Proof-of-Balance: Game-Theoretic Consensus for Controller Load Balancing of SDN

Siyi Liao*†, Jun Wu*†, Jianhua Li*†, and Ali Kashif Bashir§
*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
†Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai, China
§Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, United Kingdom
{syliao, junwuhn, lijh888}@sjtu.edu.cn, {dr.alikashif.b}@ieee.org

***Abstract*—Software Defined Networking (SDN) focus on the isolation of control plane and data plane, greatly enhancing the network's support for heterogeneity and flexibility. However, although the programmable network greatly improves the performance of all aspects of the network, flexible load balancing across controllers still challenges the current SDN architecture.. Complex application scenarios lead to flexible and changeable communication requirements, making it difficult to guarantee the Quality of Service (QoS) for SDN users. To address this issue, this paper proposes a paradigm that uses blockchain to incentive safe load balancing for multiple controllers. We proposed a controller consortium blockchain for secure and efficient load balancing of multi-controllers, which includes a new cryptographic currency balance coin and a novel consensus mechanism Proof-of-Balance (PoB). In addition, we have designed a novel game theory-based incentive mechanism to incentive controllers with tight communication resources to offload tasks to idle controllers. The security analysis and performance simulation results indicate the superiority and effectiveness of the proposed scheme.**

***Index Terms*—Software Defined Networking (SDN), blockchain, game theory, load balance;**

I. Introduction

Software Defined Networking (SDN) decouples the control plane and data plane of the network and has been regarded as one of the mainstream architectures of the next generation network [1]. Due to the fine-grained network control, SDN has obvious advantages over traditional networks in many aspects, such as network management and security policy implementation [2]. SDN focuses on providing networking functionality by employing a logically centralized SDN controller that communicates with programmable network devices. To ensure the Quality of Service (QoS) of SDN users, it is necessary to use idle communication resources to alleviate the resource tension of other controllers. Therefore, the secure and efficient load balance of multi-controllers in SDN is an important part of SDN QoS guarantee.

SDN load balancing is an efficient way to improve network performance, availability, and minimize latency and avoid network congestion. On the other hand, the operation and maintenance of SDN, the fast provision of services, and the development of technology need to rely on high efficient load balancing of SDN controllers. However, the trust multi-controllers cooperation and load balancing issues still challenges the deployment and application of SDN. The open standard and open source implementations of southbound SDN interfaces will significantly facilitate interoperability among different controllers. With the development of software defined technology, trusted and reliable SDN load balancing is worthy of attention [3]. Therefore, a high efficient and trust load balancing problem for SDN controllers needs to be solved urgently.

A blockchain is a distributed data structure that is replicated and shared among the members of a network. Compared with public blockchains, consortium blockchains have advantages in terms of efficiency, cost, flexibility, and privacy protection. The application of blockchain and smart contracts can realize distributed and trusted transactions, which are traceable and irreversible [4]. This provides opportunities and cooperation models for cross-controllers collaboration and interoperation in order to realize load balancing. Due to the decentralized and immutable characteristics of blockchain, the cooperation paradigm and security of different controllers can be guaranteed in the form of smart contracts. However, traditional public blockchains have certain shortcomings in terms of energy consumption, scalability, and transaction processing speed. The goal of this paper is to provide a detailed description of how blockchains and smart contracts work to realize and guarantee the efficient and secure load balance of different SDN controllers.

Therefore, motivated by some previous works, we exploit the blockchain technologies, lncluding consortium blockchain and consensus mechanism, to achieve load balance between controllers and to ensure the QoS of SDN users. The contributions of this paper are the following:

- Based on the consortium blockchain, a novel scheme for the secure and high efficient load balancing of the SDN controllers is proposed. Secure and reliable SDN services are provided between different controllers, and load balance transactions are written into the blockchain.
- A novel consensus mechanism Proof-of-Balance (PoB) and a new crypto currency called Balance Coin (BC) are proposed to encourage and motivate the load balancing between controllers.
- A two-stage Stackelberg game model is formulated for the load balancing problem of the SDN controllers to obtain the optimal strategies of service requesters and providers.

The rest of this paper is organized as follows. In Section II, we investigate the related papers about load balance and blockchain in SDN. The general overview and key components of the basic architecture are given respectively in Section III. Section IV illustrates specific mathematics methods and algorithms of the proposed scheme. Simulation and the preliminary results are provided in Section V.

