking the cost into consideration, buyers should account for the real demand of heir users when purchasing RBs from the sellers. The revenue can further be modeled as a logarithmic function as [17];

$$\mathcal{R}(d_{m^{buy}}, \delta_b) = \log_2\left(1 + \frac{b_{m^{buy}}}{d_{m^{buy}}}\right). \tag{11}$$

It can be observed from (11) that, the higher the number of allocated RBs, the higher the revenue. Likewise, the utility of seller MVNO m^{sell} is the revenue obtained from selling RBs to m^{buy} neglecting operational costs, and is calculated as;

$$\mathcal{U}_{m^{sell}}(\delta, d) = \mathcal{R}(\delta_b \cdot d_{m^{buy}}). \tag{12}$$

IV. Problem Formulation

At negotiation intervals, MVNOs can negotiate resource allocation updates based on needs, following traffic fluctuations. By resource trading results, the resources of MVNOs are readjusted depending on who needs resources and who needs revenue.

A. Consortium Blockchain-Based Resource Trading

Consortium blockchain is a hybrid of public and private blockchains, which provides authorization and solves the problem of monopoly. We prefer consortium blockchain to public blockchain and private blockchain because, it offers high throughput, low latency, and high level of security in resource sharing environments. Also, the mining costs in consortium blockchain are less since it does not involve processing fees, and it is not computationally expensive to publish new blocks [21]. The actors in the consortium blockchain network are the MVNOs (buyers and seller), and the InPs. Resources are allocated to buyers based on the execution of the SCs. Hyperledger [27] is used to execute the SCs, hence the specific name hyperledger Iroha SC. Hyperledger Iroha SCs are lines of code that are stored on a blockchain platform to automatically execute, when predefined terms and conditions are met. The SC is preferred to conventional service level agreement (SLA) due to its benefits of speed, accuracy, and trust. The detailed operation mechanism of our proposed consortium blockchain is presented below.

1) System Initialization: As consortium blockchain is permissioned, nodes are required to register with the regulator to become legitimate participants of the network. We utilize elliptic curve digital signature algorithm, and asymmetric cryptography [28] for system initialization. After creating a blockchain account, the regulator (NSB) issues a certificate *Cert* to each node that qualifies it to be given unique transaction details in the tuple {*ID*, *SK*, *PK*, *ADD*, *BAL*}. The asymmetric cryptography of information integrity from the sender *x* during system initialization is expressed mathematically as;

$$Dec_{PK_x}\left(Sgn_{SK_x}(Hash(msg))\right) = Hash(msg) \tag{13}$$

where Sgn_{SK_x} denotes the digital signature of the sender *x* with its private key *SK*, Dec_{PK_x} is used to decode the signed data with the sender's public key *PK*, and *Hash*(*msg*) is the hash digest of the message [29].

2) Reputation-Based Verifier Selection: We assume that not all the nodes in the blockchain network are honest enough to take part in block verification and audit. Therefore, we apply a reputation-based verifier selection based on a subjective logic model as used in [22]. Subjective logic model defines a trust threshold agreed on by all nodes. With respect to previous verification and audit records of the nodes involved, all nodes express their opinions on other nodes by voting them as *'honest'* or *'dishonest'*. After the voting process, a selection criterion is enforced using the formulation;

$$rep_{m/j} > rep_{thr} \tag{14}$$

where $rep_{m/j}$ denotes the reputation of MVNO *m* or InP *j*, and rep_{thr} is the minimum reputation threshold. The *'honest'* nodes are then selected for the block verification and audit phase.

3) Resource Trading Between Buyer and Seller: An MVNO in a bid to adjust its slice resource, broadcasts a request Req_m throughout the decentralized blockchain network. All nodes involved in the SC on the blockchain platform must agree on the rules that govern the transactions and explore all possible exceptions. Each buyer and seller seeks to maximize their demand and pricing strategies, respectively through an interest competition game. Depending on the outcome of the game, the NSB matches the buyer to the right seller and the SC arbitrates. The seller then leases the necessary RBs to the buyer for slice resource allocation, and the buyer pays for the resource.

4) Block Generation and Broadcast: After successful resource trading, the nodes in the blockchain network select a leader node *n* to create a block for the transaction. Based on the *"opinion"* of each entity, the node with the highest vote becomes the leader for the block creation of the particular transaction until its acceptance onto the ongoing chain. The elected leader records the transaction in a tamper-resistant block, encrypts, and digitally signs on it to guarantee block authenticity. Then, the leader broadcasts the block to all the nodes in the network for verification and audit.

5) Block Verification and Audit With Consensus Mechanism: In our consortium blockchain network, we deploy a lightweight practical Byzantine fault tolerant (PBFT) [30] consensus mechanism for approving blocks. In PBFT, there is no need for extensive block mining like in Bitcoin's proof of work (PoW), ensuring higher energy efficiency. Also, PBFT is recommended for small-scale networks with fewer nodes so that every pre-selected verifier node is incentivized. The pre-selected nodes check for the correctness of the block, and report their audit results to the leader for analysis. The leader analyzes the audit results of the verifying nodes and accepts the block, if consensus is reached. If the correctness of the block is approved by all verifiers, the leader broadcasts the accepted block, and the new block is added to the ongoing chain, which contains the hash of the previous block. Algorithm 1 describes the RB trading steps between MVNOs.

