Blockchain and Game Theory Convergence for Network Slice Brokering

Tharaka Hewa, *Student Member, IEEE,* Pawani Porambage, *Member, IEEE,*
Anshuman Kalla, *Member, IEEE,* Diana Pamela Moya Osorio, *Member, IEEE*
Madhusanka Liyanage, *Senior Member, IEEE* and Mika Ylianttila, *Senior Member, IEEE*

***Abstract*—As a distributed ledger technology, blockchain has received significant attention in revolutionizing telecommunication and networking domains. Besides, network slicing is a key enabling technology in 5G and a road map to the envisioned 6G, which expects to support multi-tenant and multi-operator environments. In this context, the concept of Network Slice (NS) broker has emerged as a promising business entity to facilitate dynamic resource trading between network operators or resource providers and multiple tenants. This work proposes a blockchain-based NS brokering mechanism for multi-operator and multi-tenant environments of the envisioned 6G networks. Our solution, the so-called SFSBroker, utilizes the Stackelberg game based approach to find the best matching NS offered by a Resource Provider (RP) for a resource request created by a tenant. We provide a detailed implementation of the SFSBroker mechanism, which runs as a blockchain service in such a multi-operator multi-tenant platform, focusing on possible future improvements.**

***Index Terms*—Blockchain, Network Slicing, Game theory, multi-operator, multi-tenant, 5G, 6G.**

I. Introduction

The sixth-generation (6G) telecommunication infrastructure is expected to facilitate more diversified consumer requirements arising from various emerging use cases. The fifth-generation (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 multi-tenant multi-operator 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 (RP) including mobile network operators (MNO) [1]. Slicing allows the realization of a multi-tenancy 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 Network Operator (MVNO), or an Over-The-Top (OTT) service provider. An 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 trust-less environment comprising multiple market players' macro- and micro-level 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 multi-operator and multi-tenant 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 (i.e., digital ledger) at all 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 timestamped 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 (i.e., 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 (i.e., the requesting network tenant) and

Tharaka Hewa, Pawani Porambage, Anshuman Kalla, Diana Moya Osorio and Mika Ylianttila are with the Center for Wireless Communications, University of Oulu, Finland. e-mail:{firstname.lastname}@oulu.fi

Madhusanka Liyanage is with the School of Computer Science, University College Dublin, Ireland and the Center for Wireless Communications, University of Oulu, Finland. e-mail:madhusanka@ucd.ie

This research is funded by European Union under INSPIRE5G-plus(Grant No: 871808), Academy of Finland under 6Genesis Flagship (grant 318927), and Science Foundation Ireland under CONNECT phase 2 (Grant no. 13/RC/2077_P2) projects. The work of Diana P. M. Osorio is funded by Academy of Finland project FAITH (grant 334280).

maximize the profit gained by the supplier (i.e., 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 multi-operator multi-tenant 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 multi-operator multi-tenant scenarios in the envisioned 6G networks.

II. Network Slice Broker

A. 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 multi-tenancy use cases. As defined in [9], 5G NS broker is co-located at Master Operator-Network Manager (MO-NM), which monitors and controls the shared RAN, and interacts with Sharing Operator-Network Manager (SO-NM), which provides feedback.

B. Blockchain-based Network Slice 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 multi-operator multi-tenant platform, which is foreseen in the next-generation networks. In [11], blockchain is introduced as an additional trust layer in slice broker for trading and dynamic billing. The blockchain-based slice brokering mechanism in [12] uses smart contracts for enabling dynamic and autonomous slice management.

The blockchain-based distributed market in [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 [14] enables InPs to allocate network resources for slice brokers through smart contracts and re-distribute resources among tenants in a secure, automated and scalable manner. In [15], a slice provider receives a request to build an end-to-end (E2E) slice, thus it publishes in the blockchain a request for resources of each sub-slice composing the E2E slice. The work in [16] proposes a DLT-based solution for the federation of 5G network services including registration, negotiation, and charging through smart contracts.

