Adaptive Broadcast Routing Assignment Algorithm for Blockchain Synchronization Services

Frank Yeong-Sung Lin[1], Chiu-Han Hsiao[1], Yean-Fu Wen[2], and Yang-Che Su[1]

[1]Department of Information Management
National Taiwan University
Taipei, Taiwan (R.O.C.)
yslin@im.ntu.edu.tw;
{d98725001; r06725052}@ntu.edu.tw

[2]Graduate Institute of Information Management
National Taipei University
New Taipei City, Taiwan (R.O.C.)
yeanfu@mail.ntpu.edu.tw

***Abstract*—Transactions and blocks must be synchronized among the blockchain miners on the Internet. Software-defined networking and network function virtualization techniques support dynamically assigning computing resources into servers of the core and edge clouds. In this paper, an adaptive broadcast algorithm is proposed for blockchain authentication, authorization, and accounting (AAA) services. The cryptography is propagated throughout the Internet by using a broadcast mechanism. The broadcast message may incur a propagation delay and duplicate transmissions. The total propagation delay is assumed to be a combination of transmission time and computational time for data verification. A mathematical programming model is formulated to address the secure broadcast problem as a minimum spanning tree problem. The objective is to minimize the processing and transmission delay through reduced duplicate transmissions. Computational experiments demonstrate proof of concept to adopt blockchain techniques. The dynamic AAA architecture and path selection enable the blockchain operator to efficiently make decisions and achieve more secure services.**

Keywords—Broadcast, AAA, Blockchain, Routing Assignment

I. INTRODUCTION

Blockchain has been proposed to duplicate transaction data for all miner devices on the Internet. Network resource management has been adapted through software-defined networking (SDN) and network function virtualization (NFV) into resource containerization [1]. SDN is a programmable mechanism enabling dynamic and flexible control of the routing path and link management for end-to-end communication. NFV is the concept that network functions can be transformed into software-based applications. Virtualized network functions (VNFs) are assigned to nonproprietary hardware [2]. This concept can be adopted for the blockchain broadcast method from the viewpoint of the overlay network and application development. With SDN enabled, the application can be informed the detail routing and traffic load information with the network status, which helps making overlay network forward node selection for application-level broadcast efficiently. .

Three main tasks of data processing have been introduced through the blockchain technique: i) transaction processing coordination by intermediary miners, ii) sessions of transaction processing monitoring, and iii) distributed writing of transaction data to blockchains [1]. Task i) converts requests into specification descriptions of transactions, such as transactions in Bitcoin and content authorization contracts. According to specification descriptions, the request is assigned and delivered to hosts with lower transmission costs. Task ii) ensures that transactions are correctly processed on the basis of specification descriptions. The transaction is completely verified using a digital signature for a successfully executed transmission. Task iii) requires each node to broadcast a completely processed block to other miners in the system. Task iii) is a competition among miners that only the first block is broadcast completely and immediately to be written into the ledger. The miner can receive rewards after all nodes receive the complete messages. This is called proof-of-work, indicating that each participant sends verified results to the blockchain.

The generated block is connected to the existing blockchain in the processes of constructing a distributed ledger. The host who first generates this block has a responsibility to broadcast the block to other hosts who must store this block in the network. Here, we defined “broadcast” as a process that a host intends to send newly generated transactions/blocks to all other hosts, which have joined the blockchain system as miner hosts. However, under most circumstances, unicast is used to send message since broadcast are not enabled on the Internet router or switch by default with the running network protocol, TCP/IP. Unicast messages cause many identical packets to be transmitted repeatedly; this may cause network congestion. This paper intends to reduce network traffic using SDN to improve transmission performance and efficiently achieve secure dynamic authentication, authorization, and accounting (AAA) services and path selection.