Algorithm 1 Consortium Blockchain-Based RB Trading

```
Initialization: Number of InPs J, $(M^{sell}, M^{buy})$, NSB
Registration: Register InPs and MVNOs
for period $p$ do
    */Stackelberg game for trading and block creation/*
    for all M, J, and $NSB$ do
        Verify $Cert_m$ and $Cert_j$ using batch verification
        if $Ver(Cert_m, Cert_j) = True$ then
            Set-up a two-stage Stackelberg game based on
            Eqns. (15) and (17)
            Execute SC for trading and create block
        else
            Terminate SC
        end if
    end for
    */ Block verification and PBFT consensus /*
    for miner leader $n$ do
        Broadcast the current BLK data to all nodes
        for all miner nodes do
            if miner Tx data = BLK data then
                set verify BLK = True
            else
                set verify BLK = False
            end if
            Broadcast audit results to the leader n for
            analysis
        end for
        Accept and add block to ongoing chain or discard
        block that fails verification
    end for
end for
```

B. Two-Stage Stackelberg Game Formulation

For resource exchange between the seller and buyers, game theory is the most suitable for seller-buyer matching [31]. We consider a Stackelberg game where resource buyers and the resource seller are rational entities, i.e., each player selects its own strategy to maximize its utility given other players' strategies. We propose a two-stage Stackelberg game with a single seller as the leader, and multiple buyers as the followers to design an incentive mechanism for efficient resource trading. Thus, the seller sets the unit price of its resource first. Then, the buyers consider the unit price set by the seller to make a matching decision. The leader needs to find an optimal price δ to maximize its utility within its available resources. Similarly, each follower makes a decision to maximize its utility while achieving a desired QoS. Both the leader and followers can constantly adjust their strategies to earn more profit.

The two-stage Stackelberg game is formulated below.

Stage 1 (Leader's Pricing):

$$\underset{\delta \geq 0}{max} \ \mathcal{U}_{m^{sell}}(\delta, d) \tag{15}$$

$$s.t. \ \sum_{m=1}^{M} d_{m^{buy}} \leq \mathcal{B} \tag{16}$$

where $\mathcal{U}_{m^{sell}}(\delta, d)$ is the utility function of the seller MVNO, δ is the unit RB price vector with $[\delta_1, \delta_2, \ldots, \delta_M]^T$, and *d* is a vector of RBs demand of buyers with $[d_1, d_2, \ldots, d_M]^T$. The expected revenue of a leader is defined in (12). By observing the pricing strategy of the seller, the buyers set their demand to earn more profit.

Stage 2 (Followers' Demand):

$$\underset{d_{m^{buy}} \geq 0}{max} \mathcal{U}_{m^{buy}}(d_{m^{buy}}, \delta_b) \tag{17}$$

where $\mathcal{U}_{m^{buy}}(d_{m^{buy}}, \delta_b)$ is the utility function of the buyer MVNO. The expected revenue of a follower is defined in (9).

Equations (15) and (17) form the Stackelberg game with the objective of finding an SE, where neither a leader (m^{sell}) nor a follower (m^{buy}) has incentive to deviate. The SE is defined as follows.

Definition 1 (Stackelberg Equilibrium): Let (δ^*, $d_{m^{buy}}{}^*$) be the SE, where δ^* is the solution for RB pricing and $d_{m^{buy}}{}^*$ is a solution for the RB demand problem. Then, the point (δ^*, $d_{m^{buy}}{}^*$) holds if for any (δ, d) with $\delta \geq 0$ and $d \geq 0$,

$$\mathcal{U}_{m^{sell}}(\delta^*, d^*) \geq \mathcal{U}_{m^{sell}}(\delta, d^*) \tag{18}$$

$$\mathcal{U}_{m^{buy}}(d_{m^{buy}}{}^*, \delta^*) \geq \mathcal{U}_{m^{buy}}(d_{m^{buy}}, \delta^*) \tag{19}$$

Taking the second order derivatives of the utility function of buyers with respect to *d* and utility function of the seller with respect to δ, we can verify the uniqueness and existence of the SE in our formulated game. It is proven in [17] that backward induction can be used to achieve SE for $\mathcal{U}_{m^{sell}}$ and $\mathcal{U}_{m^{buy}}$. However, this method of finding SE requires all nodes to disclose their private information. More so, this practice may affect the fairness of the game. In contrast, DRL approach learns the optimal policy without prior knowledge. Therefore, we design a DRL-based method for obtaining optimal pricing and demand strategies for autonomous resource management.

C. DRL-Based Optimal Pricing and Demand Policies for Resource Management

1) MDP Formulation: We formulate the optimal pricing and demand problem as an MDP by defining states, actions, and rewards. MDP can be expressed as a stochastic process in the tuple $(S, A, P(s'/s, a), R, S')$, where *S* represents the state set and *A* represents the action set. $P(s'/s, a)$ is the state transition probability, *R* is the reward and S' denotes the next state. A detailed MDP formulation for slice resource allocation is presented in our prior work [32]. From the Markov property, the policy π can be expressed as;

$$\mathcal{V}^{\pi}(s)=\mathbb{E}_{\pi}\left\{r\left(s^t, a^t\right) + \gamma \sum_{s'} P\left(s'|s^t, a^t\right)\mathcal{V}^{\pi}\left(s'\right)\right\}, \tag{20}$$

where $r(s^t, a^t)$ is the present reward, $\mathcal{V}^{\pi}(s)$ is the present utility, and $\mathcal{V}^{\pi}(s')$ is the future utility. The state-value function for an optimal policy based on the Bellman equation [8] is given as;

$$\mathcal{V}^{\pi^*}(s) = arg \underset{a^t \in A}{max}\{\mathcal{V}^{\pi}(s)\}. \tag{21}$$

2) State-Action Mapping: Due to the varying mobility of the wireless network environment, the state information of the slices change at time intervals. To cope with such mobility, a DRL agent deployed on the SDN controller observes each slice's state and suggests optimal pricing and demand prediction policies for resource allocation among buyer and seller MVNOs.