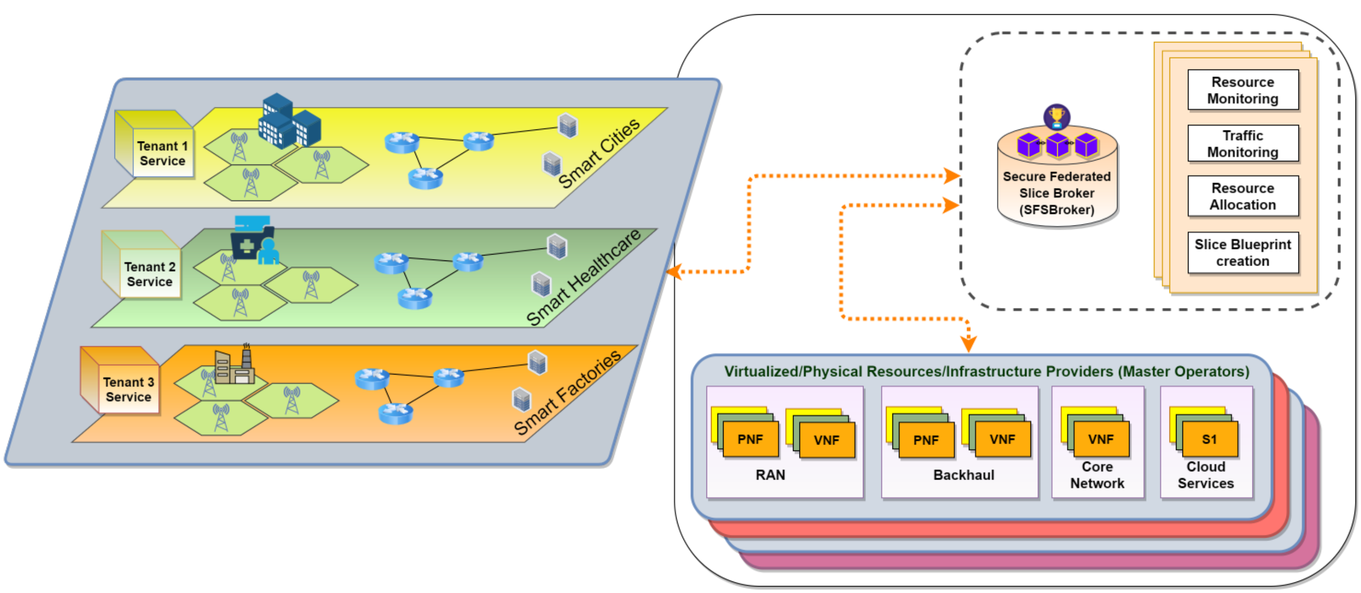


Fig. 1: A use case scenario for SFSBroker that serves multiple tenants in different application domains.

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 IoT networks.

III. Proposed SFSBroker Mechanism

A. 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 (i.e., 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 (i.e., assure authentication, availability, privacy, trust, and access control).

B. 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.

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

*2) **Resource Provider(RP)**:* In a multi-operator 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 (CSP).

*3) **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 sub-modules, namely Prime Mover, Mediator, and Global Slice 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 RP's offer (or formulate a new slice with multiple RPs) for a given NS blueprint is performed by running an algorithm modelled using the Stackelberg game [17]. Global Slice Manager coordinates the final slice offer to IoT tenants via the fog node.

C. 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 the available RPs (*step 4*). Broadcasting is accomplished by writing in the blockchain so that all 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.

