

Evolutionary Game for Consensus Provision in Permissionless Blockchain Networks with Shards

Zhengwei Ni*, Wenbo Wang*, Dong In Kim[†], Ping Wang[‡] and Dusit Niyato*

*School of Computer Engineering, Nanyang Technological University, Singapore 639798

[†]School of Information and Communication Engineering, Sungkyunkwan University (SKKU), Suwon, Korea 16419

[‡]Department of Electrical Engineering & Computer Science, York University, Toronto, Canada ON M3J 1P3

Abstract—With the development of decentralized consensus protocols, permissionless blockchains have been envisioned as a promising enabler for the general-purpose transaction-driven, autonomous systems. However, most of the prevalent blockchain networks are built upon the consensus protocols under the crypto-puzzle framework known as proof-of-work. Such protocols face the inherent problem of transaction-processing bottleneck, as the networks achieve the decentralized consensus for transaction confirmation at the cost of very high latency. In this paper, we study the problem of consensus formation in a system of multiple throughput-scalable blockchains with sharded consensus. Specifically, the protocol design of sharded consensus not only enables parallelizing the process of transaction validation with sub-groups of processors, but also introduces the Byzantine consensus protocols for accelerating the consensus processes. By allowing different blockchains to impose different levels of processing fees and to have different transaction-generating rate, we aim to simulate the multi-service provision eco-systems based on blockchains in real world. We focus on the dynamics of blockchain-selection in the condition of a large population of consensus processors. Hence, we model the evolution of blockchain selection by the individual processors as an evolutionary game. Both the theoretical and the numerical analysis are provided regarding the evolutionary equilibria and the stability of the processors' strategies in a general case.

Index Terms—Permissionless blockchains, sharding, ELAS-TICO, evolutionary game

I. INTRODUCTION

The past decade has witnessed the fast development of the blockchain technologies, especially as the decentralized immutable ledger database (i.e., cryptocurrencies) in the FinTech sector. Most of the studies on blockchains have been focused on the development in cryptocurrencies and the related domains [1]. However, in recent years, more focus about blockchain applications is also put upon the domain of self-organization in general-purpose decentralized systems [2]. Given the decentralized consensus achieved by the blockchain network, smart contracts [3] are deployed in the form of general-purpose scripts/functions and stored on each consensus node (i.e., processor)¹ in the network. Following the order prescribed by the consensus about the blockchain states, each smart contract is executed in a replicated manner and thus guarantees to produce a uniform output across the network. Thanks to the technical maturation of smart contracts, blockchains are now envisioned as an enabler for self-organization in wireless networks and decentralized

¹We use the two terms, i.e., node and processor, interchangeably.

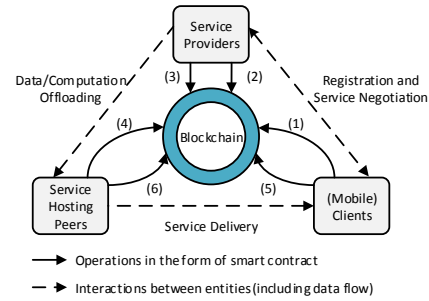


Fig. 1. A generic framework of blockchain-based self-organization in a service system of three parties. All the deals are settled in a sequence of smart contracts: (1) service requesting by the clients, (2) access granting by the providers, (3) requesting service hosting (e.g., auction for computation/storage/utility offloading) by the providers, (4) settlement of the hosting requests, (5) delivery negotiation between hosting peers and clients and (6) service completion and payment settlement upon proofs of delivery.

cyber-physical systems. More specifically, existing studies, e.g., autonomous access control [4] and service provision [5], employ blockchains as an integrator to channel the services upon demands as well as audit the operations of different parties in the system.

A generic paradigm for blockchain-based self-organization in networking applications is described by Figure 1 from the perspective of blockchain users. With the embedded cryptographic functionalities (e.g., asymmetric keys [2]) and the automated transactions based on smart contracts, blockchains are ready to provide the overlaid/virtual channels of secured data/service/payment delivery among trustless parties in the system [2]. As illustrated in Figure 1, this is achieved by encapsulating the controlling rules into smart contracts and the data (e.g., control signals) into blockchain transactions. In particular, when blockchain networks are deployed atop permissionless consensus protocols, it is possible to realize the scheme of network management in a purely decentralized manner. Furthermore, when the blockchain maintenance is delegated to groups of nodes with dedicated storage and computing power, we have the blockchain as a Platform as a Service (PaaS) in a similar way to that in the context of cloud computing.