State(s): We define the state s_m^t of slice *m* at decision step *t* as the tuple $s_m^t = \{\xi_m^t, \psi_m^t, \mathcal{U}_m^t\}$, where ξ_m^t denotes the QoS satisfaction of slice *m*, ψ_m^t denotes the resource utilization of slice *m*, and $\mathcal{U}_m^t$ is the trading utility of slice *m*. We recall that slice *m* could be m^{buy} or m^{sell}. The form of $\mathcal{U}_m^t$ of a slice depends on its role in the trading process. That is, m^{sell} adjusts its strategy to maximize its unit price and m^{buy} seeks to maximize its profit. The learning agent outputs a suitable action to be selected based on the input states.

Action(a): At each state s_m^t, the learning agent performs an action a_m^t based on the observations. The set of possible actions the DRL agent selects includes the demand action $(d_{m^{buy}})$, and the unit price action $(\delta_{m^{sell}})$. At time step *t*, let $d^t \in a_m^t$ and $\delta^t \in a_m^t$ denote the demand action set of m^{buy} and unit price action set of m^{sell}, respectively. The buyers and seller take action independently and sequentially. Based on the actions selected by the players to maximize their utilities, the DRL agent adjusts slice resource with a value from the set $\varpi^t \in a_m^t$ as $\varpi^t = \{-0.5, -0.4, -0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The elements of the set ϖ^t are normalized values of percentage increase and decrease of a slice resource. Therefore, a complete action set is of the form $a_m^t = \{d^t, \delta^t, \varpi^t\}$. The selected action is used for BS-level slice resource update.

Reward(r): The DRL agent interacts with the wireless network environment by exploring possible actions and exploiting the potentially best ones. We seek to maximize the trading utility of the slices, while balancing their QoS satisfaction and resource utilization. Therefore, we define the reward function of the DRL scheme as;

$$r_m^t = \left(\alpha \cdot \xi_m^t + \beta \cdot \psi_m^t\right) + \mathcal{U}_m^t \tag{22}$$

The values of α and β are constants, which denote the importance of QoS satisfaction and resource utilization. We present the proposed dueling DQN-based algorithm below.

3) Overview of Dueling DQN: Dueling DQN is made up of a single Q-network with two stream estimator functions; state-value estimator function and state-dependent action advantage function [33]. We choose dueling DQN because, it improves on the convergence of DQN by learning the important states and actions, without causing changes to the underlying DQN algorithm. It can also quickly identify the best action because, it has the ability to learn which states are important to the learning agent without learning the effect of each action for each state. From the policy π, the state-action value pair $\mathcal{Q}^{\pi}(s^t, a^t)$, and the state value $\mathcal{V}^{\pi}(s)$ can be expressed as;

$$\mathcal{Q}^{\pi}\left(s^t, a^t\right) = \mathbb{E}_{\pi}\left\{r\left(s^t, a^t\right), \pi\right\} \tag{23}$$

$$\mathcal{V}^{\pi}(s) = \mathbb{E}_{a \sim \pi(s)}\left[\mathcal{Q}^{\pi}\left(s^t, a^t\right)\right]. \tag{24}$$

From Eqns. (23) and (24), we define the advantage function as;

$$\mathcal{A}^{\pi}\left(s^t, a^t\right) = \mathcal{Q}^{\pi}\left(s^t, a^t\right) - \mathcal{V}^{\pi}(s) \tag{25}$$

where $\mathcal{V}^{\pi}(s)$ denotes how good a state is, $\mathcal{Q}^{\pi}(s^t, a^t)$ denotes the value of choosing an action in a state, and $\mathcal{A}^{\pi}(s^t, a^t)$ describes the importance of an action in a particular state. For an optimal policy $max_{a^t \in A}\mathcal{Q}(s^{t+1}, a^{t+1})$, it follows that

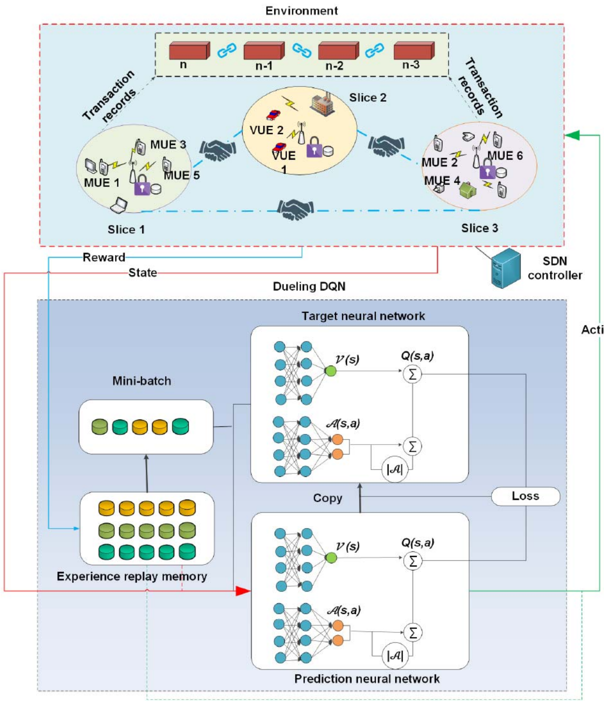


Fig. 2. Dueling DQN framework.

