Blockchain and Game Theory Convergence for Network Slice Brokering

Tharaka Hewa, Pawani Porambage, Anshuman Kalla, and Diana Pamela Moya Osorio, University of Oulu
Madhusanka Liyanage, University College Dublin
Mika Ylianttila, University of Oulu

As a distributed ledger technology, blockchain has received significant attention in revolutionizing telecommunication and networking domains. This article proposes a blockchain-based network slice brokering mechanism for multioperator and multitenant environments of the envisioned 6G networks.

The 6G telecommunication infrastructure is expected to facilitate more diversified consumer requirements arising from various emerging use cases. The 5G network slicing allows on-demand creation of multiple end-to-end (E2E) logical networks over a common physical (mobile network) infrastructure. Following the trends observed in the 5G era, 6G is envisioned to intensively use sophisticated and secure slicing for complex multitenant multioperator scenarios.

Efficient network sharing is one of the most vital requirements in future telecommunication in terms of consumer service values and profitability of resource providers (RPs) including mobile network operators (MNOs).[1] Slicing allows the realization of a multitenancy paradigm where multiple network tenants can simultaneously access the shared computing, storage, and networking resources offered by an infrastructure provider (InP). Here, network tenants can be an industry vertical, a mobile virtual

Digital Object Identifier 10.1109/MC.2022.3165533
Date of current version: 8 March 2023

network operator, or an over-the-top service provider. A network slice (NS) broker is an entity that facilitates the formation of new slices based on consumers' requirements. Slicing also allows InPs to virtualize and trade their resources dynamically to network tenants, thereby allowing better business models with optimal slices that provide a lower price to the tenant and a higher profit to the MNOs. Since 6G mobile networks are seen to nurture more diversified applications and heterogeneous traffic scenarios, an NS broker needs to be executed autonomously in a trustless environment comprising multiple market players' macro and microlevel participation.

Resource allocation applications in different contexts, including telecommunications, have been modeled using game theory.[2,3] The players of such game models consist of tenants and MNOs who have well-established objectives in terms of profit and usability. In contrast, network slicing in multioperator and multitenant scenarios require on-demand federation of MNOs per request basis along with low latency selection operation and high scalability to handle massive consumer groups.

Distributed ledger technology (DLT) is a disruptive technological infrastructure with many potential synergies in the telecommunications and networking industries.[4] The rationale behind a DLT is the distributive storage of the entire database of records (that is, digital ledger) at all of the nodes in a network. Thus, DLT aims to eliminate the use of a centralized server and brings in place a decentralized cryptographic mechanism to record transactions in a secure and immutable manner. As the most popular DLT, blockchain comprises immutable and time-stamped blocks containing validated transactions and connected using hash-based chain and timestamps.

Consensus is an agreement procedure between the members in the blockchain for appending a new block. Many consensus protocols exist, and each has distinguishing features, including fault tolerance, mining overheads, and block verification time, which have to be considered in the application of 5G and beyond 5G scenarios.[5,6] The consensus protocol features, such as block-mining time, and mining computational overheads, directly affect the performance of the entire sequential workflow of 5G and beyond networks.

An optimal NS, offered by RP(s) to a network tenant (that is, the consumer), is defined as a slice that provides the best match for the requested resources in terms of consumer price and RP profit. Such an optimal slice aims to minimize the price to be paid by the consumer (that is, the requesting network tenant) and maximize the profit gained by the supplier (that is, MNO). We formulated the optimal slice selection using the Stackelberg game.[7] The entire slice selection process, including the optimal slice selection algorithm, is encoded in smart contracts to achieve decentralized, transparent, and immutable operation of the slice selection. Blockchain has immense potential to improve various technical aspects and use cases of current and next-generation mobile networks such as enhanced security features, spectrum sharing, decentralized network management, and security orchestration.[8]

To tackle heterogeneous traffic scenarios, 6G networks may need to build up the complex connectivity among the tenants and highly diversified resource and service providers in a more autonomous manner. A DLT-based NS broker will be helpful to mediate the given two ends. Although the NS brokering concept is a recently evolving topic,[9] the pragmatic usage of it along with blockchain technology is yet to be discovered. To the best of our knowledge, there is no current work to demonstrating a fully functional blockchain-based NS brokering mechanism for the multioperator multitenant scenario with practical implementation. Herein, we propose a game-theoretic model to select the best match of tenants on one side and the MNOs or RPs on the other side. This would ensure both customer and service provider ends can reach their optimal utilities. In this article, we develop a blockchain-based secure and federated NS brokering (SFSBroker) mechanism for multioperator multitenant scenarios in the envisioned 6G networks.

RESOURCE ALLOCATION APPLICATIONS IN DIFFERENT CONTEXTS, INCLUDING TELECOMMUNICATIONS, HAVE BEEN MODELED USING GAME THEORY.

NS BROKER

Overview

In 5G, NS brokering is introduced as a new business model for dynamic network sharing wherein a logically centralized entity named the *slice broker* governs the resource trading between InPs at one end and multiple network tenants at the other end.[9] Apart from facilitating on-demand resource allocation, the slice broker performs admission control based on traffic monitoring and forecasting and mobility management based on a global network view. It configures radio access network (RAN) schedulers to support multitenancy use cases. As defined in Samdanis et al.,[9] 5G NS broker is colocated at master operator-network manager, which monitors and controls the shared RAN, and interacts with sharing operator-network manager, which provides feedback.

NETWORK SLICING DIFFERS FROM NS BROKERING SINCE NETWORK SLICING FACILITATES THE CUSTOM LOGICAL NETWORK CREATION ON TOP OF THE SHARED INFRASTRUCTURE.