Nevertheless, although permissionless blockchains provide a promising approach to transaction-driven automation for network control problems, most of the existing permissionless consensus protocols are based on Proof-of-Work (PoW) [1] and sacrifice the efficiency, i.e., transaction-processing throughput, for a higher level of consensus se-

curity [2]. For example, the popular Ethereum network [6] with a framework of Turing-complete smart contracts can only support less than 20 Transactions Per Second (TPS). As a result, these blockchains cannot satisfy the low-latency requirements in most of the networking applications and services such as access handing-off between groups of road side units in vehicular-to-infrastructure communication. To guarantee controller response in milliseconds, throughput-scalable protocols such as *ELASTICO* [7] are proposed to support both open access as in permissionless networks and low latency as in consortium distributed systems [8]. In brief, the throughput-scalable blockchain networks adopt the PoW-based crypto-puzzle design for node-identity verification and the classical Byzantine Fault-Tolerant (BFT) protocols (e.g., practical BFT [9]) for distributed transaction ordering. Furthermore, the concept of sharding is adopted from the distributed database [10] to enable the parallelization of transaction processing. Thus, the blockchain network is able to increase the TPS as the number of consensus processors increases.

In this paper, we investigate the scenario of general-purpose PaaS based on permissionless blockchains using sharding-based consensus protocols. In particular, we study the problem of consensus provision at the node level for multiple blockchains. Noting that the decentralized processors in a blockchain network are trustless, we assume that the processors are rational (i.e., profit-driven) and non-malicious. Namely, the independent and homogeneous processors participate in the consensus processes of parallel blockchains and dedicate their resources in exchange for the optimal consensus rewards, i.e., the transaction fees collected from the clients. For ease of exposition, we use the *ELASTICO* protocol [7] to exemplify the approach of system analysis for consensus participation. Without limiting the blockchains to any specific service provision system, we essentially study a general case of eco-system formation for self-organization with blockchains. Then, by formulating the behaviors of consensus nodes as an evolutionary game, we provide a series of analytical results regarding the equilibrium states and their stability in the evolution of the eco-system.

II. PRELIMINARIES OF SHARDED BLOCKCHAINS

A. Protocol Fundamentals of *ELASTICO*

A blockchain network using sharding protocol partitions the processing nodes in the network into smaller BFT committees. Each committee processes a disjoint set of transactions, i.e., a *shard*. The tasks of transaction processing are divided into multiple groups and done in parallel. Blockchains with shards overcome the fundamental scalability limits of many popular blockchain systems, e.g., Bitcoin [10]. The transaction processing rate is able to scale almost linearly with the number of processors in the network, which ensures that the requirements of real-time resource-access management systems can be met.

Now, we briefly introduce how *ELASTICO* [7] works. *ELASTICO* proceeds in loosely-synchronized epochs, each of which processes a set of transactions. According to [7], in each epoch, one processor mainly executes 5 procedures:

- 1) The processor is first required to solve PoW puzzles based on the concatenation of a public random seed, its own public key and Internet protocol (IP) address. This procedure prevents a processor to create pseudo-identities at no cost, and the puzzle solution can be considered as a public identity proof. The processor is randomly assigned to a committee based on the last few bits of its PoW solution. For example, assuming a total number of $2^2 = 4$ committees, if the last 2 bits of the PoW solution is "01", the processor will be assigned to committee 2 if this committee is not full. However, since the target committee to assign to may be full, the processor may need to solve more than one PoW puzzle.
- 2) Once a processor is accepted by the network and assigned to a committee, it will discover and establish point-to-point connections with other committee peers following an algorithm of decentralized randomness generation described in [7, Section 3.3].
- 3) Then, an authenticated Byzantine agreement protocol, e.g., practical BFT (PBFT) [9], is run within a committee to agree on the set of transactions (i.e. shard) allocated to it. Since different committees work in parallel, the network latency only depends on the number of processors in one committee rather than the entire network.
- 4) Once an agreement is reached in each committee, all the results will be merged. Then, the final result is broadcast to the network.
- 5) Finally, a scheme described in [7, Section 3.6] is executed by a global committee to randomly generate a new random seed for the next epoch.

B. Average Epoch Time for *ELASTICO*

As per the experimental results given by [7], the epoch time, i.e., the duration of one epoch, is mainly dominated by two parts, committee formation time and consensus time.

1) *Average committee formation time*: Committee formation time is the time used for randomly dividing processors into different committees. This time is mainly due to the cost of solving PoW puzzles. We assume that there are totally n processors in the blockchain network. They are divided into 2^s committees, with a fixed number of c processors in each committee. Thus, $n = 2^s c$. As we mentioned previously, one processor needs to solve more than one PoW puzzle if the originally assigned committee is full. The problem of calculating the total number of the required PoW solutions is equivalent to the extended coupon collector problem [11]. The expected number of PoW solutions is given in [7, Section 10.1]. When c is fixed, it has a superlinear relationship with n , which means that the expected number of PoW puzzles solved by one processor is increasing with n . In other words, when the number of processors per committee is fixed, if there are more processors in the blockchain network, one processor is expected to solve more PoW puzzles. In this paper, the expected number of PoW puzzles solved per processor is defined as a continuously differentiable and monotonically increasing function of n , $f(n)$, and we have $f(0) = 0$. Assume

that given a fixed puzzle difficulty, the average time for solving one PoW puzzle is T . Then, the average committee formation time can be expressed as $Tf(n)$.