Algorithm 2 Dueling DQN-Based Algorithm

```
Initialization: Set replay memory $D$, action-value $\mathcal{Q}$ with random weights $\partial$ and $\vartheta$, epsilon $\varepsilon$ and Q-table
for each episode do
    Set up the environment
    for each decision step $t$, do
        Observe state $s_m^t = \{\xi_m^t, \psi_m^t, \mathcal{U}_m^t\}$
        /***RB Trading and Resource Allocation***/
        Via blockchain, execute Algorithm 1
        Select a random action $a_m^t$ with probability $\varepsilon$,otherwise, select best action using Eqn. (21)
        Update the resource pool $b_m$ of slice $m$ and observe reward $r_m^t$ and new state $s_m^{t+1}$
        Update the required allocation $b_{m,j}^{t+1}$ at BS-level
        /******Learning Update******/
        Store the tuple ($s_m^t$, $a_m^t$, $r_m^t$, $s_m^{t+1}$) in $D$
        Sample random mini-batch of transitions $(s^j, a^j, r^j, s^{j+1})$ from $D$
        Combine $\mathcal{A}^{\pi}(s^t, a^t)$ and $\mathcal{U}^{\pi}(s)$ using Eqn. (25)
        Set $y^j = r^j + \varepsilon \max_{a^{j+1}} \hat{\mathcal{Q}}(s^{j+1}, a^{j+1}; \theta')$
        Perform a gradient descent step
        Every $C$ steps, reset $\hat{\mathcal{Q}} = \mathcal{Q}$
    end for
end for
```

$\mathcal{Q}^{\pi}(s^t), a^t) = \mathcal{V}^{\pi}(s)$, hence $\mathcal{A}^{\pi}(s^t, a^t) = 0$. The output of the dueling network is given as;

$$\mathcal{Q}\big(s^t, a^t; \varnothing, \partial, \vartheta\big) = \mathcal{V}\big(s^t; \varnothing, \vartheta\big) + \mathcal{A}\big(s^t, a^t; \varnothing, \partial\big) \quad (26)$$

where $\varnothing$ represents common network parameters, ∂ represents advantage stream parameters, and ϑ represents value stream parameters. To solve the issue of identifiability, we generate the Q-values for each action *a* at state *s* using the aggregation layer as follows:

$$\mathcal{Q}\big(s^t, a^t; \varnothing, \partial, \vartheta\big) = \mathcal{V}\big(s^t; \varnothing, \vartheta\big) + \mathcal{A}\big(s^t, a^t; \varnothing, \partial\big) - \frac{1}{|A|} \sum_{a^{t+1}} \mathcal{A}\Big(s^t, a^{t+1}; \varnothing, \partial\Big). \quad (27)$$

After an action is selected to be enforced on the resource pool of a slice, the slice resource pool is updated. The resource update is expressed as;

$$b_m^{t+1} = \begin{cases} b_m^t, & if,\ a_m = 0 \\ (1 + a_m) b_m^t, & otherwise \end{cases} \quad (28)$$

where b_m^{t+1} is the updated resource of slice *m*, b_m^t denotes the amount of resource initially allocated to slice *m*, and a_m is the action enforced on slice *m* at decision step *t*. Algorithm 2 presents the description for dueling DQN-based pricing and demand prediction solution for autonomous resource allocation. The computational complexity of Algorithm 2 is of $O(\mathbb{G}^{\varphi}\mathbb{L}_{\rho})$, where $\mathbb{G}^{\varphi}$ and $\mathbb{L}_{\rho}$ denote the number of hidden layers and number of neurons in the NN.

V. Performance Evaluation

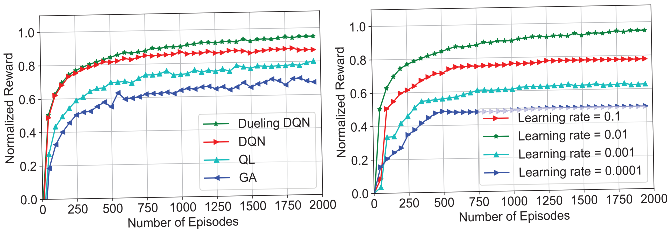
A. Scenario Configuration

In this section, we evaluate the performance of our proposed Dueling DQN algorithm based on extensive simulations and comprehensive analysis of results. All simulations are performed with reference to 5G specifications and standards [3]. We implement the simulations using Python 3.6 environment and Tensorflow 1.13 on a computer with a core i7 CPU running on a processor speed of 2.4GHz, and 16GB RAM. We consider a given coverage area of 500m × 500m with 2 BSs that are 120m apart. The system bandwidth is set to 20MHz with 100 RBs. The BS transmit power is set to 30dBm assuming negligible interference between BSs. Moreover, we consider a log-normal distribution for shadow fading of the channels. With reference to the QCI index table [23], we define 3 slices as eMBB, uRLLC, and mMTC. Due to changing user population, we make use of the random walk mobility model to predict user mobility across the network [34].

To implement the consortium blockchain in our work, we set up a hyperledger Iroha platform [35] with SC on Ubuntu 16.04 LTS Bionic OS. We deploy hyperledger Iroha SC on our blockchain platform for secure, accurate, and quick execution of SLA. The hyperledger Iroha SC is written as a program, often referred to as *'Chaincode'*. We use one chaincode to handle all the contracts to improve its efficiency. When peers submit resource trading requests, the chaincode initializes and manages the transaction ledger states. In the Dueling DQN algorithm, we set the size of the replay memory to 10^5, and minibatch size to 128. The discount factor γ, epsilon-greedy ϵ, and learning rate α are, 0.85, 0.1 and 0.01, respectively to ensure stable performance. The ANN structure consists of an input layer, two hidden layers (32 neurons in each) and an output layer. We utilize ReLU as the activation function for the hidden layers and sigmoid function is employed at the output layer. To optimize the loss function, we use the