In this paper, we propose a new method adapted from the concept of database validation of Bitcoin applications of blockchain emulated as distributed AAA modules among miner hosts on the Internet. The accuracy of decentralized

block synchronization is critical. Maintaining the cryptography information of each miner host, including numerous of transaction, authentication, or authorization records, is also a challenge. According to the decentralized processing in overlay network topology, we propose an application-layer broadcasting method to analyze how the cryptography information is propagated in a message to all hosts in a shared network pool. A broadcasting tree is constructed as the overlaying network topology to improve the information validation capabilities with minimal delay. This includes limiting transmission and computation delays through appropriate transmission path selection.

This paper expands the concept of Bitcoin applications of blockchain to a new AAA system architecture of decentralized and virtualization machines. A literature survey of the blockchain concept of Bitcoin applications and its relationship to this work are explained in Section II. The problematic characteristics of network topology and the concept of database validation of blockchains are illustrated as a mathematical formulation in Section III. For routing path design or link assignment in operation and optimization, solutions are developed in Section IV. We present a proof of concept that demonstrates our approach with computational experiments in Section V. Finally, we conclude this paper and outline future research in Section VI.

II. Literature Review

A. Cryptocurrency of Blockchain

Blockchain is a technology allowing each host verifiable and permanent decentralized ledgers of recorded transactions between two parties. A popular application of this is cryptocurrency (e.g., Bitcoin). The following briefly describes the steps of blockchain operation [4].

1. A transaction is initialized when someone requests a transaction.

2. Transactions are combined into a block broadcast to P2P hosts.

3. Other hosts (miners) in the network verify the transaction.

4. Validation is performed for transactions combined as a data block.

5. New block is added to existing blockchain.

6. Transaction is completed.

These steps are a blockchain example of the overall procedures of Bitcoin transactions. The latest block becomes a part of the attached blockchain when validations are completed by another host on the Internet. Each block comprise numerous transactions encrypted using hash functions and encryption keys. Sets of metadata in the block header are recorded to the relationship and chained with other blocks. Any host on the Internet can jointly validate the transactions as a verification contributor and check whether a block is correct. This collaboration of hosts is called mining in Bitcoin applications. The list of transactions is a form of database. When using the blockchain, the block has a digital signature of a serial of processes (e.g., transactions and payments) that can be traced back to an individual for identification, verification, and validation. The block held by nodes is decentralized for sharing to each host. This decentralized system protects the blockchain from tampering, deletion, and revision [5].

To maintain the consistency of the ledger replicas, the hosts need to agree on the transaction. A broadcasting mechanism delivers a new block created by one of the nodes for tentatively committing transactions, and it synchronizes at regular intervals. The block is broadcast and distributed to all hosts for validation and verification [6]. In Bitcoin, a new block is mined every 10 minutes. The fastest node to complete the validation receives rewards or cryptocurrency amount [5]. Delays, including transmission and computation delays, are critical factors for miners.

B. Broadcasting and Quality-of-Service Routing

By considering other factors of broadcasting messages mentioned in [7], the quality-of-service (QoS) requirements of routing mechanisms can be classified into two categories:

• Link constraints are restrictions on the use of links to form a routing tree, such as link bandwidth, node capacity, or buffer.

• Path constraints (or tree constraints) are restrictions on the whole broadcasting tree delay (e.g., the maximum end-to-end delay from source to all destinations). The goal of QoS routing is to find a feasible path with sufficient available resources to address the QoS requirements for nodes in a network [8] and to achieve efficient resource use. Delay, bandwidth, delay jitter, throughput, and packet loss ratio are the QoS measurements of a routing strategy to broadcast a block to all miner hosts. In addition, a link cost in the broadcasting tree can be defined in dollars or as a function of the buffer or bandwidth use.