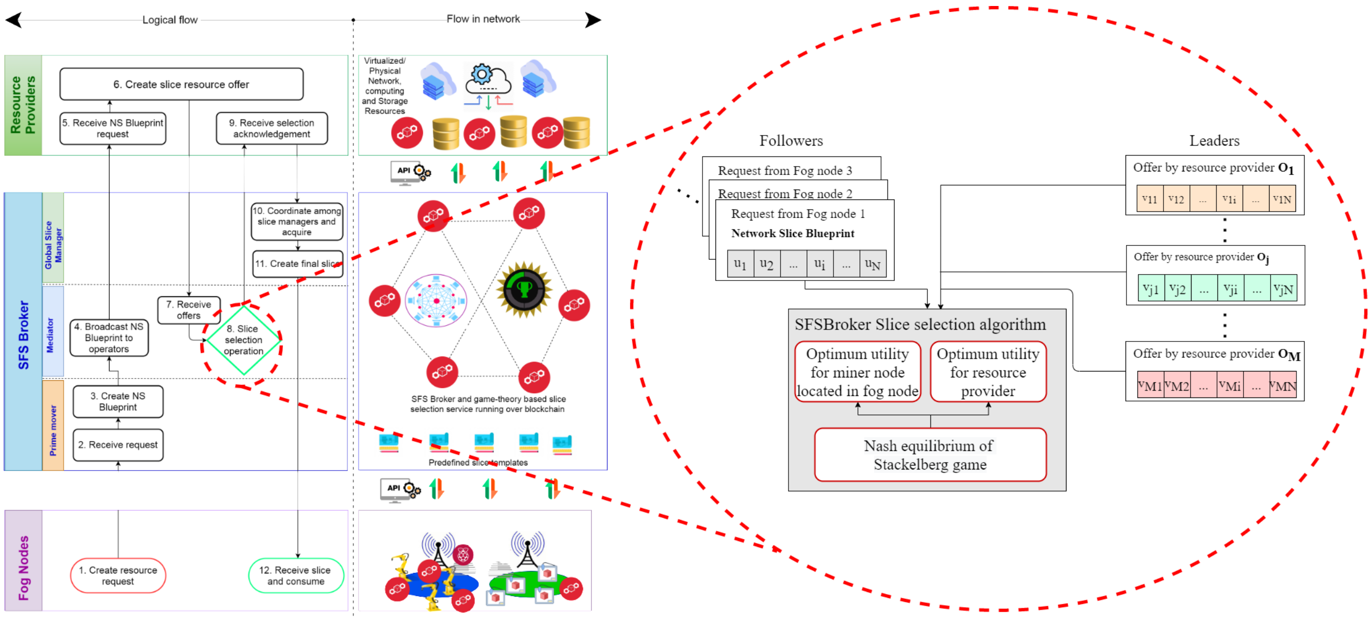


Fig. 2: The Workflow of SFSBroker Algorithm

Subsequently, the mediator module verifies all 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 acknowledgement 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.

D. Slice Selection Algorithm

From 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 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 i^{th} resource, where $i \in \{1, 2, ..., n\}$. There are m number of RPs such that O_j denotes the j^{th} RP where $j \epsilon \{1, 2, ..., m\}$. We consider that RP (or operator) O_j sets the pricing strategy $\{v_j = [v_{ji} i \in N : 0 < v_{ji} < \overline{v}]\}$ as the unit price of i^{th} resource, where v_{ji} is the price offered and $\overline{v}$ 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 resource provider 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):

$$P_i = P \times \frac{u_i}{\sum_{i=1}^{N} u_i} - v_{ji} \times u_i \quad (2)$$

As described above, after having all 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 Eq. 1 and Eq. 2 as described in [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 [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 [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 I. By referring to the values in Table I, the NS is formed in such a way as to minimize the total price and match the resource availability with the operators.

TABLE I: Table for optimal unit prices of operators and optimal resource demand from each category

		Optimal resource demand				
Operator	**Optimal unit price**	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}^*

IV. 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 (PoC) of the SFSBroker is illustrated in Figure 3.

A. 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 16GB RAM. A cloud instance with Intel(R) Xeon(R) CPU 2.33GHz and 16GB RAM used to deploy the Hyperledger with public IP access.

IoT tenants, i.e., Fog nodes, are implemented using Raspberry Pi 4 Model A devices with 5G dongles (i.e., 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 Centre 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.