2) *Average consensus time*: The consensus time is determined by the intra-committee agreement for the given shard and the inter-committee agreement for the final result. It is mainly due to the network latency, which is usually caused by the propagation delay of physical links, the forwarding latency of gateways, and the queueing and processing delays of intermediate nodes. In ELASTICO, we can observe that most of communications among processors are limited within the individual committees, so for a given committee size, the time to reach consensus remains almost constant for different network sizes. As [7, Figure 1] shows, the consensus requires 103 seconds for 400 processors and 110 seconds for 800 processors. Hence, we can think the consensus time is only dependent on the committee size c , which is denoted by $g(c)$.

C. Average Reward and Cost in ELASTICO

In this subsection, we quantitatively model the average reward and cost of one processor per epoch in ELASTICO.

1) *Average reward*: The processor receives a payment by adding new transaction records into the blockchain. We assume that the transaction records are generated by the users with a rate μ , and the price per transaction is set as r . Thus, the average reward of one processor per epoch is $\mu r(Tf(n) + g(c))/n$.

2) *Average cost*: The cost of one processor is dominated by the energy used for solving PoW puzzles. We assume that the cost of getting one PoW solution is ς on average. Thus, the average cost of one processor per epoch is $\varsigma f(n)$.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. Payoff Functions

We consider N individual processors organizing themselves into M blockchain networks built upon ELASTICO. That is, the processors choose to participate in one of the blockchain networks to receive their revenue. We assume that the processors have identical computing power and the average time for solving one PoW puzzle of fixed difficulty is T . We also assume that all blockchain networks adopt the same parameter of committee size c . We use the subscript i to denote other parameters for the i th blockchain network. Without loss of generality, the index of one blockchain network is determined by $\mu_i r_i$ in a descending order. That is, $\mu_1 r_1 \geq \dots \geq \mu_M r_M > 0$. The vector of population fractions of the blockchain networks is denoted by $\mathbf{x} = [x_1, \dots, x_M]^T$, where $[\cdot]^T$ is the notation of transpose. Thus, \mathbf{x} is in an $(M-1)$ -simplex, i.e., $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}_+^M : \sum_{i=1}^M x_i = 1\}$. We call \mathbf{x} *state vector* (or *state*) and \mathcal{X} *state space*.

According to Section II, the expected payoff per unit time (i.e., second) of a processor in the i th blockchain network can be expressed as

$$u_i(\mathbf{x}) = \frac{\frac{\mu_i r_i (Tf(Nx_i) + g(c))}{Nx_i + \tilde{\tau}} - \varsigma f(Nx_i)}{Tf(Nx_i) + g(c)}, \quad 0 \leq x \leq 1, \quad (1)$$

where $\tilde{\tau} > 0$ can be regarded as the share taken by network operators (e.g., the boosting nodes). By defining $\alpha_i = \mu_i r_i / N$, $\tau = \tilde{\tau} / N$, and $h(x_i) = \varsigma f(Nx_i) / (Tf(Nx_i) + g(c))$, $u_i(\mathbf{x})$ can be simplified as

$$u_i(\mathbf{x}) = \frac{\alpha_i}{x_i + \tau} - h(x_i), \quad 0 \leq x \leq 1. \quad (2)$$

We can easily obtain that $h(x_i)$ is monotonically increasing with x_i and $h(0) = 0$.

B. Dynamical System Formulation

In this process, some processors may switch from one blockchain network to another, causing a change of \mathbf{x} . Since the payoff of the processor is dependent on \mathbf{x} , other processors may also adjust their choice of consensus participation accordingly to choose new blockchain networks. Hence, in this paper, we study population fractions of the blockchain networks as a dynamical system. The state vector at time t is denoted by $\mathbf{x}(t) = [x_1(t), \dots, x_M(t)]^T$, and we define $\mathbf{x}_0 = \mathbf{x}(0)$ as the initial state. At time t , the rate at which the population fraction of i th blockchain network grows is $\frac{dx_i(t)}{dt}$, and we define $\dot{\mathbf{x}}(t) = \left[\frac{dx_1(t)}{dt}, \dots, \frac{dx_M(t)}{dt} \right]^T$. We assume that all the processors are bounded rational and self-interested, so the forces regulating the state vector are from the difference of payoffs. That is, the processors always switch from a blockchain network with low payoff to one with high payoff. Since the payoffs at time t are determined by $\mathbf{x}(t)$, $\dot{\mathbf{x}}(t)$ can be described by a function of $\mathbf{x}(t)$, here defined as $\varphi(\cdot) : \mathcal{X} \rightarrow \mathbb{R}^M$. Thus, this dynamical system is described by the following ordinary differential equations (ODEs):

$$\dot{\mathbf{x}}(t) = \varphi(\mathbf{x}(t)), \quad \forall t \in \mathbb{R}, \quad i = 1, \dots, M. \quad (3)$$

Specially, we adopt the following replicator equations [12]:

$$\dot{\varphi}_i(\mathbf{x}) = x_i (u_i(\mathbf{x}) - \bar{u}(\mathbf{x})), \quad (4)$$

where $\bar{u}(\mathbf{x}) = \sum_{i=1}^M x_i u_i(\mathbf{x})$, which can be regarded as the average payoff. Notice that here we ignore time t since $\varphi(\cdot)$ is autonomous, that is, does not depend explicitly on time. We can easily find $\sum_{i=1}^M \dot{\varphi}_i(\mathbf{x}) = 0$, so that if $\mathbf{x}_0 \in \mathcal{X}$, we always have $\mathbf{x}(t) \in \mathcal{X}$ for any $t \in \mathbb{R}$.