TABLE I
SIMULATION PARAMETERS

Parameters and Units	Values
Number of BSs $\mathcal{J}$	2
Number of Slices $\mathcal{M}$	3
System bandwidth $\mathcal{W}$	20MHz
Number of RBs $\mathcal{B}$	100
Transmit power of BS P_j	30dBm
BS coverage radius	500m
Distance between BSs	120m
Noise power density θ^2	-174dBm/Hz
Number of users $\mathcal{I}$	[eMBB:150, uRLLC:100, mMTC:200]
User distribution	Uniform
Packet arriving rate	[eMBB:100, uRLLC:100 ,mMTC:100] packets per sec.
Minimum data rate	[eMBB:500, uRLLC:10, mMTC:15] kbps
Maximum delay	[eMBB:100, uRLLC:10, mMTC:100] ms
Packet size	[eMBB:400, uRLLC:120, mMTC:500] bits
Number of hidden layers	2 (32 neurons in each)
Number of episodes	2000
Discount factor γ	0.85
Replay memory size D	10^5
Mini batch size D'	128
Learning rate α	0.01



(a) Based on reward

(b) Based on learning rates

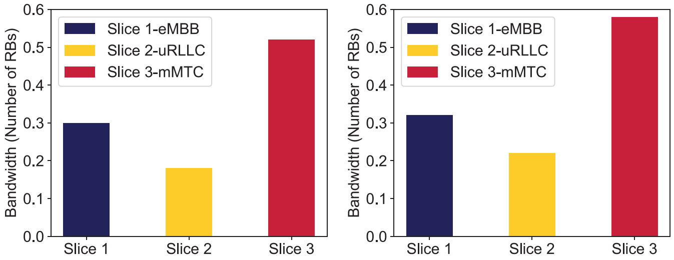
Fig. 3. Convergence analysis.

AdamOptimizer. Otherwise stated, we run all the simulations over 2000 episodes. We summarize the simulation parameters in Table I.

B. Convergence Analysis

In this simulation, we compare the convergence performance of our proposed dueling DQN-based (Dueling DQN) algorithm with greedy approach (GA) and other RL algorithm variants namely; classical Q-Learning (QL) and classical DQN (DQN) based on normalized reward and different learning rates. We run the simulations for 2000 episodes, while taking the maximum value at each 100 episodes for performance comparison. Fig. 3(a) shows the performance of Dueling DQN, DQN, QL and GA with respect to convergence on normalized reward. Fig. 3(b) shows the effect of different learning rates on the convergence of the proposed Dueling DQN algorithm.

From Fig. 3(a), we can observe that all the algorithms achieve convergence, with the proposed algorithm achieving the fastest convergence at about 450 episodes, and the highest normalized reward at approximately 0.95. The reason for this trend is due to the fact that Dueling DQN skips actions that may have no effect on the learning process. The performance of DQN is very close to the proposed Dueling DQN because they both have similar underlying architecture with the difference being the number of streams deployed. Thus; DQN suffers from overestimation of the Q values, thereby slowing its convergence. QL performs the worst among the RL algorithms because of the discretization of the state space. However, GA achieves the worst convergence with the normalized reward at about 0.65. This is because, GA selects actions in a greedy manner. Intuitively, GA is suitable for solving nonlinear integer programming (NLIP) problems, which may be unable to capture the high dynamics in the wireless environment.



(a) No double spending

(b) Double spending

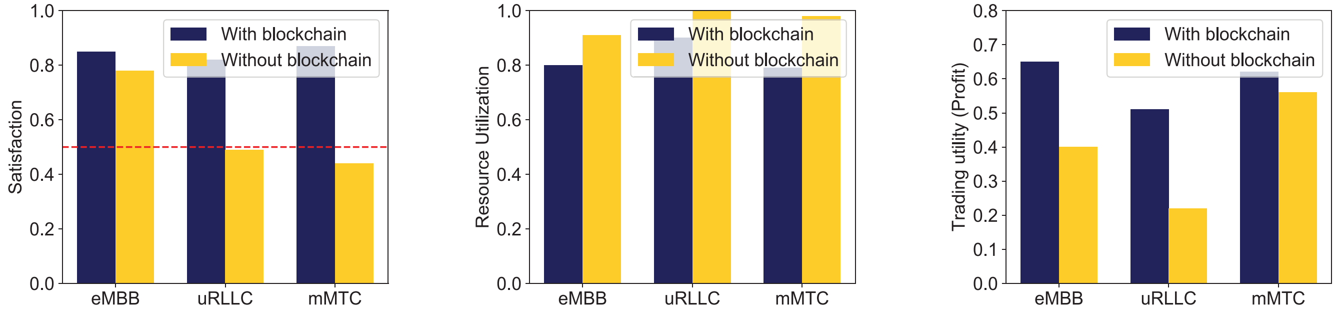
Fig. 4. Impact of double spending.

Since the proposed Dueling DQN algorithm achieved the fastest convergence and the highest reward value, we further evaluate its convergence under varying learning rates. Fig. 3(b) shows that at each learning rate, convergence is achieved, with $\alpha = 0.01$ achieving the highest normalized reward. We can conclude that selecting a high learning rate for our proposed algorithm achieves a better convergence. However, this does not hold for every scenario since the choice of α depends on the algorithm structure and the state space involved.

C. Impact of Double Spending on Slice Resource

In this simulation, we evaluate the effect of allocating the same RBs to more than one MVNO/slice on the overall system bandwidth. This problem in wireless network resource allocation is known as double spending. We convert the system bandwidth of 20 MHz to 100 RBs, and allocate them to the various slices in this experiment. We evaluate the performance of our proposed Dueling DQN scheme with blockchain against a similar scheme without blockchain, in terms of slice resource allocation. Note that, we take a snapshot of one decision cycle for this evaluation. We assume the physical networks do not reserve portions of their resources for their own use. Therefore, 100% of the system bandwidth is being allocated to the various slices. Fig. 4(a) shows the performance of our proposed Dueling DQN with blockchain and Fig. 4(b) shows the performance of Dueling DQN without blockchain, in terms of slice resource allocation.