Studies have determined the available feasible paths of optimization problems and found the lowest-cost feasible solutions. Chen et al. [9] conducted a survey of various QoS routing algorithms, divided into three broad classes: i) source routing algorithms, ii) distributed routing algorithms, and iii) hierarchical routing algorithms; the authors proposed a QoS-aware multicast routing protocol for nonadditive metrics to discover a feasible path with sufficient requested link bandwidth and buffer space [9]. Khadivi et al. [10] introduced new single mixed metrics for multiconstraint routing. To adequately reduce routing complexity, QoS routing may discard some potentially useful information in the process. Nevertheless, from an operational standpoint, the aforementioned are typically considered. Bazlamaçcı and Hindi [11] and Pettie and Ramachandran [12] proposed a definition of a minimum spanning tree (MST) or minimum weight spanning tree involving the allocation of an undirected spanning tree. The sum of the weights of the selected edges is minimized. When an MST is used in networks, considering QoS issues is necessary. Similar to the shortest path problem subject to QoS constraints for the routing tree, the MST problem is an NP-hard problem.

III. MATHEMATICAL FORMULATION

A. Problem Description

According to the proposed AAA architecture, blocks or transactions are introduced as VNFs initialized in virtual machines (VMs). The network topology is initialized by a VM. The information of the block, such as user identification, authentication, authorization, and delegation, is signed digitally using public key cryptography [6]. The cryptography is propagated throughout the network topology using a broadcast mechanism. Each recipient can validate the transactions using the private key. The public keys are used for authentication and identification. The broadcast message may incur a propagation delay. The total delay is assumed to be a combination of transmission and process time along the path.

Broadcasting is an automatic communication technique for all miner hosts to consistently validate important data, such as transactions and ledgers, related to blockchain applications. However, the confidentiality and validity of the broadcast environment should be considered. Maintaining decentralized functional verification and performance is difficult, which has is NP-hard. This problem can be abstractly constructed and modeled with a broadcast tree model with resource allocations and routing assignment problems.

B. Problem Formulation

We propose a broadcasting scheme to analyze how the cryptography information adopted by blockchain propagates the message to every node on the Internet. The network topology can be constructed by a cost of a broadcasting tree to improve the information validation capabilities with minimum delay (including transmission and processing delay) from the viewpoint of the overlay network. The following is the mathematical formulation. The given parameters and the decision variables are shown in Tables I and II, respectively.

TABLE I. GIVEN PARAMETERS

Notation	Description
B	All the blocks {1,2,3,..,b} that requires to be broadcasting to the miner hosts.
V	The set of nodes {1,2,3,…,v} in the overlay network
L	The set of links {1,2,3,…,l} in the overlay network.
r_b	The multicast root of block $b \in B$, where $r_b \in V$.
D_b	The set of destinations for block $b \in B$, while $D_b \in V - \{r_b\}$.
P_{bd}	The set of candidate paths that destination d of block b, while $d \in D_b$ and $b \in B$
δ_{pl}	Indicate function that 1 if link l is on path p, 0 otherwise.
I_v	The incoming links to node v, while $I_v \in L$.
h_b	The minimum number of hops to the farthest destination node for sending block b.
e_l	End node on link $l \in L$, while $e_l \in V$.
α_l	Transmission cost on link $l \in L$.
β_{e_l}	Processing cost on end node $e_l \in V$.
γ_{bl}	Penalty cost on link $l \in L$ for block $b \in B$.
w	The weight on previous penalty.
C_l^E	Penalty cost on link $l \in L$.

TABLE II. DECISION VARIABLES

Notation	Description
x_p	1 if path $p \in P_{bd}$ is selected for block b destined for destinations d, and 0 otherwise.
y_{bl}	1 if block b adopt link l, and 0 otherwise.

Objective function:

$$\min \sum_{b \in B} \sum_{l \in L} (\alpha_l + \beta_{e_l} + \gamma_{bl}) y_{bl} \qquad \text{(IP)}$$

Subject to:

Each block b is chosen to "adopt" (equal to 1) or "not adopt" (equal to 0) link l, as shown in (1).

$$y_{bl} = 0 \text{ or } 1 \qquad \forall b \in B, l \in L \qquad (1)$$

For each block b, the number of links being adopted should be larger than hopping times to the farthest node and the number of destination nodes, as shown in (2).

$$\sum_{l \in L} y_{bl} \geq \max\{h_b, |D_b|\} \qquad \forall b \in B \qquad (2)$$