We are interested in how the vector of population fractions, i.e., state vector, changes with time for different initial states. Usually, it is described by a function $\xi(\cdot, \mathbf{x}_0) : \mathbb{T} \rightarrow \mathcal{X}$, where \mathbb{T} is an open interval containing $t = 0$, such that $\xi(0, \mathbf{x}_0) = \mathbf{x}_0$, and $\forall t \in \mathbb{T}$,

$$\frac{d}{dt} \xi(t, \mathbf{x}_0) = \varphi(\xi(t, \mathbf{x}_0)). \quad (5)$$

The function $\xi(\cdot, \mathbf{x}_0)$ is called a *solution* of (3).

IV. ANALYSIS OF THE GAME

A. Uniqueness of Solutions for Different Initial Points

In this subsection, we show that $\forall \mathbf{x}_0 \in \mathcal{X}$, we have a unique solution $\xi(\cdot, \mathbf{x}_0)$. It means that once the initial state is determined, how the population fractions evolve over time is totally determined. It is stated in the following theorem.

Theorem 1. For $\varphi(\cdot) : \mathcal{X} \rightarrow \mathbb{R}^M$ described in (4) and $\forall x_0 \in \mathcal{X}$, the system (3) has a unique solution.

Proof: We can obtain that

$$\begin{aligned} \frac{\partial \varphi_i(\mathbf{x})}{\partial x_i} &= (1 - 2x_i) \left(\frac{\alpha_i}{x_i + \tau} - h(x_i) \right) \\ &- (x_i - x_i^2) \left(\frac{\alpha_i}{(x_i + \tau)^2} + \frac{dh(x)}{dx} \Big|_{x=x_i} \right) \\ &- \sum_{j=1, j \neq i}^M x_j \left(\frac{\alpha_j}{x_j + \tau} - h(x_j) \right), \end{aligned} \quad (6)$$

$$\begin{aligned} \frac{\partial \varphi_i(\mathbf{x})}{\partial x_j} &= -x_i \left(\frac{\alpha_j}{x_j + \tau} - h(x_j) \right) \\ &+ x_i x_j \left(\frac{\alpha_j}{(x_j + \tau)^2} + \frac{dh(x)}{dx} \Big|_{x=x_j} \right). \end{aligned} \quad (7)$$

(6) and (7) indicate that $\frac{\partial \varphi_i(\mathbf{x})}{\partial x_i}$ and $\frac{\partial \varphi_i(\mathbf{x})}{\partial x_j}$ exist and are continuous in \mathcal{X} . Hence, $\varphi(\mathbf{x})$ is Lipschitz continuous in \mathcal{X} . By the Picard-Lindelöf theorem [12, Theorem 6.1], we obtain Theorem 1. ■

B. Existence of Equilibria

Mathematically, an *equilibrium* (a.k.a., *rest point* or *critical point*) under a solution mapping ξ is a state vector $\mathbf{x} \in \mathcal{X}$ such that $\xi(t, \mathbf{x}) = \mathbf{x}$ for all $t \in \mathbb{R}$ [12, Definition 6.4]. In our model, it means that if the vector of population fractions is at an equilibrium, this population distribution will remain the same, which implies that there are no “job-hoppings” in the blockchain networks. In addition, according to [12, Proposition 6.3], if the vector of population fractions finally converges over time, it will converge to an equilibrium.

Now, we are ready to give all the possible equilibria. Based on whether there are any processor, we can divide the considered blockchain networks into two specific sets, i.e., the *working blockchain set* $\mathcal{W} = \{i : x_i > 0\}$ and the *resting blockchain set* $\bar{\mathcal{W}} = \{i : x_i = 0\}$. A state is an equilibrium if and only if $\varphi(\mathbf{x})$ vanishes at this state. For a given $\mathcal{W} = \{i_1, \dots, i_{|\mathcal{W}|}\}$, it means that

$$u_{i_1}(\mathbf{x}) = \dots = u_{i_{|\mathcal{W}|}}(\mathbf{x}). \quad (8)$$

Theoretically, there are totally $2^M - 1$ possible \mathcal{W} . However, for some values of \mathcal{W} , there may not be any equilibria. Now we give the conditions that a given \mathcal{W} has at least one equilibrium.