B. Simulation of NS Using Docker

We simulate NS instances using Docker containerization. The NS blueprint is simulated using a pre-built 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.

C. SFSBroker Deployment in the Implementation Setup

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

Blockchain is implemented using 5 node Hyperledger Fabric [19] 1.4.4 instance with RAFT consensus configuration that runs Java smart contracts.

SFSBroker: The components of SFSBroker (i.e., prime mover, mediator, global slice manager in Figure 2) are implemented as smart contracts.

Resource Providers: 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 pre-allocated computational resources. One slice is a subset of VM's resources that will be available to the tenants.

Connectivity between fog nodes, blockchain, and RP s 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.

Implementation steps of SFSBroker reflected in Figure 3:

- Step 1: Fog nodes place NS blueprint request (i.e., with 1...N resource categories) to SFSBroker service by invoking 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.

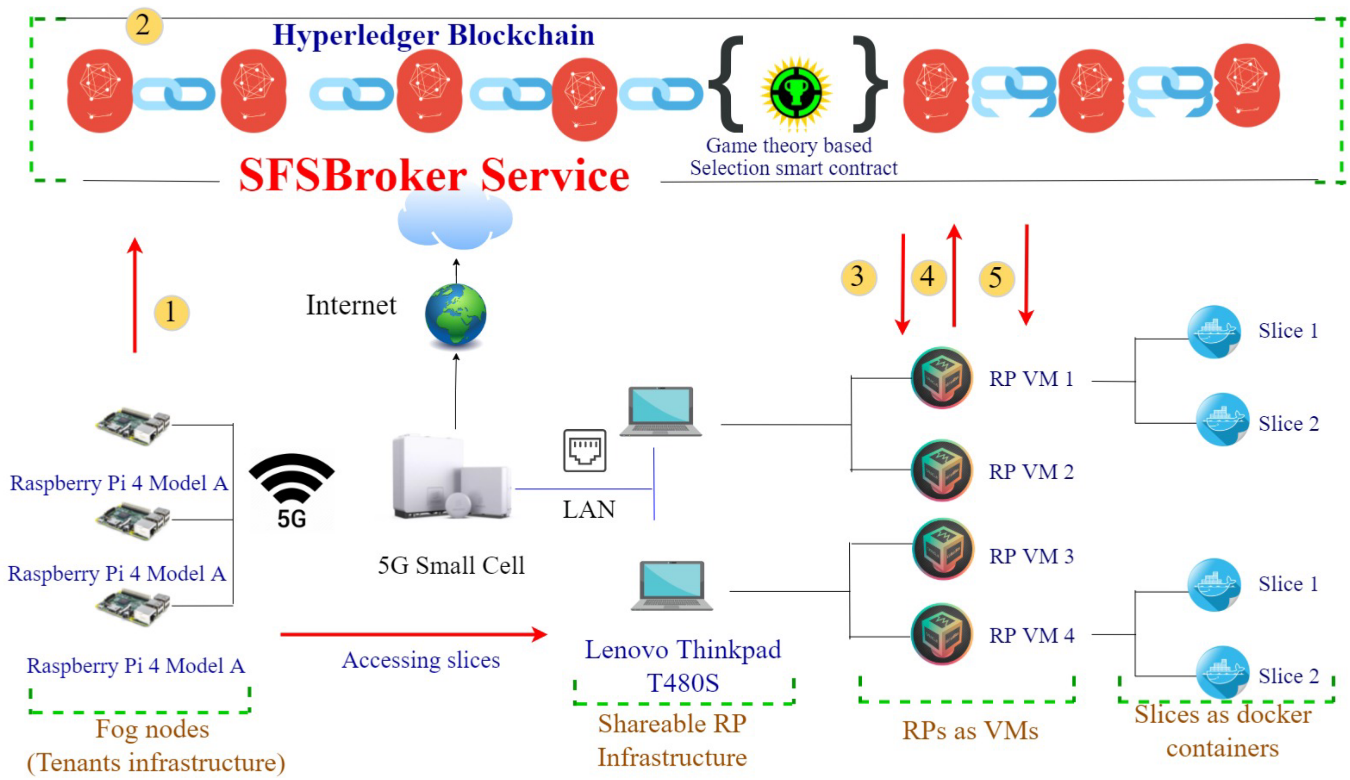


Fig. 3: The SFSBroker Testbed Implementation Setup

- 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 Section III-D 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.