For each block b, the number of incoming links of every destination node should be equal to or smaller than 1, as shown in (3).

$$\sum_{l \in I_{d_b}} y_{bl} \leq 1 \qquad \forall b \in B \qquad (3)$$

For each block b, the number of incoming links of root nodes should be 0, as shown in (4).

$$\sum_{l \in I_{r_b}} y_{bl} = 0 \qquad \forall b \in B \qquad (4)$$

For each block b broadcast to destination d, there is only one path adopted, as shown in (5).

$$\sum_{p \in P_{bd}} x_p = 1 \qquad \forall b \in B, d \in D_b \qquad (5)$$

For each block b broadcast to destination d, there are many paths that can be adopted. For each path p, we choose to "adopt" (equal to 1) or "not adopt" (equal to 0), as shown in (6).

$$x_p = 0 \text{ or } 1 \qquad \forall b \in B, p \in P_{bd}, d \in D_b \qquad (6)$$

For every block b broadcast to destination d, if path p is adopted, then all the links on path p should be set to 1 (is adopted), as shown in (7).

$$\sum_{p \in P_{bd}} x_p \delta_{pl} \leq y_{bl} \qquad \forall b \in B, l \in L, d \in D_b \qquad (7)$$

For each block b broadcast to all the destination nodes, if link l has been adopted, then the total times l has been

adopted should less than the number of destination nodes, as shown in (8).

$$\sum_{d\in D_b}\sum_{p\in P_{bd}} x_p \delta_{pl} \le |D_b| y_{bl} \qquad \forall b \in B, l \in L \qquad (8)$$

To find the penalty costs for block b, we linearly combine the link's penalty (if it is been adopted in block b - 1) and the penalty cost of block b - 1, as shown in (9).

$$\gamma_{bl} = wC_l^{\mathrm{E}} \cdot y_{(b-1)l} + (1-w)\gamma_{(b-1)l} \qquad \forall b \in B, l \in L \qquad (9)$$

IV. Solution Approach

A. Minimum Spanning Tree

To find a minimum cost for our objective function, we apply our model as an MST problem. An MST is a subset of the edges of a connected edge-weighted undirected graph. As a type of spanning tree, an MST has the following characteristics:

- The tree connects all the vertices.
- It has no cycle.
- If the graph has n vertices, then each tree has n-1 edges.
- The word "minimum" means the total edge weights should be minimized. That is, an MST is a spanning tree whose sum of edge weights is as small as possible.

Fig. 1 shows a connected undirected weighted graph. After applying the MST algorithm, we can find an MST for this graph. Fig. 2 shows a minimum total edge-weight spanning tree that connects all the nodes in the graph.

B. Solution Procedure

In our mathematical formulation, the maximum aggregated delay on a link is a combination of transmission delay and process delay among source node to destination node along the tree. However, the confidentiality and validity of the broadcast environment should be considered. By adding a penalty on the link after it is used, the solution approach would avoid to pick the same link for further broadcast, which gives a randomized topology for every broadcast block.