The results in Fig. 4(a) show that with blockchain, eMBB, uRLLC and mMTC slices are allocated 30, 18, and 52 RBs respectively at a specific slicing period. That is, the total number of allocated resources is equal to the system bandwidth. Conversely, in Fig. 4(b), the total allocated RBs of eMBB, uRLLC and mMTC slices are 32, 22, and 58, respectively, exceeding the total number of RBs in the system by 12%. This implies that 12 RBs have been allocated more than once to

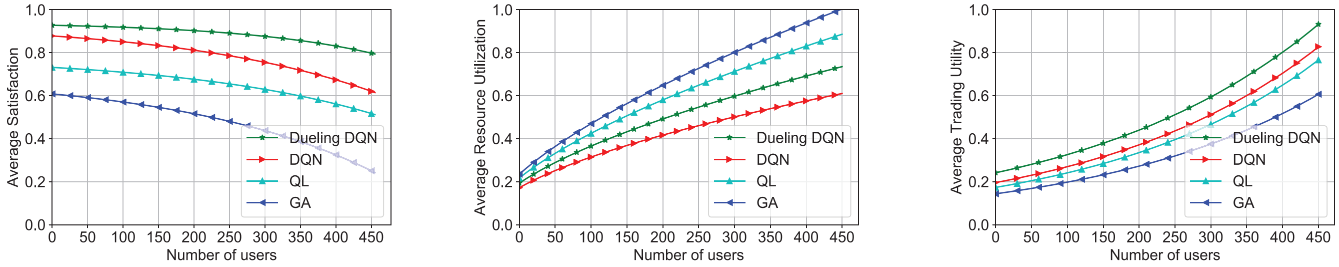


(a) Slice satisfaction.

(b) Slice resource utilization.

(c) Slice trading utility

Fig. 5. Performance on slice level.



(a) System satisfaction.

(b) System resource utilization.

(c) System trading utility

Fig. 6. Performance on system level.

different slices at the specific negotiation interval. We can conclude that the Dueling DQN scheme with blockchain ensures that each RB is allocated once at a specific slicing period and that double spending is prevented.

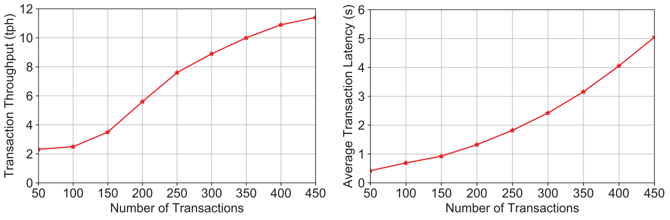
D. Slice-Level Performance Analysis

In this experiment, we analyze the performance of our proposed Dueling DQN algorithm with blockhain in terms of slice QoS satisfaction, resource utilization, and trading utility. We compare the performance of our proposed algorithm with Dueling DQN without blockchain. We recall that the slice QoS satisfaction, resource utilization and trading utility are calculated using (5) or (7), (8), and (9) or (12), respectively. Fig. 5(a), 5(b), and 5(c) show the performance of eMBB, uRLLC, and mMTC slices in terms of slice level satisfaction, resource utilization and trading utility. For simplicity, we consider the slices in the trading utility simulation as buyers only. As seen in Fig. 5(a), the satisfaction levels of the three slices under the proposed algorithm exceed the minimum satisfaction threshold, i.e., 0.5. The eMBB, uRLLC, and mMTC slices achieve satisfaction levels of approximately 0.85, 0.82, and 0.90, respectively. Under Dueling DQN without blockchain, the satisfaction levels decrease in each slice with uRLLC and mMTC slices achieving levels of approximately 0.49 and 0.42 respectively, which are below the minimum satisfaction threshold. From Fig. 5(b), we observe that the resource utilization levels of the three slices under the proposed algorithm are approximately 0.79, 0.90, and 0.78 respectively. However, under Dueling DQN without blockchain, the resource utilization levels of the slices are higher, with uRLLC and mMTC achieving nearly 100% resource utilization levels. The simulation results in Fig. 5(c) depict that under the proposed Dueling DQN with blockchain, the slices are able to maximize their profits better compared with Dueling DQN without blockchain. It can be concluded that the proposed algorithm better balances slice QoS satisfaction and resource utilization, while maximizing profits, compared to dueling DQN without blockchain.

E. System-Level Performance Analysis

In this simulation, we evaluate the system level performance of the proposed Dueling DQN algorithm in terms of system QoS satisfaction, resource utilization and trading utility. We compare our algorithm with DQN, QL and GA. To obtain the whole system's utility metrics, we average the QoS satisfaction, resource utilization and utility of all the slices. We analyze the results from light-load and heavy-load perspectives.

As shown in Fig. 6(a), under light-load scenario, say 50 users, all the four algorithms achieve satisfaction levels above the threshold. The proposed algorithm achieves the highest system satisfaction at about 0.92, followed by DQN, QL, and GA respectively. However, at heavy-load scenario, only Dueling DQN, DQN, and QL algorithms are able to keep their system satisfaction levels above 0.50. With 450 users, GA achieves a system satisfaction level of about 0.22. This implies that some users may not be satisfied in the system at this point. In Fig. 6(b), the average resource utilization of all the four algorithms increase with increasing number of users in the system. At heavy load, i.e., 450 users, DQN achieves the lowest system resource utilization level, followed by the



(a) Throughput analysis

(b) Latency analysis

Fig. 7. Blockchain analysis.

proposed Dueling DQN, QL and GA with the levels at approximately 0.60, 0.72, 0.90, and 1.00. This implies that to achieve satisfaction, GA and QL have to utilize almost 100% of the system resource. From Fig. 6(c), we observe that the two DRL algorithms are able to maximize system profits from purchased RBs better, compared with GA and QL schemes. It can be concluded that both Dueling DQN and DQN can balance system satisfaction and resource utilization at acceptable levels, and can achieve higher profits in terms of trading utility. However, Dueling DQN has a slight edge over DQN.