II. Related Work

Recent studies have focused on security issues such as control, monitoring, and trust in SDN. [5] introduced a novel distributed SDN security architecture which integrated the efficiency of SDN control and monitoring with the resilience and scalability of a distributed system. [6] proposed and evaluated a joint entropy-based security scheme called JESS to protect the SDN by detect and mitigate DDoS attacks. At the same time, the security assurance of the control plane and the trustworthy cooperation and interoperability of multi-vendor controllers have been extensively studied. A novel transport SDN (T-SDN) framework is introduced in [7] to enable the flexible orchestration among vendor-diverse software and hardware components of functionally unified systems. Considering the different solutions and approaches that various vendors are offering, authors of [8] focused on the manage scenarios including multi-vendor and multi-owner setups of SDN. Authors of [9] presented an SDN-based underwater acoustic sensor networks (UASNs) framework and proposed a load balancing mechanism involving multiple controllers, based on the consistent hashing algorithm. Authors of [10] proposes a big data analysis-based secure cluster management architecture for the optimized control plane. A security authentication scheme is proposed for cluster management. In order to increase the robustness of the communications of control plane while enhancing their performance, authors of [11] proposed a modular secure SDN control plane communications architecture by decreasing the complexity of the support infrastructure.

The potential advantages of blockchain in terms of distribution provide a new perspective on the solution of SDN systems. Authors of [5] investigated the security and privacy issue in the transportation system and the vehicular IoT environment in SDN-enabled 5G-VANET and designed a blockchain-based security framework. A novel scheme for updating a flow rule table using a blockchains technique is proposed in [12] to securely verify a version of the flow rule table, validate the flow rule table, and download the latest flow rules table for the IoT forwarding devices. Along with detailed consensus steps and theoretical analysis, authors of [13] proposed a blockchain (BC)-based consensus protocol in SDIIoT, where BC works as a trusted third party to collect and synchronize network-wide views between various SDN controllers. A blockchain-based distributed controller forensic architecture in [14] is designed to use the Linear Homomorphic Signature (LHS) algorithm for validating users. Authors of [15] proposed a novel blockchain-based distributed software-defined VANET framework (block-SDV) to establish a secure architecture to improve the dynamicity and infrastructure-less of VANETs.

The strength of this paper is listed as follows. This paper takes the advantages of blockchain and designs a system for load balancing of SDN controllers. The proposed PoB consensus mechanism takes actual communication resources as a measure and encourages SDN load balancing through resource pricing, rather than wasting a lot of energy and computing resources like traditional PoW. To the best of our knowledge, no research has yet used consortium blockchain to encourage SDN load balancing.

III. Basic Architecture

A. Scenarios of SDN Controllers Load Balancing

As shown in Fig.2, SDN controllers from different vendors are widely distributed, and they cooperate to ensure efficient and fast data distribution. However, due to the changing communication resource requirements brought by complex and changeable applications, the QoS of users in an overloaded SDN will be greatly decreased. SDN load balancing can occur between SDN controllers belonging to the same vendor, or it can be a trusted controller from different vendors. Requester of load balancing (RLB) requests communication resources from providers of load balancing (PLB) and completes the transaction by paying Balance Coin (BC). The selected blockchain validator packs all communication resource transactions into a block over a period of time and joins it to the current blockchain. In addition to get BC from load balancing, the elected block validators can also get a reward for validating transactions in the block.

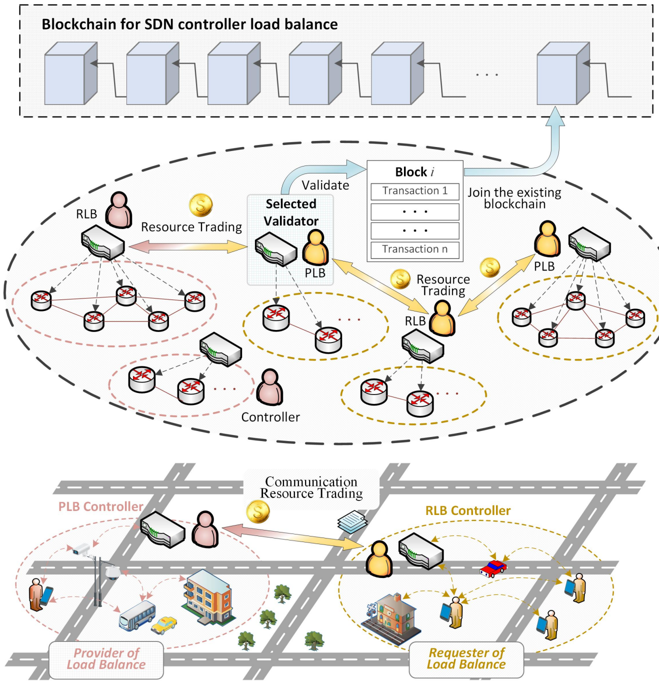


Fig. 1. Scenarios of load balancing between SDN controllers.

B. Proposed PoB Consensus Mechanism

Due to the large number of controllers, we elect representatives within the vendor to build the consortium blockchain. As shown in Fig.2, we first elect the top $\tau\%$ controllers (denote as $C_{i,j}$) to participate in the validator election based on the amount of controller load balancing in the past time $T = k \cdot t$.