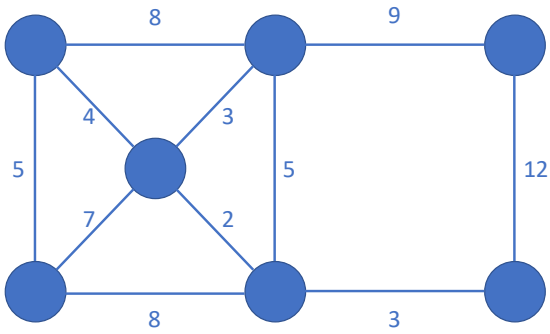

Fig. 1. Network Topology

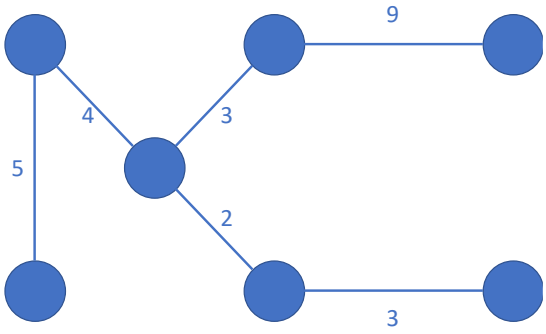

Fig. 2. Minimum Spanning Tree

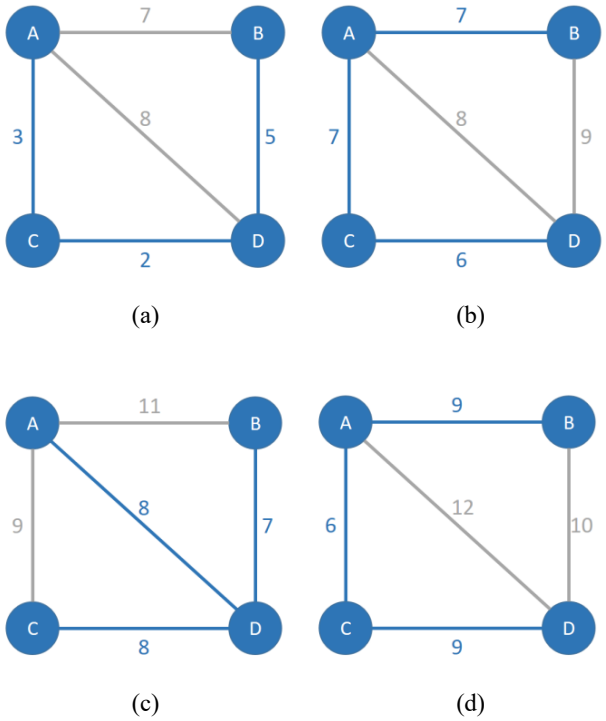

Fig. 3. MST for every broadcast block

Fig. 3 illustrates how a randomized topology be conducted. Fig. 3(a) shows an MST for the first broadcast block, the total cost on a link would be a combination of transmission cost and processing cost, without duplicate penalty. But In Fig. 3(b), a duplicate penalty is added to the link which is selected in Fig. 3(a), hence a different topology is applied. The penalty in Fig. 3. is calculated by using constraint (9) in section III. B., with w=0.5 and C_l^E=8. In Fig. 3(c), The link connects A with B and A with C are no longer be selected, because the total cost after adding penalty cost makes it not a better choice. The duplicate penalty causes the diverse topologies in Fig. 3(a)-(d), which produces a randomized routing path and high secure broadcast environment.

Based on the proposed formulation, we consider every block broadcast as an individual MST problem. The weight

on each link is a combination of transmission cost, processing cost, and penalty cost for re-using the link. For experiments in Section V., we use Prim's algorithm to solve every individual MST problem in a consequent block broadcast, and obtain a total objective value as experiment result. By setting various parameters, the relationships between the objective value and parameters settings can be found, which can have a further discussion.

V. COMPUTATIONAL EXPERIMENTS

A. Environments

In this paper, we conduct several experiments to verify our proposed model and compare the results under different circumstances. A self-implemented experiment program is developed using Python. To compare the differences among our experiments, we assign fixed values to some given parameters in the model (Table I). Table III shows the attributes of the given parameters used in the experiments. The transmission costs of each link and computing cost of each node are randomly assigned by multiplying a random number with the transmission weight and computing weight, respectively. The values assigned in Table III are the default values for every parameter if not used as an independent variable in the following experimental cases.

TABLE III. GIVEN PARAMETERS FOR EXPERIMENTS

Given Parameter	Value
All the blocks that need to be broadcasted	10
The set of nodes in the network	10
The ratio of core nodes and edge nodes	0.3~0.7
Repeat penalty's weight	0.5
Transmission weight	300
Compute weight	100
Repeat weight	100
Transmission cost for every edge	Random
Computing cost for every node	Random

B. Performance Evaluation

This section describes the three experiments conducted in this paper.