Blockchain-based NS broker

Many research efforts have already been taken to investigate how to combine blockchain and 5G network slicing technology.[4] However, only a few works are explicitly focusing on developing an NS brokering framework using blockchain.[10] Moreover, they are still not close enough to the actual deployment phase in a multioperator multitenant platform, which is foreseen in the next-generation networks. In Backman et al.,[11] blockchain is introduced as an additional trust layer in slice broker for trading and dynamic billing. The blockchain-based slice brokering mechanism in Valtanen et al.[12] uses smart contracts for enabling dynamic and autonomous slice management.

The blockchain-based distributed market in Afraz and Ruffini[13] uses a novel double auctioning mechanism and trades NS as a commodity comprised of parameters such as RAN, computational resources, and storage. The blockchain-based hierarchical architecture in Zanzi et al.[14] enables InPs to allocate network resources for slice brokers through smart contracts and redistribute resources among tenants in a secure, automated, and scalable manner. In Nour et al.,[15] a slice provider receives a request to build an E2E slice, thus it publishes in the blockchain a request for resources of each subslice composing the E2E slice. The work in Antevski and Bernardos[16] proposes a DLT-based solution for the federation of 5G network services including registration, negotiation, and charging through smart contracts.

Network slicing differs from NS brokering since network slicing facilitates the custom logical network creation on top of the shared infrastructure. In contrast, NS brokering facilitates the selection strategy of the shared infrastructure based on specific requirements such as the best price for the tenants and maximum profit to the RPs.

The blockchain-based NS brokering frameworks aim to ideally cater for the scalable and shorter time-to-market deployment of NS in future networks. The smart contracts running on top of blockchain decentralize and scale up the entire capacity of NS brokering. Furthermore, smart contracts accelerate the selection process by moving the automated selection service from the cloud to the edge of Internet of Things (IoT) networks.

PROPOSED SFSBROKER MECHANISM

System model

In the 6G era, it would be necessary to maintain interoperability between the massive number of business vertical tenants from different domains. As illustrated in Figure 1, we consider a holistic scenario where multiple tenants (that is, different use cases) are accessing services from a common resource pool.

RPs include virtualized resources, physical resources, and infrastructure for communication and computation. These resources are granted to the consumers in the form of NSs where RAN, core network, computational infrastructure, and storage are potential candidates to be shared with the consumers as per requirement.

SFSBroker acts as a global mediator between two ends to facilitate the delivery of NSs to the tenants, which are acquired from infrastructure providers. To provide a coherent and real time service, the brokering mechanism should have a holistic knowledge about the demand and supply status of consumers and service providers. SFSbroker handles tasks such as receiving a slice request from tenants and disseminating it to RPs, selecting an optimal slice offer from a pool of proposals from RPs, monitoring traffic and coordinating with orchestration services. This mechanism should cater to extensive service requests generated by the massive number of tenants with assured security (that is, assure authentication, availability, privacy, trust, and access control).

Functional architecture

The high-level system model is shown in Figure 1. Next, we describe the architectural framework of the SFSBroker mechanism and flow diagram in Figure 2.

Fog nodes. The fog nodes represent the consumer end of the proposed solution, which directly communicates with the IoT tenants (deployed in different use cases). In the multitenant scenario, each fog node is serving one or multiple IoT tenant clusters in specific use cases.

RP. In a multioperator platform, RPs are considered as the entities that provide networking and computational resources or infrastructure to the consumers. These RPs may include cloud computational infrastructure, storage, network services, and mobile data connectivity. There can be a versatile collection of service providers under this, such as local (micro) network operators, MNOs, and cloud service providers.

SFSBroker. The middle layer in Figure 2 represents SFSBroker, which acts as the mediator between fog nodes and RPs and is deployed as a blockchain network running as a service in the central cloud. The SFSBroker mainly consists of three submodules, namely prime mover, mediator, and global slice