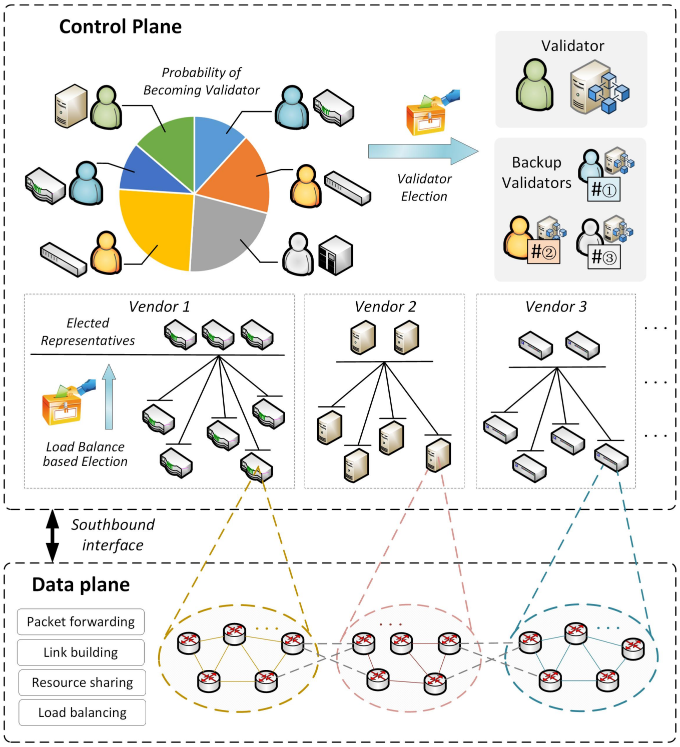


Fig. 2. Proof of Balance based block validator election.

After the representatives set $\mathbb{R}$, the total load balancing amount $Load_{i,j}$ of the controller j from vendor i during the past k time periods will be counted.

$$Load_{i,j} = \sum_{k=0}^{T_{i,j}/t} \lambda_{(i,j),k} \cdot t \quad (T_{i,j} = k \cdot t \in \mathbb{T}) \tag{1}$$

In $Eq.(1)$, t denotes the basic time period of load balancing. Therefore, we can get the probability that each representative becomes the final block verifier as follows.

$$P_{i,j} = \frac{Load_{i,j}}{\sum_i \sum_j Load_{i,j} \cdot \delta_{i,j}} \quad \begin{cases} \delta_{i,j} = 0 & C_{i,j} \in \mathbb{R} \\ \delta_{i,j} = 1 & C_{i,j} \notin \mathbb{R} \end{cases} \tag{2}$$

IV. Proposed Scheme

A. Stackelberg Game based Controller Load Balancing

1) Problem Formulation: In the load balancing scheme proposed in this paper, we divide the state of the SDN controllers into three types: 1) requester of load balancing (RLB, leader), 2) provider of load balancing (PLB, follower), and 3) not participate in load balancing. We denote the certain load balancing requester controller as C_{RLB}. According to the connectivity of the date plane devices, the potential controllers that can provide load balancing services are denoted as $\mathbb{P} = \{C_{PLB,1}, C_{PLB,2} \cdots C_{PLB,n}\}$. When C_{RLB} is fully loaded, the QoS of its users will decrease significantly, so it requests communication resources from $C_{PLB} \in \mathbb{C}$ according to the minimum load balancing demand D_{min}. Therefore, our goal is to satisfy the conditions of load balancing, while enabling RLB and PLB participating in load balancing to maximize their benefits.

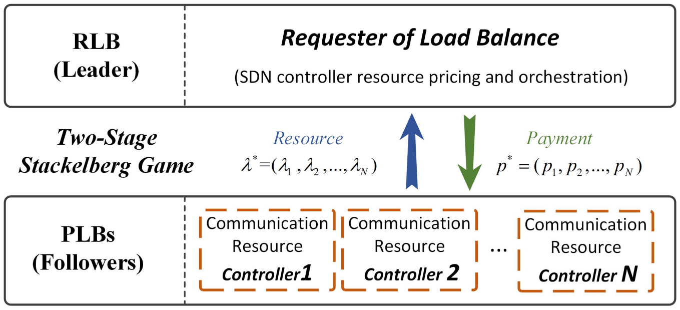

RLB (Leader)
Requester of Load Balance
(SDN controller resource pricing and orchestration)
Two-Stage Stackelberg Game
Resource
$\lambda^* = (\lambda_1, \lambda_2, ..., \lambda_N)$
Payment
$p^* = (p_1, p_2, ..., p_N)$
PLBs (Followers)
Communication Resource
Controller1
Communication Resource
Controller 2
...
Communication Resource
Controller N

Fig. 3. Two-stage Stackelberg Game based Controller Load Balancing.