1) Case A:

The goal is to identify the relationship between the number of nodes and the objective value of the model. In the real world, the number of nodes can be considered the scale of system. An operator who has a larger scale system would have more nodes than those who operate in a small system. To conduct the experiment, we keep every parameter at the same value in each test. This case starts by tuning the number of nodes to 10, 20,…, 100. The results are shown in Fig. 4, which shows a trend of objective values monotonically increasing when the number of nodes increases. This trend is easy to understand because a bigger scale system means the operator have to cost more to operate the whole system.

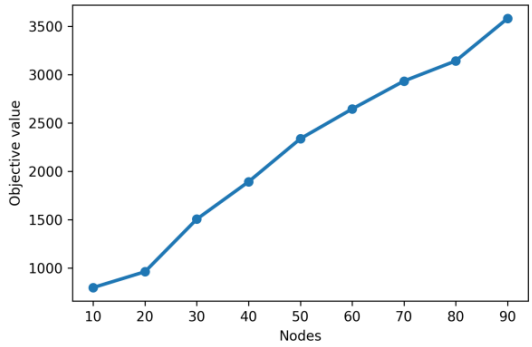

Fig. 4. Objective Value with Different Numbers of Nodes

2) Case B:

The goal is to understand the growth of costs when different core ratios are applied. In the experiment, all the nodes are separated into two types: core and edge nodes. In the real world, the system can also be separated into core and edge computing nodes. Core computing nodes have more powerful computing but higher transmission costs between nodes. Edge computing nodes are less powerful but have low transmission costs. We assign different core ratios to these two node types from 0.2 to 0.6 (edge ratios would be 0.8 to 0.4, respectively). The results (Fig. 5) provide information on the change in different types of cost mentioned in the model at different assigned ratios.

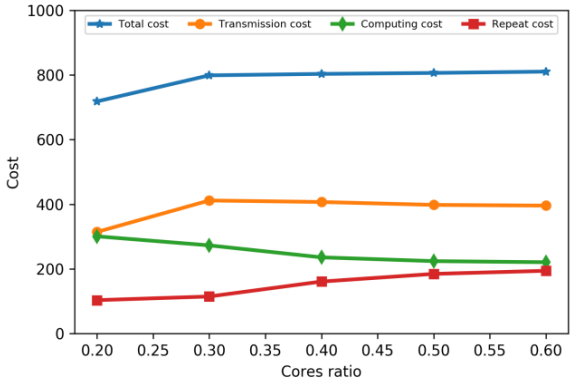

Fig. 5. Cost Distribution Evaluation

As the core ratio increases, so does the total cost, but it gradually reaches a maximum; transmission cost behaves in the similar manner, except it slightly decreases at the end. When the core ratio increases first, some nodes join to core nodes, thus increasing transmission costs; but the core ratio continually increases, the transmission costs of the two types of nodes balance as well as reach their maximum.

The computing cost decreases and reaches its minimum when the core ratio increases. The more edge nodes set as core nodes, the higher the computing power of the system

becomes. This causes the computing cost of the whole system to decline.

The repeat cost increases and gradually reaches a maximum. This shows that when the ratio of the two types of nodes becomes more balanced, its repeat cost increases. A lower core ratio would result in a lower repeat cost.

3) Case C:

The goal is to know increase in cost when various repeat penalty weights (w) are applied. In the real world, repeat penalty weights can be regarded as the urgency the operator places on the repeated use of a link. Fig. 6 shows the experimental results of assigning weights from 0.0 to 1.0. The repeat cost increases when w is small, but it gradually decreases when w becomes larger than 0.4. This phenomenon shows that repeat cost increases at first because the penalty weight increases. However, after the repeat cost is increased, the system will attempt to find another path to achieve a lower total cost. The repeat cost decreases when different paths are selected. This case is helpful for the operator of the network system because the result provides guidance on values to be assigned.