D. *Performance Evaluation*

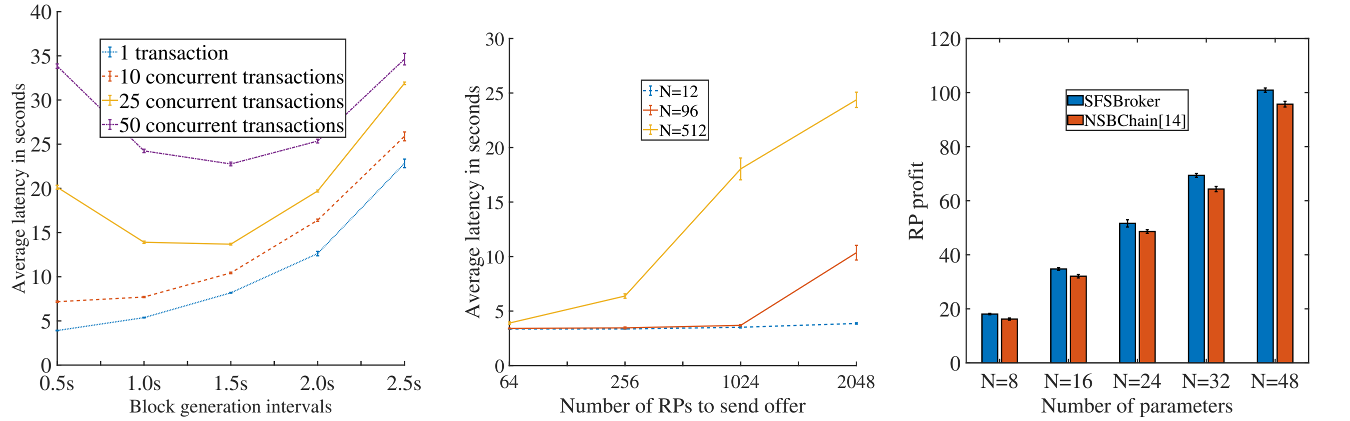
1) E2E Slice Creation Latency with Variable Block Time: This experiment demonstrates the multi-tenant scenario and the evaluation of E2E slice creation latency (steps 1 to 10 in Figure 2). E2E latency is measured with variable block generation time intervals as configured in Hyperledger. For a particular configured block generation time, the fog node initiates a set of concurrent transactions (i.e., 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 (CIs) (i.e., 95%) are computed (Figure 4a).

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 (i.e., 25 and 50) and low block generation time (i.e., 500ms 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.

2) 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 1000 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 4b, 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 2048 RPs and 512 parameters.

Since we need to evaluate the performance on different scales of the inputs, we increased the number of RPs upto 2048 in the experiment. We considered RPs and MNOs at the same ground and scaling upto 2048 RPs (Figure 4b) simulates the scenarios when local 5G operators deliver the services as RPs.



(a) The E2E latency of slice selection with different block mining time configurations

(b) E2E latency of slice selection for multiple RPs and different parameter counts

(c) Simulation result comparison of SFSBroker and NSBChain [14]

Fig. 4: Performance evaluation results in the implementation and simulation

E. 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 4c, 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.

V. Conclusion

In this work, we proposed a blockchain-based NS brokering mechanism (SFSBroker) for applications in the multi-operator multi-tenant 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.

References

[1] A. Antonopoulos, "Bankruptcy Problem in Network Sharing: Fundamentals, Applications and Challenges," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 81–87, 2020.

[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 Communications*, vol. 24, no. 5, pp. 118–125, 2017.