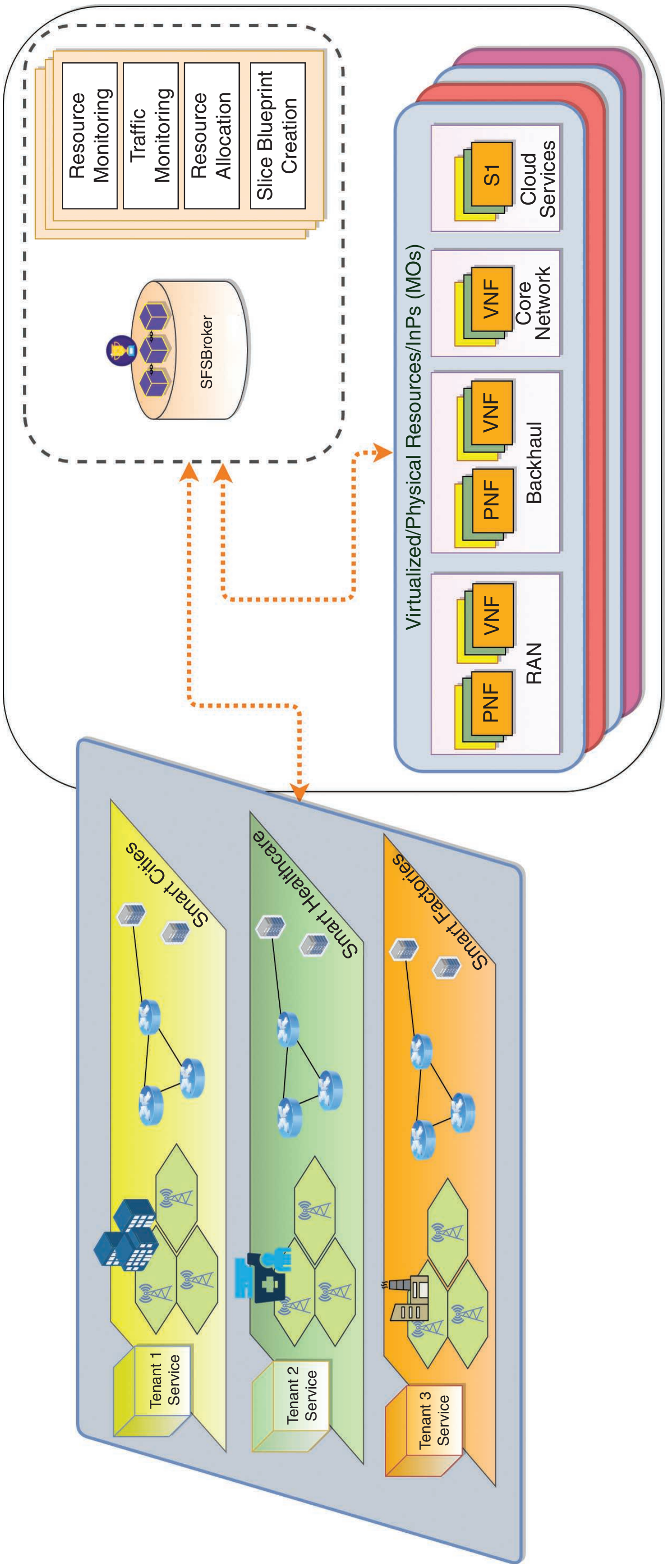

FIGURE 1. A use case scenario for SFSBroker that serves multiple tenants in different application domains.

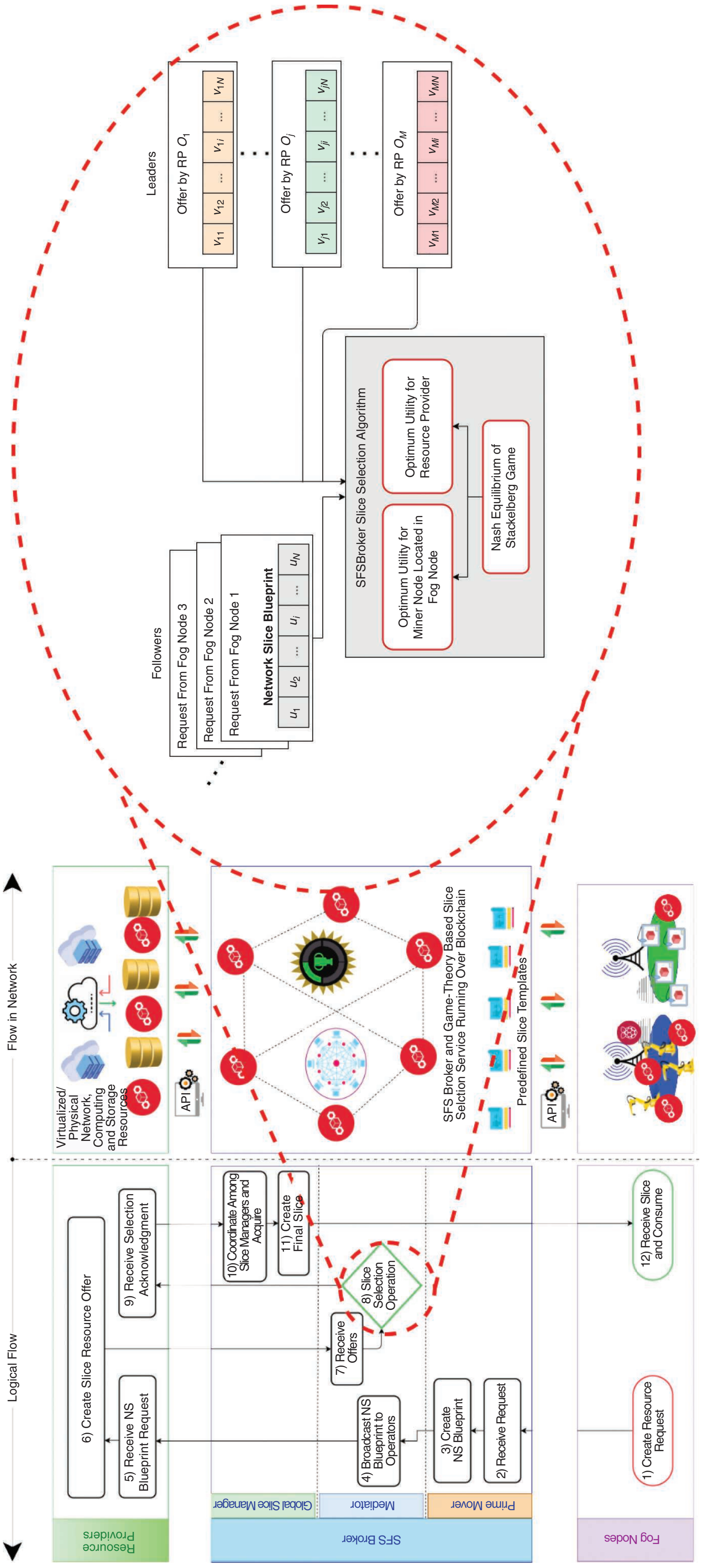

FIGURE 2. The workflow of SFSBroker algorithm. API: application programming interface.

manager. Prime mover is responsible for handling resource requests and creating the NS blueprint. Mediator broadcasts NS blueprints to RPs and runs the slice selection algorithm. The selection of best matching RPs offer (or formulate a new slice with multiple RPs) for a given NS blueprint is performed by running an algorithm modeled using the Stackelberg game.[17] Global Slice Manager coordinates the final slice offer to IoT tenants via the fog node.