Consider a field $\mathcal{K} = \{(a, b) : a, b > 0, \frac{a}{1+\tau} - h(1) \leq b \leq \frac{a}{\tau}\} \subset \mathbb{R}^2$, and a function $K(\cdot) : \mathcal{K} \rightarrow [0, 1]$ such that $K(\hat{a}, \hat{b})$ is the solution for the equation

$$\frac{\hat{a}}{x + \tau} - h(x) = \hat{b}. \quad (9)$$

Notice that when $K(\hat{a}, \hat{b})$ is continuous and monotonically increasing with \hat{a} and decreasing with \hat{b} . Then, we give the following theorem.

Theorem 2. For a given set of working blockchains $\mathcal{W} = \{i_1, \dots, i_{|\mathcal{W}|}\}$, if $\frac{\alpha_{i_1}}{1+\tau} - h(1) < \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}$ and $\sum_{j=1}^{|\mathcal{W}|-1} K\left(\alpha_{i_j}, \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}\right) < 1$, a unique equilibrium exists. Otherwise, there is no equilibrium.

Proof: First, we show if $\frac{\alpha_{i_1}}{1+\tau} - h(1) < \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}$ and $\sum_{j=1}^{|\mathcal{W}|-1} K\left(\alpha_{i_j}, \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}\right) < 1$, a unique equilibrium exists. Since $\alpha_{i_1} \geq \dots \geq \alpha_{i_{|\mathcal{W}|}}$, if $\frac{\alpha_{i_1}}{1+\tau} - h(1) < \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}$, we must have $(\alpha_{i_j}, \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}) \in \mathcal{K}$, $j = 1, \dots, |\mathcal{W}| - 1$, and

$$0 \leq K\left(\alpha_{i_j}, \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}\right) < 1, \quad j = 1, \dots, |\mathcal{W}| - 1. \quad (10)$$

Notice that $K\left(\alpha_{i_{|\mathcal{W}|}}, \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}\right) = 0$. In addition, since for $j = 2, \dots, |\mathcal{W}|$,

$$\frac{\alpha_{i_j}}{1+\tau} - h(1) \leq \frac{\alpha_{i_1}}{1+\tau} - h(1) < \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau} \leq \frac{\alpha_{i_j}}{\tau}, \quad (11)$$

we have

$$0 < K\left(\alpha_{i_j}, \frac{\alpha_{i_1}}{1+\tau} - h(1)\right) \leq 1, \quad j = 2, \dots, |\mathcal{W}|. \quad (12)$$

Obviously the following holds

$$K\left(\alpha_{i_1}, \frac{\alpha_{i_1}}{1+\tau} - h(1)\right) = 1. \quad (13)$$

Thus,

$$\sum_{j=1}^{|\mathcal{W}|} K\left(\alpha_{i_j}, \frac{\alpha_{i_1}}{1+\tau} - h(1)\right) > 1. \quad (14)$$

Since $\sum_{j=1}^{|\mathcal{W}|} K\left(\alpha_{i_j}, \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}\right) < 1$ and $K(\hat{a}, \hat{b})$ is continuous and monotonically decreasing with \hat{b} , there must exist a unique $\bar{b} \in \left(\frac{\alpha_{i_1}}{1+\tau} - h(1), \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}\right)$ such that

$$\sum_{j=1}^{|\mathcal{W}|} K\left(\alpha_{i_j}, \bar{b}\right) = 1, \quad (15)$$

and the population fraction of the i_j th blockchain network is indeed $K(\alpha_{i_j}, \bar{b})$.

When $\frac{\alpha_{i_1}}{1+\tau} - h(1) \geq \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}$, $\forall x_{i_1}, x_{i_{|\mathcal{W}|}} \in (0, 1]$, we can easily obtain that

$$\begin{aligned} &\frac{\alpha_{i_{|\mathcal{W}|}}}{x_{i_{|\mathcal{W}|}} + \tau} - h(x_{i_{|\mathcal{W}|}}) < \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau} \\ &\leq \frac{\alpha_{i_1}}{1+\tau} - h(1) \leq \frac{\alpha_{i_1}}{x_{i_1} + \tau} - h(x_{i_1}). \end{aligned} \quad (16)$$

Hence, there are no equilibria in this case.

Finally, it is obvious that when $\frac{\alpha_{i_1}}{1+\tau} - h(1) < \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}$ and $\sum_{j=1}^{|\mathcal{W}|-1} K\left(\alpha_{i_j}, \frac{\alpha_{i_{|\mathcal{W}|}}}{\tau}\right) \geq 1$, $u_{i_1}(\mathbf{x}) = \dots = u_{i_{|\mathcal{W}|}}(\mathbf{x}) > 0$ will lead to $\sum_{j=1}^{|\mathcal{W}|} x_{i_j} > 1$. Hence, there are no equilibria in this case. ■

C. Asymptotic Stability of the Equilibria

Among all the equilibria, we are especially interested in those which are “robust”. That means, all sufficiently small perturbations of the equilibrium induce a backward movement. The mathematical definition of asymptotic stability in a dynamical system can be found in [12, Definition 6.5].