[3] N. Zhao, H. Wu, and Y. Chen, "Coalition game-based computation resource allocation for wireless blockchain networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8507–8518, 2019.

[4] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for 5G and beyond networks: A state of the art survey," *Journal of Network and Computer Applications*, p. 102693, 2020.

[5] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of Consensus Protocols on Blockchain Applications," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2017, pp. 1–5.

[6] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A Survey of Distributed Consensus Protocols for Blockchain Networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[7] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3602–3609, 2019.

[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 *2020 2nd 6G Wireless Summit (6G SUMMIT)*. IEEE, 2020, pp. 1–5.

[9] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, 2016.

[10] A. Boubendir, F. Guillemin, C. Le Toquin, M.-L. Alberi-Morel, F. Faucheux, S. Kerboeuf, J.-L. Lafragette, and B. Orlandi, "Federation of cross-domain edge resources: A brokering architecture for network slicing," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. IEEE, 2018, pp. 415–423.

[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 *2017 Internet of Things Business Models, Users, and Networks*. IEEE, 2017, pp. 1–8.

[12] K. Valtanen, J. Backman, and S. Yrjölä, "Creating value through blockchain powered resource configurations: Analysis of 5G network slice brokering case," in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 2018, pp. 185–190.

[13] N. Afraz and M. Ruffini, "5G network slice brokering: A distributed blockchain-based market," in *2020 European Conference on Networks and Communications (EuCNC)*. IEEE, 2020, pp. 23–27.

[14] L. Zanzi, A. Albanese, V. Sciancalepore, and X. Costa-Pérez, "NSBchain: A secure blockchain framework for network slicing brokerage," in *IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7.

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

[16] K. Antevski and C. J. Bernardos, "Federation of 5G services using distributed ledger technologies," *Internet Technology Letters*, p. e193, 2016.

[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," *Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11 339–11 351, 2017.

[18] E. Piri, P. Ruuska, T. Kanstrén, J. Mäkelä, J. Korva, A. Hekkala, A. Pouttu, O. Liinamaa, M. Latva-Aho, K. Vierimaa *et al.*, "5GTN: A Test Network for 5G Application Development and Testing," in *2016 European Conference on Networks and Communications (EuCNC)*. IEEE, 2016, pp. 313–318.

[19] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

Tharaka Hewa is a Doctoral Student in the Centre for Wireless Communication, University of Oulu, Finland. His research interests are Blockchain, PKI, 5G, Banking Systems Security, Healthcare Security, and Smart Cities.

Pawani Porambage is a postdoctoral researcher at the Centre for Wireless Communications, University of Oulu, Finland. Her main research interests are lightweight security protocols, security and privacy on IoT & MEC, and WSNs.

Anshuman Kalla is a Postdoctoral Visiting Researcher at the Centre of Wireless Communications, University of Oulu, Finland. His research interests include Blockchain, Future Networks, ICN, and IoT. For more info: https://sites.google.com/site/kallanshuman/

Diana P. M. Osorio is a Senior Research Fellow at the 6G Flagship Program, CWC, University of Oulu, Finland. She is also a Postdoctoral Researcher for the Academy of Finland and Adjunct Professor at the University of Oulu. Her research interests include wireless communications in general, 5G and 6G networks, and PHY security.

Madhusanka Liyanage is an Ad Astra Fellow/ Assistant Professor at the School of Computer Science, University College Dublin, Ireland. He is also an adjunct professor at the University of Oulu, Finland. His research interests are SDN, IoT, Blockchains, MEC, mobile and network security. More info: http://www.madhusanka.com

Mika Ylianttila (M. Sc, Dr.Sc, eMBA) is a full-time associate professor (tenure track) at the Centre for Wireless Communications - Networks and Systems research unit, University of Oulu, Finland. He leads NSOFT (Network security and softwarization) research group which studies and develops secure, scalable and resource-efficient techniques for 5G and beyond 5G and IoT systems.