Flow of SFSBroker

SFSBroker is deployed as a decentralized entity using consortium blockchain and follows a modular approach for better scalability. As shown in Figure 2, there are 12 steps in the flow of SFSBroker. An instance of the process is triggered when a fog node receives a service request from IoT tenants. In response (step 1), the fog node creates a resource request, embeds it in a transaction, digitally signs it, and sends the request (transaction in blockchain parlance) to SFSBroker. Fog nodes initiate the request on behalf of the IoT tenants as fog nodes are the gateways of IoT nodes to connect with SFSBroker.

This research proposes a Stackelberg game model-based algorithm to select the optimal NS, based on the two input types: IoT tenant requests and RP resource offers. The game model-based selection algorithm has been encoded as a smart contract. Such an implementation offers various advantages such as the elimination of a single point of failure, the capability to move the selection service from the cloud to the edge, and an immutable transaction ledger for better transparency of operations.

The prime mover receives the request (step 2), verifies the digital signature (to check the authenticity of the requesting fog node) and stores the verified request in the blockchain. Then (step 3), the prime mover creates a blueprint of a NS based on the quantitative demand (for various predefined categories of resources) in the received request and sends the NS blueprint to the mediator module. The mediator module simply broadcasts the NS blueprint requests to all of the available RPs (step 4). Broadcasting is accomplished by writing in the blockchain so that all of the authentic RPs can access the NS blueprints. At the time of broadcasting, the mediator module also starts a timer t corresponding to each NS blueprint request.

When RPs retrieve the NS blueprint request, they analyze it to check feasibility (step 5). Meaning that every RP categorically compares the amount of resources demanded with the available unoccupied resources. Then (step 6), the interested RPs, who are willing to lease their resources (as per the demand), create offers comprising price and other specifications, embed in digitally signed transactions, and send them to the SFSBroker mediator module.

Subsequently, the mediator module verifies all of the incoming offers and stores them on the blockchain upon reaching consensus through the approval of offer values (step 7). For a given NS blueprint request, the expiration of timer t marks the end of the time window accepting offers from RPs. Furthermore, it triggers the commencement of step 8, which starts the execution of a selection algorithm on the offers received. With the outcome of the selection algorithm, the mediator, as per the optimal offer, sends one or more acknowledgment to the winning RP(s) by writing to their blockchain address(es) upon reaching a consensus.

Once the winning RP receives information about its selection to offer a complete or a part of an NS (step 9), it (virtually) slices the resource(s) and informs the global slice manager module of SFSBroker. Then, the global slice manager coordinates with the slice manager of the winning RP(s), acquires the constituent resources (step 10), creates the final federated NS and hands it over to the fog node (step 11). Finally, the fog node receives the federated NS (step 12). Note that all communications between fog nodes, SFSBroker, and RPs are recorded in immutable transactions and digitally signed. Furthermore, using blockchain-based SFSBroker, optimal offers are selected in a decentralized manner that gives trust to the stakeholders.

Slice selection algorithm

From a consumer's perspective, the lowest price is important and from the RP's perspective, maximized profit is important. SFSBroker's slice selection algorithm needs to consider both the viewpoints of RPs and fog nodes (Figure 2). Both RPs and fog nodes are constantly adjusting their strategies to maximize their utilities. In the selection algorithm, we discuss how one RP becoming an exclusive winner is merely a special case where the winning RP can provide the best offer for all of the resource categories in the given NS. However, with the proper adjustments, an output of the same selection algorithm may create an optimal offer in which resources from multiple providers form a federated NS. At most, the total number of winning RPs can be as many as the total number of distinct categories of resources.

We consider that a particular NS blueprint is created with n number of resource (or network functions) categories. In a resource request created by a certain fog node, u_i denotes the amount of resource demand for the ith resource, where $i \in \{1, 2, \ldots, n\}$ There are m number of RPs such that O_j denotes the jth RP where $j \in \{1, 2, \ldots, m\}$ We consider that RP (or operator) O_j sets the pricing strategy $\{v_j = [v_{ji} i \in N : 0 < v_{ji} < \bar{v}]\}$ as the unit price of ith resource, where v_{ji} is the price offered and $\bar{v}$ is the maximum price. Moreover, c is taken as the common and constant cost resulting from the general operation and maintenance cost.

As mentioned in Figure 2, the selection algorithm should find the optimum expected utilities (reward) by each RP, offered for a given NS blueprint. Herein, we consider one NS blueprint formed based on a resource request created by a miner node located in a fog node. Moreover, the expected utility should be computed for a given resource category requested by the miner node.

Therefore, the expected utility (reward) by O_j RP can be expressed as

$$P_j = \sum_{i=1}^{N} u_i v_{ji} - \sum_{i=1}^{N} c u_i. \tag{1}$$

In addition to that, we define a utility function P_i expected utility (reward) for R_i resource category requested by the miner node located at fog node (based on the offer given by O_j):

$$P_i = P \times \frac{u_i}{\sum_{i=1}^{N} u_i} - v_{ji} \times u_i. \tag{2}$$

As described previously, after having all of the offers from RPs, the selection algorithm first computes the total service demand of fog nodes and sets the offer prices to earn more profit for RPs.