In order to get the load balancing strategy between controllers and the corresponding pricing strategy, we propose the game $\mathbb{G} = \{S_{RLB}, S^{PLB}_{C_{PLB} \in \mathbb{P}}; \Pi_{RLB}, \Pi^{PLB}_{C_{PLB} \in \mathbb{P}}\}$. $S_{RLB} = \{\sum_{C_{PLB,i} \in \mathbb{P}} \lambda_i \geqslant D_{min}, p_{min} \leqslant p_i \leqslant p_{max}\}$ is the pricing strategy of RLB that is designed to meet its load balancing needs. $S^{PLB}_{C_{PLB} \in \mathbb{C}} = \{\lambda_i, i \in N, \eta \cdot L_{i,max} \leqslant \lambda_i \leqslant \mu \cdot L_{i,max}\}$, where λ_i represents the computing resources provided by $C_{PLB,i}$ to C_{RLB} and $\eta, \mu \in [0,1]$ limit the range of communication resources provided by $C_{PLB,i}$. In $\mathbb{G}$, Π_{RLB} and $\Pi^{PLB}_{C_{PLB} \in \mathbb{C}}$ are the utility functions of RLB and PLB. Given $\lambda^* = \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ and $P^* = \{p_1, p_2, \cdots, p_n\}$, the objective function of RLB can be formulated as follows.

$$\begin{aligned} & max \quad \Pi_{RBL}(\lambda^*, P^*) \\ s.t. \quad & \begin{cases} \sum_{i \in \mathbb{P}} \lambda_i \geqslant D_{min} & i \in \mathcal{N} \\ p_{min} \leqslant p_i \leqslant p_{max} \end{cases} \end{aligned} \tag{3}$$

On the other hand, taking into account of the load balancing limitations of PLB, the objective function of PLB can be formulated as follows.

$$\begin{aligned} & max \quad \Pi_{PBL}(\lambda_i) \\ s.t. \quad & \eta \cdot L_{i,max} \leqslant \lambda_i \leqslant \mu \cdot L_{i,max} \quad i \in \mathcal{N} \end{aligned} \tag{4}$$

2) Utility Function Formulation: For the RLB controller, its overloaded communication traffic will be load balanced by different PLB controllers. Correspondingly, the RLB will provide a certain reward for the communication resources of the PLB according to the load balanced by the PLB. We set the QoS obtained by RLB load balancing as a logarithmic expression according to [17]. Since RLB is in a full load state, we consider its expenditure on energy consumption to be constant C_E. Therefore, we have the following utility function for RLB.

$$\Pi_{RBL}(\lambda^*, P^*) = \sum_{i \in \mathbb{P}} \theta_i ln(1+\lambda_i) - p_i \cdot \lambda_i - C_E \tag{5}$$

On the other hand, the PLB controller needs to utilize part of the communication resources for RLB load balancing. Therefore, although it has received a reward from RLB, it needs to face the decline in QoS of its own users. In addition, more communication load means a corresponding increase in energy consumption. According to [17] [18], the utility function of PLB are formulated as:

$$\begin{aligned} \Pi_{PBL}(\lambda_i) = p_i \cdot \lambda_i + \theta_i ln(1+Load_i) - \\ \theta_i ln(1+\lambda_i) - (\alpha_i \cdot \lambda_i^2 + \beta_i \cdot \lambda_i + \gamma_i) \end{aligned} \tag{6}$$

where θ_i is a parameter used to indicate user QoS. For a communication systems, the utility of QoS is usually greater than energy, so $\alpha_i, \beta_i, and \gamma_i$ are a smaller number for θ_i they are all positive numbers.

3) Calculating Nash Equilibrium: Given that $\lambda^* = \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ is the load balancing strategy of PLB and $P^* = \{p_1, p_2, \cdots, p_n\}$ is the reward strategy of RLB, we have the following definition.

Definition 1: In $\mathbb{G} = \{S^{PLB}_{C_{PLB} \in \mathbb{P}}; \Pi^{PLB}_{C_{PLB} \in \mathbb{P}}\}$, λ^8 is the Nash Equilibrium (NE) of the game if $S_{PLB,i}$ if the best response to the non-cooperative sub-game and $\Pi_{PLB}(\lambda^*, p^*) \geqslant \Pi_{PLB}(\lambda', p^*)$.

Theorem 1: An unique NE of $\mathbb{G} = \{S^{PLB}_{C_{PLB} \in \mathbb{P}}; \Pi^{PLB}_{C_{PLB} \in \mathbb{P}}\}$ exists when $\lambda_i \geqslant \sqrt{\frac{\theta_i}{2 \cdot \alpha_i}}$.