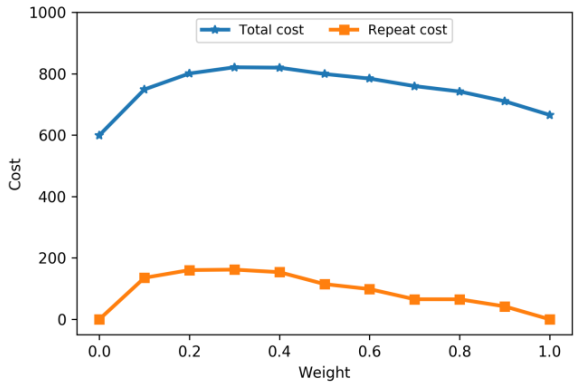

Fig. 6. Repeating Link Cost Evaluation

VI. Conclusions

The resource management of blockchain has been proposed to adapt SDN and NFV techniques into a cloud infrastructure distributed in edge and core cloud environments. Our proposed solution is to develop a novel networking method to make broadcast forward nodes assignment in shared network architecture for AAA services, blocks, or transactions that are introduced as the VNFs initialized in VMs. AAA services are provided by orchestrating the VNFs to use SDN and NFV techniques for resource management dynamically and automatically. The network topology is initialized by one of the VMs.

The cryptography is propagated throughout the network topology using a broadcast mechanism. The broadcast message may incur a propagation delay. The total propagation delay is assumed to be a combination of transmission time and computational time for data verification. In this paper, we presented a proof of concept that demonstrates our approach through computational experiments. The broadcasting and routing strategies were adopted with various topologies for the blockchain technology to minimize the processing and transmission delay through less reuse of transmission links.

Acknowledgment

This work was supported in part by the National Science Council of Taiwan (R.O.C.) (Grant Number MOST 106-2221-E-002-032 and 106-2221-E-305-004).

References

[1] S. Sharma, R. Miller, and A. Francini, "A Cloud-Native Approach to 5G Network Slicing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 120-127, August 2017.

[2] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks (vEPC)," *IEEE Network*, vol. 28, no. 6, pp. 18-26, December 2014.

[3] S. Wong, N. Sastry, O. Holland, V. Friderikos, M. Dohler, and H. Aghvami, "Virtualized Authentication, Authorization, and Accounting (V-AAA) in 5G Networks," in the *2017 IEEE Conference on Standards for Communications and Networking (CSCN 2017)*, Helsinki, Finland, September 2017, pp. 175–180.

[4] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008), http://www.bitcoin.org.

[5] https://bitcoinexchangeguide.com/blockchain-distributed-ledger-technology/, Retrieved from Internet, [4 February 2018]

[6] C. Decker and R. Wattenhofer, "Information propagation in the Bitcoin network," in *the 2013 IEEE Thirteenth International Conference on Peer-to-Peer Computing (P2P 2013)*, Povo Trento, Italy, September 2013, pp. 1-10.

[7] A. Striegel and G. Manimaran, "A Survey of QoS Multicasting Issues," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 82–87, June 2002.

[8] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture," *IEEE Network*, vol. 14, no. 1, pp. 78–88, February 2000.

[9] S. Chen, K. Nahrstedt, and Y. Shavitt, "A QoS-aware Multicast Routing Protocol," in *the IEEE Conference on Computer Communications (INFOCOM 2000)*, Tel Aviv, Israel, March 2000, pp. 1594–1603.

[10] P. Khadivi, S. Samavi, T. D. Todd and H. Saidi, "Multi-constraint QoS Routing Using a New Single Mixed Metric," in *the 2004 IEEE International Conference on Communications (ICC 2004)*, Paris, France, June 2004, pp. 2042–2046.

[11] C.F. Bazlamaçcı and K.S. Hindi, "Minimum-weight Spanning Tree Algorithms: A Survey and Empirical Study," *Computers & Operations Research*, vol. 8, no. 8, pp. 767–785, July 2001.

[12] S. Pettie and V. Ramachandran, "An Optimal Minimum Spanning Tree Algorithm," *Journal of the ACM*, vol. 49, no. 1, pp. 16–34, January 2002.