On the other hand, the miner nodes located in fog nodes, need to maximize the reward received for each resource requirement. Therefore, observing the price strategies of RPs, the selection algorithm will formulate the optimization problem of miners using (1) and (2) as described in Yao et al.[7]

The mathematical model is formulated for two sides in the Stackelberg game, taking the RPs as leaders and fog nodes (miner nodes) as followers. The selection algorithm is responsible for updating both RPs and the fog nodes about how they are capable of constantly adjusting strategies to maximize their utilities. The objective of the Stackelberg game is to find the Nash equilibrium, where no player has the intention to deviate from its strategy after considering its opponent's choice. As explained in Yao et al.,[7] the utility functions are strictly concave, and the Nash equilibrium exists. To find the NE, a reinforcement learning algorithm is used as described in Yao et al.[7] In the first part of the selection, the algorithm should be run for each resource request and compute the optimal values for the operator price and required resource amount for each resource category as shown in Table 1. By referring to the values in Table 1, the NS is formed in such a way as to minimize the total price and match the resource availability with the operators.

EVALUATION

The proposed solution includes four main service components: fog nodes, SFSBroker, 5G infrastructure, and blockchain service. The implementation setup developed to perform a proof of concept of the SFSBroker is illustrated in Figure 3.

Infrastructure placement of the implementation setup

As shown in Figure 3, fog nodes are taken as Raspberry Pies, and the shareable RP infrastructure is simulated using the Ubuntu 18.04 virtual machines deployed on a Lenovo Thinkpad T480S on a Windows 10 (64-bit) host machine with 16-GB RAM. A cloud instance with Intel Xeon CPU 2.33 GHz and 16-GB RAM used to deploy the Hyperledger with public IP access.

IoT tenants, that is, Fog nodes, are implemented using Raspberry Pi 4 Model A devices with 5G dongles (that is, Huawei E3372). We used the 5G test network (5GTN)[18] to connect different components in the testbed. The 5G test network is an experimental 5G network deployed at the University of Oulu and VTT Technical Research Center of Finland, used for 5G-related experiments. 5GTN supports 5G new radio (5GNR) connectivity, edge computing resources, and high-speed connectivity for cloud resources. In our experiment, IoT tenets are connected to the 5GTN, and we used the high-speed Internet connection offered by 5GTN backhaul to connect them with the cloud layer.

Simulation of NS using Docker

We simulate NS instances using Docker containerization. The NS blueprint is simulated using a prebuilt Docker image. The instantiated NS is simulated using the running Docker container initialized with the different resource categories requested by the fog nodes. The Docker containers with specified resources (such as memory and storage), which run on the VMs as indicated in Figure 3, simulate the RP resource utilization by NS. For the evaluation, we assume that the corresponding services of each resource request are running in different ports of the instantiated Docker container, and each service is accessible to the consumers through the ports. However, the simulation ensures that the selected slice has been instantiated.

SFSBroker deployment in the implementation setup

The implementation setup demonstrates a near realistic transaction simulation (Figure 3) for the proposed architecture:

- *Blockchain*: Blockchain is implemented using five-node Hyperledger Fabric[19] 1.4.4 instance with Raft consensus configuration that runs Java smart contracts.
- *SFSBroker's components*: Prime mover, mediator, and global slice manager (shown in Figure 2) are implemented as smart contracts.

TABLE 1. The optimal unit prices of operators and optimal resource demand from each category.

Operator	Optimal unit price	Optimal resource demand				
		R_1	R_2	R_3	...	R_n
O_1	v^*_1	u^*_{11}	u^*_{12}	u^*_{13}		u^*_{1n}
O_2	v^*_2	u^*_{21}	u^*_{22}	u^*_{23}		u^*_{2n}
...						
O_m	v^*_m	u^*_{m1}	u^*_{m2}	u^*_{m3}		u^*_{mn}

- *RPs*: For simplicity of this initial prototype, we consider all types of RPs and MNOs in a common ground as identical entities capable of providing computational or networking resources. Therefore, both RP and MNO are terms used in the rest of the section. MNOs are represented by the virtual machines (VMs) that have access to preallocated computational resources. One slice is a subset of VM resources that will be available to the tenants.
- *Connectivity*: Connectivity among fog nodes, blockchain, and RPs is established using Message Queuing Telemetry Transport (MQTT). Hyperledger software development kit integrates with MQTT library to push the resource requests and offers to the blockchain.

The implementation steps of SFSBroker reflected in Figure 3 are:

- *Step 1*: Fog nodes place NS blueprint request (that is, with 1 ... *N* resource categories) to SFSBroker service by invoking application programming interface (API).
- *Step 2*: SFSBroker retrieves the NS blueprint from the tenants and the smart contract checks parameters with the blocked transaction committed in the request.
- *Step 3*: SFSBroker publishes the NS request in the blockchain. RPs receive a request and formulate individual offers.
- *Step 4*: RPs respond with offers. SFSBroker receives the responses within designated time window. The received offers are committed as transactions to the blockchain. The corresponding NS blueprint is queried from the ledger, and parameters are validated with the ledger transaction. The smart contract runs the selection algorithm described in the "Performance Evaluation" section and selects the best RP offer.
- *Step 5*: SFSBroker formulates the slice and acknowledges the RPs and the fog node about the optimal offer. The selected RP instantiates the slice.