F. Blockchain Performance Analysis

In this subsection, we seek to confirm the gains of the consortium blockchain algorithm by analyzing its throughput and latency performance. Transaction throughput defines how many transactions the blockchain platform can handle in a period of time (per hour in this experiment). Transaction latency refers to the time taken for a transaction process to be completed (in seconds). Fig. 7(a) and 7(b) illustrate the throughput and latency performance of blockchain in our work. It can be observed from Fig. 7(a) that as the number of transactions increases, the throughput increases. This implies that the blockchain network is not saturated by increasing number of transactions. For instance, in 6 hours, the blockchain network is able to process approximately 200 transactions. This could be due to a large block size, which is always in a capacity to accept transactions. From Fig. 7(b), we can see that the transaction latency increases as the number of transactions increases. This is due to two factors: (i) the time needed to negotiate the resource trading and (ii) the time needed to vote, verify, and commit a transaction using the PBFT consensus algorithm in consortium blockchain. That is, the more the transactions, the more time needed to negotiate and also accept the transaction onto the ongoing chain. This will in turn increase the time needed to process transactions. We can conclude that although the latency increases with increasing number of transactions, the overall throughput and latency performance achieved are at acceptable levels.

VI. Conclusion

This paper proposed a novel blockchain-DRL framework for secure transparent resource trading, and autonomous slice resource allocation in 5G RAN. Based on consortium blockchain that supports hyperledger SCs, we set up resource trading transactions among seller and buyer MVNOs considering demand and pricing matching with a two-stage Stackelberg game formulation. To achieve optimal pricing and demand policies for autonomous resource allocation, we designed an advanced Dueling DQN scheme. Then, we updated the slice resources at the BSs to realize the alteration of previous slice resource allocation. Extensive simulation results confirmed the efficacy of our proposed Dueling DQN algorithm, compared with DQN, QL and GA. Our proposed algorithm can maximize the profit of players, while balancing QoS satisfaction and resource utilization of the slices. The proposed Dueling DQN algorithm with blockchain can also solve the double spending attack problem in 5G wireless network. Future work will consider extensive security analysis, and edge in-network computing for an effective consensus process.

References

[1] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[2] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 462–476, Sep. 2016.

[3] A. Machwe *et al.*, "5G PICTURE, D2.1 5G and vertical services, use cases and requirements, version 2.0," 2018. [Online]. Available: http://www.https://www.5g-picture-project.eu/

[4] S. Nakamoto. "Bitcoin: A peer-to-peer electronic cash system." 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[5] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for 5G and beyond networks: A state of the art survey," *J. Netw. Comput. Appl.*, vol. 166, Sep. 2020, Art. no. 102693.

[6] B. Nour, A. Ksentini, N. Herbaut, P. A. Frangoudis, and H. Moungla, "A blockchain-based network slice broker for 5G services," *IEEE Netw. Lett.*, vol. 1, no. 3, pp. 99–102, Sep. 2019.

[7] D. B. Rawat and A. Alshaikhi, "Leveraging distributed blockchain-based scheme for wireless network virtualization with security and QoS constraints," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, 2018, pp. 332–336.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[9] Y. Dai, D. Xu, S. Maharjan, Z. Chen, Q. He, and Y. Zhang, "Blockchain and deep reinforcement learning empowered intelligent 5G beyond," *IEEE Netw.*, vol. 33, no. 3, pp. 10–17, May/Jun. 2019.

[10] D. C. Nguyen, P. N. Pathirana, M. H. Ding, and A. Seneviratne, "Blockchain as a service for multi-access edge computing: A deep reinforcement learning approach," 2020, *arXiv:2001.08165*.

[11] T. M. Ho, N. H. Tran, S. M. A. Kazmi, and C. S. Hong, "Dynamic pricing for resource allocation in wireless network virtualization: A Stackelberg game approach," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Da Nang, Vietnam, 2017, pp. 429–434.

[12] I. da Silva *et al.*, "Impact of network slicing on 5G radio access networks," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, 2016, pp. 153–157.

[13] C. Vlachos, V. Friderikos, and M. Dohler, "Optimal virtualized intertenant resource sharing for device-to-device communications in 5G networks," *Mobile Netw. Appl.*, vol. 22, pp. 1010–1019, Feb. 2017.

[14] L. Zanzi, A. Albanese, V. Sciancalepore, and X. Costa, "NSBchain: A secure blockchain framework for network slicing brokerage," *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–7.

[15] S. Zheng, T. Han, Y. Jiang, and X. Ge, "Smart contract-based spectrum sharing transactions for multi-operators wireless communication networks," *IEEE Access*, vol. 8, pp. 88547–88557, 2020.

[16] D. B. Rawat, "Fusion of software defined networking, edge computing, and blockchain technology for wireless network virtualization," *IEEE Commun. Mag.*, vol. 57, no. 10, pp. 50–55, Oct. 2019.

[17] J. Qiu, D. Grace, G. Ding, J. Yao, and Q. Wu, "Blockchain-based secure spectrum trading for unmanned-aerial-vehicle-assisted cellular networks: An operator's perspective," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 451–466, Jan. 2020.

[18] D. B. Rawat, "Game theoretic approach for wireless virtualization with coverage and QoS constraints," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2017, pp. 601–606.

[19] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3602–3609, Jun. 2019.

[20] H. Xu, X. Qiu, W. Zhang, K. Liu, S. Liu, and W. Chen, "Privacy-preserving incentive mechanism for multi-leader multi-follower IoT-edge computing market: A reinforcement learning approach," *J. Syst. Archit.*, vol. 114, Mar. 2021, Art. no. 101932. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762120301910

[21] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019.