We assume that \mathbf{x}^* is an equilibrium and let $J_{\mathbf{x}^*}^\varphi$ be the $M \times M$ Jacobian matrix of $\varphi(\cdot)$ at the state \mathbf{x}^* . Then, $J_{\mathbf{x}^*}^\varphi$ can be expressed as

$$J_{\mathbf{x}^*}^\varphi = \begin{bmatrix} \frac{\partial \varphi_1(\mathbf{x})}{\partial x_1} & \frac{\partial \varphi_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial \varphi_1(\mathbf{x})}{\partial x_M} \\ \frac{\partial \varphi_2(\mathbf{x})}{\partial x_1} & \frac{\partial \varphi_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial \varphi_2(\mathbf{x})}{\partial x_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \varphi_M(\mathbf{x})}{\partial x_1} & \frac{\partial \varphi_M(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial \varphi_M(\mathbf{x})}{\partial x_M} \end{bmatrix}_{\mathbf{x}=\mathbf{x}^*}, \quad (17)$$

where $\frac{\partial \varphi_i(\mathbf{x})}{\partial x_i}$ and $\frac{\partial \varphi_i(\mathbf{x})}{\partial x_j}$, $j \neq i$ are given in (6) and (7), respectively. According to [13, Theorem 8.4.3], \mathbf{x}^* is asymptotically stable if the real part of every eigenvalue of $J_{\mathbf{x}^*}^\varphi$ is negative, and it is unstable if any eigenvalue of $J_{\mathbf{x}^*}^\varphi$ has a positive real part. Based on our observations of $J_{\mathbf{x}^*}^\varphi$, we can obtain the following lemma.

Lemma 1. *For a given set of working blockchains $\mathcal{W} = \{i_1, \dots, i_{|\mathcal{W}|}\}$ and its corresponding equilibrium $\mathbf{x}_{\mathcal{W}}^*$, $\forall k \in \bar{\mathcal{W}}$, $\lambda_k = \frac{\alpha_k}{\tau} - \frac{\alpha_{i_{|\mathcal{W}|}}}{x_{i_{|\mathcal{W}|}} + \tau} + h(x_{i_{|\mathcal{W}|}})$ is one eigenvalue of $J_{\mathbf{x}^*}^\varphi$.*

Proof: From (6) and (7), we can obtain that $\forall k \in \bar{\mathcal{W}}$,

$$\begin{aligned} \left. \frac{\partial \varphi_k(\mathbf{x})}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}^*} &= \frac{\alpha_k}{\tau} - \sum_{j=1, j \neq k}^M x_j \left(\frac{\alpha_j}{x_j + \tau} - h(x_j) \right) \\ &= \frac{\alpha_k}{\tau} - \left(\frac{\alpha_{i_{|\mathcal{W}|}}}{x_{i_{|\mathcal{W}|}} + \tau} - h(x_{i_{|\mathcal{W}|}}) \right) = \lambda_k. \end{aligned} \quad (18)$$

In addition, when $x_k = 0$, $\forall \tilde{k} \neq k$, we have

$$\frac{\partial \varphi_k(\mathbf{x})}{\partial x_{\tilde{k}}} = 0. \quad (19)$$

We can observe that the matrix $J_{\mathbf{x}_{\mathcal{W}}^*}^\varphi - \lambda_k I_{M \times M}$ has its elements of k th row all equal to 0, where $I_{M \times M}$ is the $M \times M$ identity matrix. Thus, $\det(J_{\mathbf{x}_{\mathcal{W}}^*}^\varphi - \lambda_k I_{M \times M}) = 0$, where $\det(\cdot)$ is the determinant. Hence, λ_k is an eigenvalue of $J_{\mathbf{x}_{\mathcal{W}}^*}^\varphi$. ■

Based on the Lemma 1, we can further obtain Theorems 3.

Theorem 3. *For a given set of working blockchains $\mathcal{W} = \{i_1, \dots, i_{|\mathcal{W}|}\}$ and its corresponding equilibrium $\mathbf{x}_{\mathcal{W}}^*$, if $\exists k \in \bar{\mathcal{W}}$, and $\lambda_k \geq 0$, this equilibrium is not asymptotically stable.*

Proof: From Lemma 1, we know that λ_k is an eigenvalue of $J_{\mathbf{x}_{\mathcal{W}}^*}^\varphi$. According to [13, Theorem 8.4.3], $J_{\mathbf{x}_{\mathcal{W}}^*}^\varphi$ has at least one non-negative eigenvalue, so $\mathbf{x}_{\mathcal{W}}^*$ is not asymptotically stable. ■

Then, we can obtain the following corollaries.