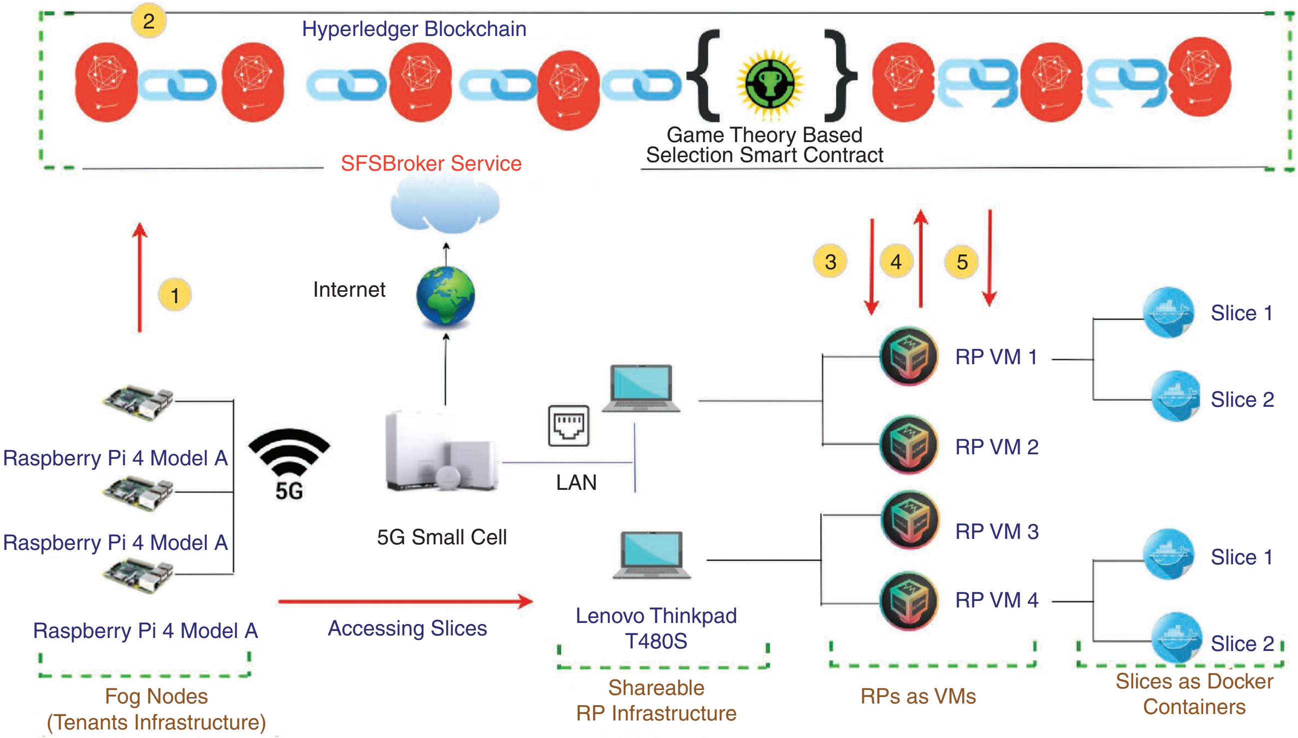

FIGURE 3. The SFSBroker testbed implementation setup. LAN: local area network; VM: virtual machine.

Performance evaluation

E2E slice creation latency with variable block time. This experiment demonstrates the multitenant scenario and the evaluation of E2E slice creation latency (steps 1–10 in Figure 2). E2E latency is measured with variable block generation time intervals as configured in Hyperledger. For a particular configured block generation time, the fog node initiates a set of concurrent transactions (that is, on each test 1, 10, 25, 50). The E2E latency is recorded for each slice with 100 trials for a given block time and concurrent transaction, where the confident intervals (that is, 95%) are computed [Figure 4(a)].

The experiment results show that the reduction in the block generation interval is not directly advantageous in terms of latency. With a high number of concurrent transactions (that is, 25 and 50) and low block generation time (that