[22] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.

[23] M. Mamman, Z. M. Hanapi, A. Abdullah, and A. Muhammed, "Quality of service class identifier (QCI) radio resource allocation algorithm for LTE downlink," *PLoS ONE*, vol. 14, no. 1, pp. 1–22, Jan. 2019. [Online]. Available: https://doi.org/10.1371/journal.pone.0210310

[24] K. Wang, F. R. Yu, and H. Li, "Information-centric virtualized cellular networks with device-to-device communications," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9319–9329, Nov. 2016.

[25] S. M. Ross, *Introduction to Probability Models*, 9th ed. Orlando, FL, USA: Academic, 2006.

[26] C. Xu, T. Li, M. Sheng, and J. Li, "Self-organized dynamic caching space sharing in virtualized wireless networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2016, pp. 1–6.

[27] M. Krstić and L. Krstić, "Hyperledger frameworks with a special focus on hyperledger fabric," *Vojnotehnicki Glasnik*, vol. 68, pp. 639–663, Jul. 2020.

[28] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 840–852, Sep./Oct. 2018.

[29] Z. Su, Y. Wang, Q. Xu, M. Fei, Y.-C. Tian, and N. Zhang, "A secure charging scheme for electric vehicles with smart communities in energy blockchain," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4601–4613, Jun. 2019.

[30] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Oper. Syst. Design Implement. (OSDI)*, 1999, pp. 173–186.

[31] Z. Liu *et al.*, "A survey on applications of game theory in blockchain," 2019. *arXiv:1902.10865*.

[32] G. Sun, K. Xiong, G. O. Boateng, G. Liu, and W. Jiang, "Resource slicing and customization in RAN with dueling deep Q-network," *J. Netw. Comput. Appl.*, vol. 157, May 2020, Art. no. 102573. [Online]. Available: https://doi.org/10.1016/j.jnca.2020.102573

[33] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, vol. 48, 2016, pp. 1995–2003.

[34] K.-H. Chiang and N. Shenoy, "A random walk mobility model for location management in wireless networks," in *Proc. 12th IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, vol. 2, 2001, pp. E43–E48.

[35] F. Muratov, A. Lebedev, N. Iushkevich, B. Nasrulin, and M. Takemiya, "YAC: BFT consensus algorithm for blockchain," 2018, *arXiv:1809.00554*.

Gordon Owusu Boateng received the bachelor's degree in telecommunications engineering from the Kwame Nkrumah University of Science and Technology, Kumasi, Ghana, in 2014, and the master's degree in computer science from the University of Electronic Science and Technology of China, where he is currently pursuing the Ph.D. degree. From 2014 to 2016, he worked under sub-contracts for Ericsson, Ghana, and TIGO, Ghana. He is also a member of the Mobile Cloud-Net Research Team, UESTC. His interests include mobile/cloud computing, 5G wireless networks, data mining, D2D communications, blockchain, game theory, and SDN.

Daniel Ayepah-Mensah received the bachelor's degree in computer engineering from the Kwame Nkrumah University of Science and Technology, Kumasi, Ghana, in 2014, and the master's degree in computer science from the University of Electronic Science and Technology of China (UESTC), where he is currently pursuing the Ph.D. degree. From 2014 to 2017, he worked as a Software Developer. He is also a member of the Mobile Cloud-Net Research Team, UESTC. His interest includes generally wireless networks, big data, and cloud computing.

Daniel Mawunyo Doe received the bachelor's degree in computer engineering from the Kwame Nkrumah University of Science and Technology, Kumasi, Ghana, in 2018. He is currently pursuing the M.Sc. degree in computer science and engineering with the University of Electronic Science and Technology of China (UESTC). From 2019 to 2021, he was a member of the intelliGame Team, UESTC. His general research interests include game theory, federated learning, wireless networks, big data, and cloud computing.

Abegaz Mohammed received the B.Sc. degree in computer science from Ambo University, Ethiopia, in 2010, the M.SC. degree in computer science from Addis Ababa University, Ethiopia, in 2015, and the Ph.D. degree in computer science from the University of Electronic Science and Technology of China (UESTC) in 2021. He is currently a Postdoctoral Researcher with Zhejiang Normal University, Jinhua, China. From 2010 to 2016, he worked with Dilla University, Ethiopia, as a Graduate Assistant and a Lecturer, and worked with the College of Engineering and Technology as a member of the Academic Committee and an Associate Registrar. He has two technical journal papers. He is also a member of the Mobile Cloud-Net Research Team, UESTC. His research interests include wireless network, mobile edge computing, fog computing, UAV network, IoT, and 5G wireless network.

Guolin Sun (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in communication and information system from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2000, 2003, and 2005, respectively. After Ph.D. graduation in 2005, he has got eight years industrial work experiences on wireless research and development for LTE, Wi-Fi, Internet of Things, cognitive radio, localization, and navigation. Before he joined UESTC, as an Associate Professor in August 2012, he worked with Huawei Technologies Sweden. He has filed over 30 patents and published over 40 scientific conference and journal papers, acts as a TPC member of conferences. His general research interests include software defined networks, network function virtualization, and radio resource management. He currently serves as the Vice-Chair of the 5G Oriented Cognitive Radio SIG of the IEEE Technical Committee on Cognitive Networks of the IEEE Communication Society.

Guisong Liu received the B.S. degree in mechanics from Xi'an Jiao Tong University, Xi'an, China, in 1995, and the M.S. degree in automatics and Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2000 and 2007, respectively. He was a Visiting Scholar with Humbolt University, Berlin, from September 2015 to December 2015. He is currently a Full Professor and the Dean of the Economic Information Engineering School, Southwestern University of Finance and Economics, China. His research interests include pattern recognition, neural networks, and machine learning.