Proof: The first derivative $\frac{\partial \Pi_{PLB}}{\partial \lambda_i}$ and the second derivative $\frac{\partial^2 \Pi_{PLB}}{\partial \lambda_i^2}$ of Π_{PLB} to λ_i are as follows.

$$\frac{\partial \Pi_{PLB}}{\partial \lambda_i} = p_i - \frac{\theta_i}{1+\lambda_i} - 2\alpha\lambda_i - \beta \tag{7}$$

$$\frac{\partial^2 \Pi_{PLB}}{\partial \lambda_i^2} = -2\alpha + \frac{\theta_i}{(1+\lambda_i)^2} \tag{8}$$

According to $Eq.(8)$, $\frac{\partial^2 \Pi_{PLB}}{\partial \lambda_i^2} \leqslant 0$ if $\lambda_i \geqslant \sqrt{\frac{\theta_i}{2 \cdot \alpha_i}}$. This means that the communication resources provided by each PLB controller have a lower limit. When λ_i is higher than this lower limit, Π_{PLB} is convex and an unique NE exist in $\mathbb{G} = \{S^{PLB}_{C_{PLB} \in \mathbb{P}}; \Pi^{PLB}_{C_{PLB} \in \mathbb{P}}\}$. In addition, we have the following price when $p_i - \frac{\theta_i}{1+\lambda_i} - 2\alpha\lambda_i - \beta = 0$.

$$p_i = \frac{\theta_i}{1+\lambda_i} + 2\alpha\lambda_i + \beta \tag{9}$$

Theorem 2: A unique Stackelberg Equilibrium (SE) exists in game $\mathbb{G} = \{S_{RLB}, S^{PLB}_{C_{PLB} \in \mathbb{P}}; \Pi_{RLB}, \Pi^{PLB}_{C_{PLB} \in \mathbb{P}}\}$ when $\theta_i > 2\alpha_i + \beta_i$. So that the pricing strategy $P^* = \{p_1, p_2, \cdots, p_n\}$ can maximize the benefits of RLB.

Proof: The first derivative $\frac{\partial \Pi_{RLB}}{\partial \lambda_i}$ and the second derivative $\frac{\partial^2 \Pi_{RLB}}{\partial \lambda_i^2}$ of Π_{PLB} to λ_i are as follows by using $Eq.(9)$.

$$\frac{\partial \Pi_{RLB}}{\partial \lambda_i} = \frac{\theta_i}{(1+\lambda_i)^2} - 2\alpha \cdot \lambda_i - 2\alpha - \beta \tag{10}$$

$$\frac{\partial^2 \Pi_{RLB}}{\partial \lambda_i^2} = \frac{-2\theta}{(1+\lambda_i)^3} - 2\alpha < 0 \tag{11}$$

According to $Eq.(11)$, the following Hessian matrix is a diagonal matrix.

$$H_{ij} = \begin{pmatrix} \frac{\partial^2 \Pi_{RLB}}{\partial \lambda_1 \partial \lambda_1} & \frac{\partial^2 \Pi_{RLB}}{\partial \lambda_1 \partial \lambda_2} & \cdots & \frac{\partial^2 \Pi_{RLB}}{\partial \lambda_1 \partial \lambda_n} \\ \frac{\partial^2 \Pi_{RLB}}{\partial \lambda_2 \partial \lambda_1} & \frac{\partial^2 \Pi_{RLB}}{\partial \lambda_2 \partial \lambda_2} & \cdots & \frac{\partial^2 \Pi_{RLB}}{\partial \lambda_2 \partial \lambda_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \Pi_{RLB}}{\partial \lambda_n \partial \lambda_1} & \frac{\partial^2 \Pi_{RLB}}{\partial \lambda_n \partial \lambda_2} & \cdots & \frac{\partial^2 \Pi_{RLB}}{\partial \lambda_n \partial \lambda_n} \end{pmatrix} \tag{12}$$

$$\frac{\partial^2 \Pi_{RLB}}{\partial \lambda_i \partial \lambda_j} = \begin{cases} \frac{-2\theta_i}{(1+\lambda_i)^3} - 2\alpha_i < 0 & i = j \\ 0 & i \neq j \end{cases} \tag{13}$$

In a SDN system, the aforementioned $\theta_i > 2\alpha_i + \beta_i$ limits the conditions under which PLB can participate in load balancing, that is, PLB can benefits from load balancing. Therefore, the unique SE of the game $\mathbb{G} = \{S_{RLB}, S^{PLB}_{C_{PLB} \in \mathbb{P}}; \Pi_{RLB}, \Pi^{PLB}_{C_{PLB} \in \mathbb{P}}\}$ exists and the load balancing strategy $\lambda^* = \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ is shown in the following $Eq.(14)$.