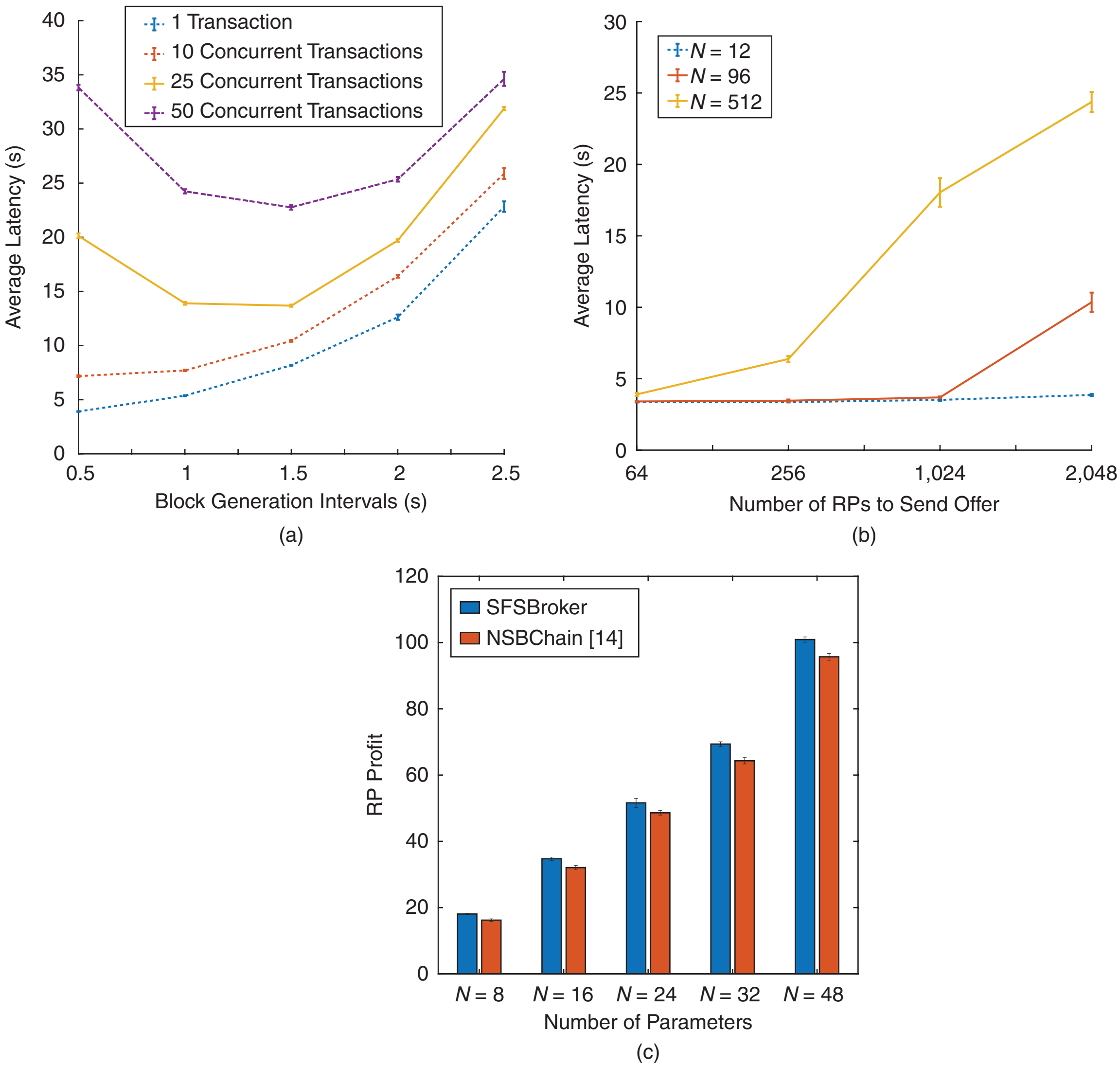

FIGURE 4. The performance evaluation results in the implementation and simulation. (a) The E2E latency of slice selection with different block-mining time configurations. (b) The E2E latency of slice selection for multiple RPs and different parameter counts. (c) A simulation result comparison of SFSBroker and NSBChain.[14]

is, 500 ms and 1 s), we observe that the blocks are mined before the completion of intermediary steps of transactions in the entire batch. In such cases, transactions of each batch are dispersed among multiple blocks. When the completion latency of the entire batch has been calculated, the delay that occurred by dispersing transactions within multiple blocks affects the completion time of entire batch. For other cases, E2E slicing creation latency increases with the block generation time.

Slice selection latency. This test focused on the latency of the game theory-based selection algorithm for different inputs. Here we added multiple RPs by increasing the number of parameters *N* in the slice request. The experiment was performed at a fixed block-mining time of 1,000 ms. One transaction per trial is sent to the SFSBroker and 100 trials are performed on each test for a specific RP and parameter setting. Under this setup, the latency is measured for steps 8 and 9, which are indicated in Figure 2. This includes selecting an optimal offer for a tenant request, committing the transaction to the ledger with the selected MNO offer, and approving the transaction in the ledger. When the number of parameters(*N*) increases, the latency also increases.

According to the graphs in Figure 4(b), the increasing number of parameters (*N*) and RPs directly impact the selection latency of the algorithm. The algorithm can select the best offer within 30 s, even with 2,048 RPs and 512 parameters. Since we need to evaluate the performance on different scales of the inputs, we increased the number of RPs up to 2,048 in the experiment. We considered RPs and MNOs at the same ground and scaling up to 2,048 RPs [Figure 4(b)] simulates the scenarios when local 5G operators deliver the services as RPs.

Comparison with related work

We compared the behavior of SFSBroker with NSBChain[14] algorithms using Matlab. The number of RPs (*M*) is kept fixed and the number of parameters (*N*) varies in every experiment. Each experiment consists of 100 trials and the RPs profits have been calculated on each trial. The consumer resource requests, the costs to deliver the resource request, and the profits were generated randomly in each trial. We assumed that the final price offered to the consumer of each slice request is the sum of randomly generated cost and profit values in each trial.

The inputs to each algorithm contain the consumer resource demand and the RPs' resource offers. According to the results obtained in Figure 4(c), we observe that the RP profits are higher for SFSBroker than for NSBChain. In SFSBroker, we consider the profit factor of RPs and the lowest price in the selection process rather than selecting the lowest offer. Therefore, SFSBroker provides more fairness to both consumers and RPs in the slice selection process than NSBChain.

In this work, we proposed a blockchain-based NS brokering mechanism (SFSBroker) for applications in the multioperator multitenant environments expected in 6G networks. In SFSBroker, the best match between tenant and operator, which ensures optimal utilities to both customer and service provider, is obtained by modeling as a Stackelberg game, where Nash equilibrium can be met. Details on the functional architecture and the implementation setup are provided. Moreover, the performance of SFSBroker is evaluated in terms of the E2E slice creation latency and the slice selection latency. The results showed that the E2E slicing creation latency increases with the block generation time. Moreover, increasing numbers of N and RPs strongly impact the slice selection latency.

ACKNOWLEDGMENTS

This research is funded by European Union under INSPIRE5G-plus (grant 871808), Academy of Finland under 6Genesis Flagship (Grant 318927), and Science Foundation Ireland under CONNECT phase 2 (grant 13/RC/2077_P2) projects. The work of Diana P. M. Osorio is funded by Academy of Finland project FAITH (Grant 334280).