Corollary 1. *A set of working blockchains $\mathcal{W} = \{i_1, \dots, i_{|\mathcal{W}|}\}$ with $|\mathcal{W}| = w$ has a corresponding equilibrium. If $\mathcal{W} \neq [1, w] \cap \mathbb{N}$, this equilibrium is not asymptotically stable.*

Proof: Since $\mathcal{W} \neq [1, w] \cap \mathbb{N}$, there must exist a positive integer l such that $l \leq w$ and $l \in \bar{\mathcal{W}}$. Clearly, $\lambda_l \geq 0$. According to Theorem 3, we have this corollary. ■

Corollary 2. *If the set of working blockchains $[1, w] \cap \mathbb{N}$ has a corresponding equilibrium, for any positive integer r such that $r < w$, the set of working blockchains $[1, r] \cap \mathbb{N}$ must have a corresponding equilibrium, which is not asymptotically stable.*

Proof: It is easy to obtain

$$\frac{\alpha_1}{1 + \tau} - h(1) < \frac{\alpha_w}{\tau} \leq \frac{\alpha_r}{\tau}, \quad (20)$$

and

$$\sum_{j=1}^{r-1} K\left(\alpha_j, \frac{\alpha_r}{\tau}\right) \leq \sum_{j=1}^{r-1} K\left(\alpha_j, \frac{\alpha_w}{\tau}\right) \leq \sum_{j=1}^{w-1} K\left(\alpha_j, \frac{\alpha_w}{\tau}\right) < 1. \quad (21)$$

Hence, according to Theorem 2, the set of working blockchains $[1, r] \cap \mathbb{N}$ must have an equilibrium.

We denote this equilibrium by $\mathbf{x}_{[1, r] \cap \mathbb{N}}^*$, with Jacobian matrix $J_{\mathbf{x}_{[1, r] \cap \mathbb{N}}^*}^\varphi$. Let $\sum_{j=1}^r K(\alpha_j, \bar{b}_r) = 1$ and $\sum_{j=1}^w K(\alpha_j, \bar{b}_w) = 1$. Clearly, $\bar{b}_r < \bar{b}_w$. Since $w \notin [1, r] \cap \mathbb{N}$, according to Lemma 1, $\frac{\alpha_w}{\tau} - \bar{b}_r$ is an eigenvalue of $J_{\mathbf{x}_{[1, r] \cap \mathbb{N}}^*}^\varphi$. Since $\frac{\alpha_w}{\tau} > \bar{b}_w > \bar{b}_r$, according to Theorem 3, $\mathbf{x}_{[1, r] \cap \mathbb{N}}^*$ is not asymptotically stable. ■

Theorem 4. *Let $w^* = \max\{w : \frac{\alpha_1}{1 + \tau} - h(1) < \frac{\alpha_w}{\tau}, \sum_{j=1}^{w-1} K(\alpha_j, \frac{\alpha_w}{\tau}) < 1\}$. Then, only the corresponding equilibrium of the set of working blockchains $[1, w^*] \cap \mathbb{N}$ has probability to be asymptotically stable.*

Proof: According to Corollaries 1 and 2, all the other equilibria are not asymptotically stable. Then, we must have Theorem 4. ■

V. PERFORMANCE EVALUATION

In this section, we provide the numerical analysis of the population dynamics of ELASTICO. We assume four blockchain networks, for which $\alpha_1 = 0.7$, $\alpha_2 = 0.5$, $\alpha_3 = 0.3$, and $\alpha_4 = 0.1$. In addition, we set $\tau = 0.01$ and $h(x) = \ln(1 + x)$. In this condition, from our analysis in Section IV-B, we can easily obtain $\mathbf{x}_{e1} = [0.4225, 0.3148, 0.1975, 0.0652]^\top$ and $\mathbf{x}_{e2} = [0.4499, 0.3369, 0.2132, 0]^\top$ are two equilibria. By calculating the eigenvalues of Jacobian matrix of $\varphi(\cdot)$ at the state \mathbf{x}_{e1} , we can find that \mathbf{x}_{e1} is asymptotically stable. Meanwhile, according to Corollary 2, we can see that \mathbf{x}_{e2} is not asymptotically stable.

We show the population dynamics from an initial point $\mathbf{x}_0 = [0.4498, 0.3369, 0.2132, 0.001]^\top$, which is a very small deviation from \mathbf{x}_{e2} . From Figure 2, we can observe that, instead of moving backward to its neighboring equilibrium \mathbf{x}_{e2} , the state moves away from this area and settles down over time toward the new equilibrium \mathbf{x}_{e1} . This observation coincides with our analysis given in the previous paragraph. Then, in Figure 3, we show the dynamics of payoff for each blockchain network. We can observe that at the beginning, since there are only a few processors in the fourth blockchain, the payoff

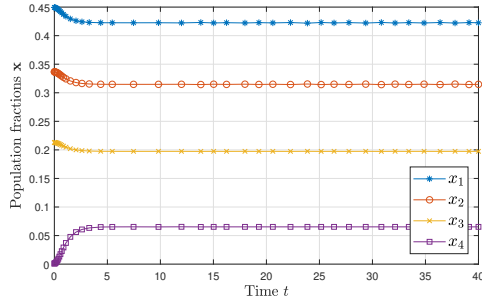


Fig. 2. The population dynamics for four blockchain networks, from the initial point $\mathbf{x}_0 = [0.4498, 0.3369, 0.2132, 0.001]^T$.