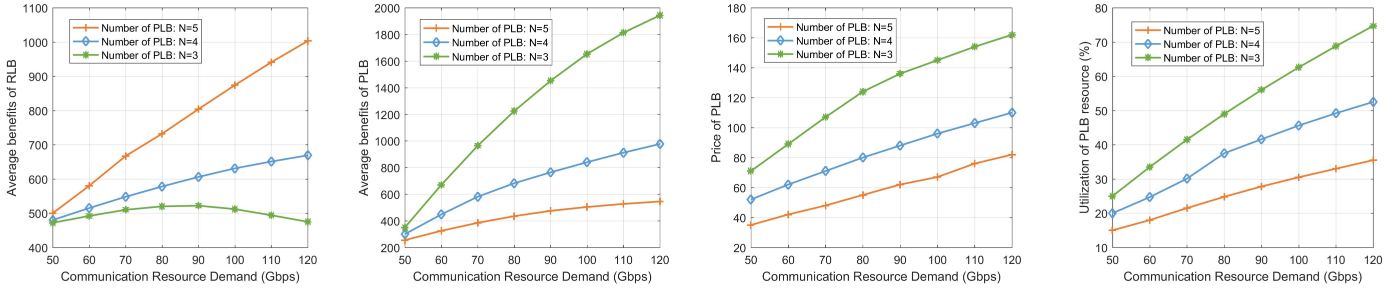
B. Game based Incentive PoB algorithm

Based on the PoB consensus mechanism and Stackelberg game proposed above, we have designed the following consensus algorithm to select the validator of the blockchain.

Algorithm 1 Game based Incentive PoB algorithm

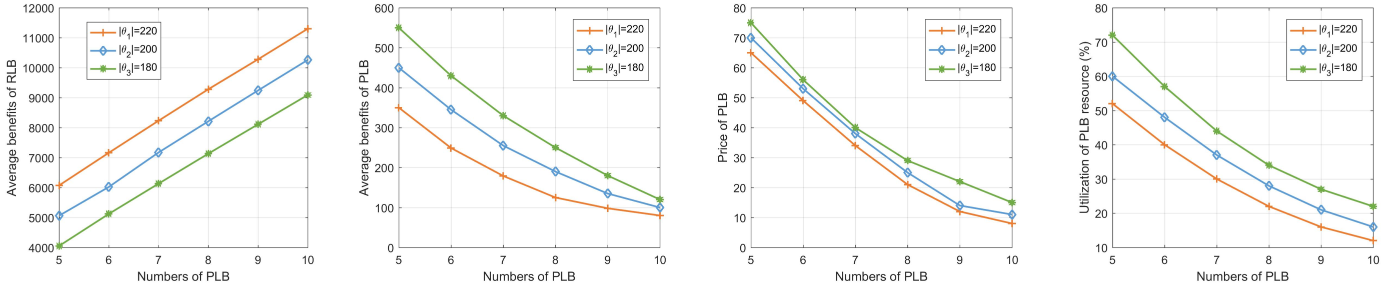
```
Input: Related parameters of $\Pi_{RLB}$ and $\Pi_{PLB}$, including
    $\theta_i$,$\alpha_i$,$\beta_i$,$\gamma_i$ and $Load_i$, representative election ratio $\tau$
Output: Validator $Vrep^*$ and backup validators
    $\{V_1^{'}, \cdots V_k^{'}\}$, strategy $\lambda^*$ and $p^*$.
Identify controllers that can participate load balancing
    for each $RLP$ and corresponding $PLB_i$ do
        Verify the conditions 1) $\lambda_i \geqslant \sqrt{\frac{\theta_i}{2 \cdot \alpha_i}}$ and
                              2) $\theta_i > 2\alpha_i + \beta_i$
        Calculate $\lambda_i$ using $Eq.(14)$ and $p_i$ using $Eq.(9)$
        Get the pricing strategy $p^*$ and balance strategy $\lambda^*$
    end for
Total load balancing of the controllers using $Eq.(1)$
    for each vendor $r_i \in \mathbb{R}$ joining PoB
        probability of representative $p = \frac{Load_j}{\sum_j Load_j}$
        Randomly select $\tau\%$ $Rep_{i,j}$ of vendor $r_i$
        Get the $P$ of each $Rep_{i,j}$ becoming $V$ using $Eq.(21)$
    end for
Select $V^*$ and $\{V_1^{'}, \cdots V_k^{'}\}$ by probability
$V^*$ Mortgage deposit and verify transactions
```