REFERENCES

1. A. Antonopoulos, "Bankruptcy problem in network sharing: Fundamentals, applications and challenges," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 81–87, 2020, doi: 10.1109/MWC.001.1900414.
2. M. Vincenzi, A. Antonopoulos, E. Kartsakli, J. Vardakas, L. Alonso, and C. Verikoukis, "Multi-tenant slicing for spectrum management on the road to 5G," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 118–125, 2017, doi: 10.1109/MWC.2017.1700138.
3. N. Zhao, H. Wu, and Y. Chen, "Coalition game-based computation resource allocation for wireless blockchain networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8507–8518, 2019, doi: 10.1109/JIOT.2019.2919781.
4. D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for 5G and beyond networks: A state of the art survey,"

ABOUT THE AUTHORS

THARAKA HEWA is a doctoral student at the University of Oulu, Oulu, 90570, Finland. His research interests include blockchain, public-key infrastructure, 5G, banking systems security, healthcare security, and smart cities. He is a Student Member of IEEE. Contact him at tharaka.hewa@oulu.fi.

PAWANI PORAMBAGE is a postdoctoral researcher at the University of Oulu, Oulu, 90014, Finland. Her main research interests are lightweight security protocols, security and privacy on the Internet of Things and mobile edge computing, and wireless sensor networks. She is a Member of IEEE. Contact her at pawani.porambage@oulu.fi.

ANSHUMAN KALLA is a postdoctoral visiting researcher at the University of Oulu, Oulu, 90570, Finland. His research interests include blockchain, future networks, information-centric networking, and the Internet of Things. He is a Member of IEEE. Contact him at kallanshu@gmail.com.

DIANA PAMELA MOYA OSORIO is a senior research fellow and adjunt professor at the University of Oulu, Oulu, 90570, Finland. She is also a postdoctoral researcher for the Academy of Finland. Her research interests include wireless communications in general, 5G and 6G networks, and physical layer security. She is a Member of IEEE. Contact her at supun.ucsc@gmail.com.

MADHUSANKA LIYANAGE is an ad astra fellow and assistant professor at University College Dublin, Dublin, Ireland. He is also an adjunct professor at the University of Oulu, Oulu, 90014, Finland. His research interests are software-defined networking, the Internet of Things, blockchains, mobile edge computing, and mobile and network security. He is a Senior Member of IEEE. Contact him at madhusanka.liyanage@oulu.fi.

MIKA YLIANTTILA is an associate professor at the University of Oulu, Oulu, 90570, Finland. His research interests include secure, scalable, and resource-efficient techniques for 5G and beyond 5G and Internet of Things systems. He is a Senior Member of IEEE. Contact him at mika.ylianttila@oulu.fi.

J. Netw. Comput. Appl., vol. 166, p. 102,693, Sep. 2020, doi: 10.1016/j.jnca.2020.102693.

5. L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *Proc. 2017 4th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, pp. 1–5, doi: 10.1109/ICACCS.2017.8014672.
6. Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1432–1465, 2020, doi: 10.1109/COMST.2020.2969706.
7. 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, 2019, doi: 10.1109/TII.2019.2902563.
8. T. Hewa, G. Gür, A. Kalla, M. Ylianttila, A. Bracken, and M. Liyanage, "The role of blockchain in 6G: Challenges, opportunities and research directions," in *Proc. 2020 2nd 6G Wireless Summit (6G SUMMIT)*, pp. 1–5, doi: 10.1109/6GSUMMIT49458.2020.9083784.
9. K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 32–39, 2016, doi: 10.1109/MCOM.2016.7514161.
10. A. Boubendir et al., "Federation of cross-domain edge resources: A brokering architecture for network slicing," in *Proc. 2018 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*. pp. 415–423, doi: 10.1109/NETSOFT.2018.8460114.
11. J. Backman, S. Yrjölä, K. Valtanen, and O. Mämmelä, "Blockchain network slice broker in 5G: Slice leasing in factory of the future use case," in *Proc. 2017 Internet Things Business Models, Users, Netw.*, pp. 1–8, doi: 10.1109/CTTE.2017.8260929.
12. K. Valtanen, J. Backman, and S. Yrjölä, "Creating value through blockchain powered resource configurations: Analysis of 5G network slice brokering case," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, 2018, pp. 185–190, doi: 10.1109/WCNCW.2018.8368983.
13. N. Afraz and M. Ruffini, "5G network slice brokering: A distributed blockchain-based market," in

Proc. 2020 Eur. Conf. Netw. Commun. (EuCNC)*, pp. 23–27, doi: 10.1109/EuCNC48522.2020.9200915.

14. L. Zanzi, A. Albanese, V. Sciancalepore, and X. Costa-Pérez, “NSBchain: A secure blockchain framework for network slicing brokerage,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–7, doi: 10.1109/ICC40277.2020.9149414.
15. 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, 2019, doi: 10.1109/LNET.2019.2915117.
16. K. Antevski and C. J. Bernardos, “Federation of 5G services using distributed ledger technologies,” *Internet Technol. Lett.*, vol. 3, no. 6, p. e193, 2016, doi: 10.1002/itl2.193.
17. Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, “Resource allocation for information centric virtualized heterogeneous networks with in-network caching and mobile edge computing,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11,339–11,351, 2017, doi: 10.1109/TVT.2017.2737028.
18. E. Piri et al., “5GTN: A test network for 5G application development and testing,” in *Proc. 2016 Eur. Conf. Netw. Commun. (EuCNC)*, pp. 313–318, doi: 10.1109/EuCNC.2016.7561054.
19. E. Androulaki et al., “Hyperledger fabric: A distributed operating system for permissioned blockchains,” in *Proc. 13th EuroSys Conf.*, 2018, pp. 1–15, doi: 10.1145/3190508.3190538.

Digital Object Identifier 10.1109/MC.2023.3241475