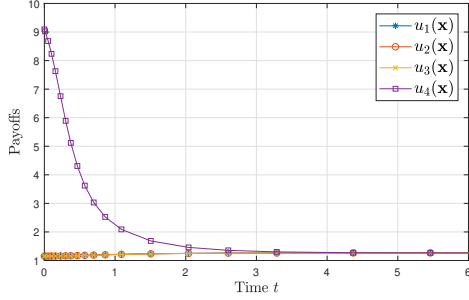


Fig. 3. The payoff dynamics for four blockchain networks, from the initial point $\mathbf{x}_0 = [0.4498, 0.3369, 0.2132, 0.001]^T$.

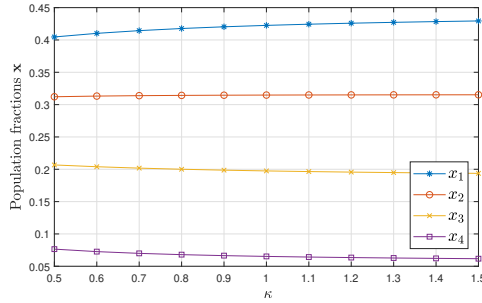


Fig. 4. The asymptotically stable equilibrium for $\alpha_1 = 0.7\kappa$, $\alpha_2 = 0.5\kappa$, $\alpha_3 = 0.3\kappa$, and $\alpha_4 = 0.1\kappa$, where κ changes from 0.5 to 1.5.

per second of one processor is much higher than other three blockchains. However, as more and more processors move to the fourth blockchain, the payoff for the fourth blockchain decreases while those for the other three blockchains increase. Finally, they reach an equilibrium and all the blockchains have the same expected payoff.

Finally, we show how the asymptotically stable equilibrium changes with the prices. We set $\alpha_1 = 0.7\kappa$, $\alpha_2 = 0.5\kappa$, $\alpha_3 = 0.3\kappa$, and $\alpha_4 = 0.1\kappa$, where κ varies from 0.5 to 1.5. Figure 4 shows that, as the prices increase, the population fractions of blockchain networks with higher prices increase while those of blockchain networks with lower prices decrease.

VI. CONCLUSION

In this paper, we have investigated the process of eco-system formation for a large population of consensus nodes to join one consensus process from multiple permissionless, sharded blockchains. In particular, we have considered the scenario of

multiple blockchains adopting the ELASTICO protocol, which combines the proof-of-work puzzle and the Byzantine fault-tolerant agreement protocol to achieve both open access and low transaction-processing latency. We have considered a general scenario where different blockchains may provide different transaction fees and have different transaction-generating rates. We have studied the blockchain-selection behaviors of the independent, bounded-rational consensus nodes with identical computational power. The behaviors of blockchain selection by the consensus nodes have been formulated as an evolutionary game based on replicator dynamics. We have provided a series of analytical and numerical results, which reveal the consensus-formation mechanism in a permissionless network of blockchains for multiple service provision.

VII. ACKNOWLEDGEMENT

This work was supported in part by WASP/NTU M4082187 (4080), in part by Singapore MOE Tier 1 under Grant 2017-T1-002-007 RG122/17, in part by MOE Tier 2 under Grant MOE2014-T2-2-015 ARC4/15, in part by Singapore NRF2015-NRF-ISF001-2277, in part by Singapore EMA Energy Resilience under Grant NRF2017EWT-EP003-041 and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government under Grant 2017R1A2B2003953.

REFERENCES

- [1] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2084–2123, third quarter 2016.
- [2] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, pp. 1–1, 2019.
- [3] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, May 2016.
- [4] K. Kotobi and S. G. Bilen, "Secure blockchains for dynamic spectrum access: A decentralized database in moving cognitive radio networks enhances security and user access," *IEEE Vehicular Technology Magazine*, vol. 13, no. 1, pp. 32–39, Mar. 2018.
- [5] N. Herbaut and N. Negru, "A model for collaborative blockchain-based video delivery relying on advanced network services chains," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 70–76, 2017.
- [6] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," Ethereum Foundation, Tech. Rep., 2014. [Online]. Available: <https://github.com/ethereum/wiki/wiki/White-Paper>
- [7] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA: ACM, 2016, pp. 17–30.
- [8] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *Open Problems in Network Security: IFIP WG 11.4 International Workshop*, Zurich, Switzerland, Oct. 2015, pp. 112–125.
- [9] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, Nov. 2002.
- [10] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains," in *Financial Cryptography and Data Security: International Workshops on BITCOIN, VOTING and WAHC*, Christ Church, Barbados, Feb. 2016, pp. 106–125.
- [11] D. J. Newman, "The double dixie cup problem," *The American Mathematical Monthly*, vol. 67, no. 1, pp. 58–61, 1960.
- [12] J. W. Weibull, *Evolutionary game theory*. MIT press, 1997.
- [13] N. Lebovitz, *Ordinary Differential Equations*, ser. Mathematics Series. Brooks/Cole, 1999.