$$\lambda_i = \frac{6\alpha_i + \beta_i}{12\alpha_i} - \frac{2\alpha_i + \beta_i - \theta_i}{4\alpha_i} + \sqrt[3]{\sqrt{(\frac{6\alpha_i + \beta_i}{54\alpha_i}^3 - \frac{6\alpha_i + \beta_i}{12\alpha_i} + \frac{2\alpha_i + \beta_i - \theta_i}{4\alpha_i}) - (\frac{6\alpha_i + \beta_i}{18\alpha_i}^2 - \frac{1}{3})^3} - \frac{6\alpha_i + \beta_i}{54\alpha_i}}$$
$$- \frac{6\alpha_i + \beta_i}{6\alpha_i} + \frac{(\frac{(6\alpha_i+\beta_i)^2}{72\alpha_i}) - \frac{1}{3}}{\frac{6\alpha_i+\beta_i}{12\alpha_i} - \frac{2\alpha_i+\beta_i-\theta_i}{4\alpha_i} + \sqrt[3]{\sqrt{(\frac{6\alpha_i+\beta_i}{54\alpha_i}^3 - \frac{6\alpha_i+\beta_i}{12\alpha_i} + \frac{2\alpha_i+\beta_i-\theta_i}{4\alpha_i}) - (\frac{6\alpha_i+\beta_i}{18\alpha_i}^2 - \frac{1}{3})^3} - \frac{6\alpha_i+\beta_i}{54\alpha_i}}} \quad (14)$$



(a) Benefit of RLB (b) Benefit of RLB (c) Resource Price (d) Resource Utilization

Fig. 4. Comparison of cache hit ratio under different cache optimization methods



(a) Benefit of RLB (b) Benefit of RLB (c) Resource Price (d) Resource Utilization

Fig. 5. Comparison of cache hit ratio under different cache optimization methods

V. Evaluation

In this section, we simulate our proposed load balancing mechanism and game based PoB consensus mechanism. We assume that each RLB controller that needs communication load balancing can find several PLB controllers that are willing to assist in offloading its communication load. In our experiment, we set $\eta = 0.10$ and $\mu = 1.00$. This means that each controller participating in load balancing needs to reserve at least 10% of communication resources to meet its own user needs.

First, we explore the impact of the number of controllers participating in load balancing on PLB and RLB, including 1) the benefits of RLB 2) the benefits of PLB 3) the price of communication resource and 4) the utilization of PLB controllers. The result of Fig.4(a) shows that if more PLBs participate in load balancing, the benefits of RLB will be higher. However, when there are fewer nodes participating in load balancing while the demand for communication resources is relatively greater, the benefits of RLB will decline. Fig.4(b) and Fig.4(c) show that when more controllers participate in load balancing, the average benefits of PLB will be lower, and the price of communication resources will decrease. Fig.4(d) shows that when the demand for communication resources increases, the utilization of PLB participating in load balancing will increase correspondingly. And the fewer nodes participating in load balancing, the higher the average resource utilization of PLB.

Secondly, the impact of the user QoS parameter θ on the aforementioned 4 aspects are investigated. For this simulation, we suppose there are N = 4 PLBs participating in RLB load balancing and $|\theta_i| = \frac{1}{4}\sqrt{\theta_{i,1}^2 + \theta_2^{i,2} + \theta_{i,3}^2 + \theta_{i,4}^2}$. A larger $|\theta_i|$ indicates that the communication resources have a greater impact on the user's QoS. Fig.5 shows that when $|\theta_i|$ gets larger, the revenue of RLB will be higher. Meanwhile, the revenue of PLB will decrease, and the price of PLB communication resources will decrease and the utilization of communication resources will decrease. Meanwhile, the price of resources and the benefits of PLB will also decrease with the increase in the number of controllers participating in load balancing, and the benefits of RLB will increase accordingly.

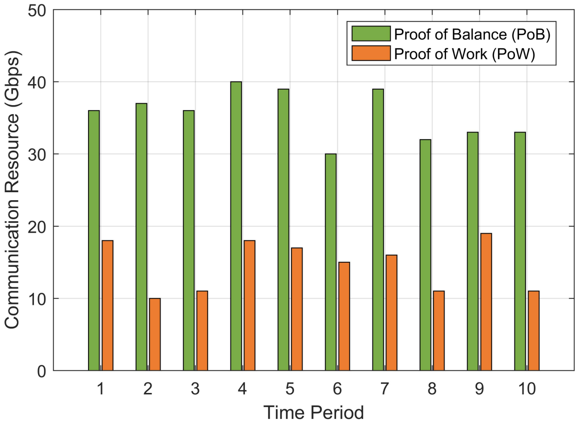


Fig. 6. PoB performance in terms of load balancing.

Finally, we compare the PoB consensus mechanism proposed in this paper with the traditional Proof-of-Work (PoW). We compare the performance of the transaction validators elected by the two consensus mechanisms in terms of load balancing. As shown in Fig.6, since the controller that contributes more communication resources for load balancing will have a greater probability to become a validator, the V^* elected by PoB performs significantly better in load balancing. On the contrary, the validators selected by PoW sometimes do not even participate in load balancing.

VI. Conclusion

In this paper, we focused on using blockchain to encourage sharing of communication resources between SDN controllers. For the load balancing of SDN controllers, a consortium blockchain based consensus mechanism is proposed, including a novel consensus mechanism PoB and a new cryptographic currency balance coin. In addition, we modeled a Starkberg game for load balancing between SDN controllers. Simulation results indicated the advantages and efficiency of the proposed scheme.

Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant No. 61972255).

References

[1] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A Survey of Security in Software Defined Networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 623-654, 2016.

[2] C. Janz, L. Ong, K. Sethuraman, and Vishnu Shukla, " Emerging transport SDN architecture and use cases," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 116-121, 2016.

[3] M. C. Dacier, H. Knig, R. Cwalinski, F. Kargl, and S. Dietrich, " Security Challenges and Opportunities of Software-Defined Networking," *IEEE Security & Privacy*, vol. 15, no. 2, pp. 96-100, 2017.

[4] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F. Wang, " Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2266 - 2277, 2019.

[5] L. Xie, Y. Ding, H. Yang, and X. Wang, " Blockchain-Based Secure and Trustworthy Internet of Things in SDN-Enabled 5G-VANETs," *IEEE Access*, vol. 7, pp. 56656 - 56666, 2019.

[6] K. Kalkan, L. Altay, G. Gr, and F. Alagz, " JESS: Joint Entropy-Based DDoS Defense Scheme in SDN," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2358 - 2372, 2018.

[7] C. Janz, L. Ong, K. Sethuraman, and V. Shukla, " Emerging Transport SDN Architecture and Use Cases," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 116 - 121, 2016.

[8] F. Granelli, A. A. Gebremariam, M. Usman, F. Cugini, V. Stamati, M. Alitska, and P. Chatzimisios, " Software Defined and Virtualized Wireless Access in Future Wireless Networks: Scenarios and Standards," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 26-34, 2015.

[9] J. Wang, S. Zhang, W. Chen, D. Kong, X. Zuo, and Z. Yu, " Design and Implementation of SDN-Based Underwater Acoustic Sensor Networks with Multi-Controllers," *IEEE Access*, vol. 6, pp. 25698 - 25714, 2018.

[10] J. Wu, M. Dong, K. Ota, J. Li, and Z. Guan, " Big Data Analysis-Based Secure Cluster Management for Optimized Control Plane in Software-Defined Networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 27 - 38, 2018.

[11] D. Kreutz, J. Yu, P. Esteves-Verssimo, C. Magalhes, and F. M. V. Ramos, " The KISS Principle in Software-Defined Networking: A Framework for Secure Communications," *IEEE Security & Privacy* , vol. 16, no. 5, pp. 60 - 70, 2018.

[12] P. K. Sharma, S. Singh, Y. S. Jeong, J. H. Park, " DistBlockNet: A Distributed Blockchains-Based Secure SDN Architecture for IoT Networks," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 78 - 85, 2017.

[13] C. Qiu, F. R. Yu, H, Yao, C. Jiang, F. Xu, and C. Zhao, " Blockchain-Based Software-Defined Industrial Internet of Things: A Dueling Deep Q -Learning Approach," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4627 - 4639, 2019.

[14] M. Pourvahab, G. Ekbatanifard, " An Efficient Forensics Architecture in Software-Defined Networking-IoT Using Blockchain Technology," *IEEE Access*, vol. 7, pp. 99573 - 99588, 2019.

[15] D. Zhang, F. R. Yu, R. Yang, " Blockchain-Based Distributed Software-defined Vehicular Networks: A Dueling Deep Q-Learning Approach," *IEEE Transactions on Cognitive Communications and Networking*, DOI: 10.1109/TCCN.2019.2944399, 2019.

[16] L. Fawcett, S. Scott-Hayward, M. Broadbent, A. Wright, and N. Race, " Tennison: A Distributed SDN Framework for Scalable Network Security," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 12, pp. 2805 - 2818, 2018.

[17] B. Gu, M. Dong, C. Zhang, Z. Liu, and Y. Tanaka, " Real-time Pricing for On-demand Bandwidth Reservation in SDN-enabled Networks," *IEEE Annual Consumer Communications & Networking Conference (CCNC)*, DOI: 10.1109/CCNC.2017.7983216, 2017.

[18] R. Spangler and R. Shoults, " Power Generation, Operation, and Control[Book Review]," *IEEE Power and Energy Magazine*, vol. 12, no. 4, pp. 90-93, 2014