Inferred based Distributed Gaming, Nash Equilibrium

Ryan J. Kung
ryankung@ieee.org
Member, IEEE Blockchain Community

April 3, 2018

1 Abstract

Since the introduction of Ethereum on 2014 [1], It brought many DApps and DGames, such as CryptoKitty, Etheremon and Decentraland. Most of them are running into issues addressed to the efficiency of BlockChain implementation, It's too expensive to store data and status in BlockChain like what traditional games does, And it's also can't be worked as a real-time or parity-real-time game during to the delay of transactions. [**?**]

In this paper, we will studied and discussed how should be a distributed game work on a blockchain based distributed system. First, we discussed distributed nash equilibrium problem, and try to solve the problem with Infered System, finally we will discuss about Zero-knowledge proof as futrue work.

2 Distributed Nash-Eq

Like DApps, DGaming is a series of Gaming Behaviors and Strategis which is Decentralized and Distributed. On Lamport's defination on 1978 [3], A Gaming is Distributed if the message transaction delay is not negligible compared to the time between event in classic gaming behavior. The issues in traditional gaming theory may easily lead us run into trouble if it's distributed, such as nash equilibrium simulation or prisoner's dilemma problem. On classic Nash Equilibrium, we modeling it with some behaviors table, and proof that for each strategy on game (S, f), and got A strategy profile which is Nash Equilibrium. But on distributed case, the issue is that the event of strategy is not effect immediately, nonnegligible delay of transaction on system may cause N-Eq point out of expectation.

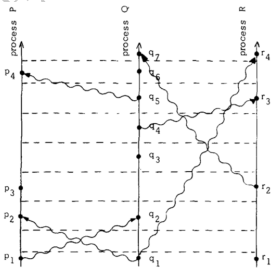

Figure 1: Lamport Timestamp

In a distributed nash equilibrium case, for example an distributed gaming system build on a simple-paxos based distributed system, actors may try to figure out best response by mining unconfirmed messages as more as they can, which entangled the gaming into a series complicated case, and break perfect information gaming in to imperfect.Actually, Nash equilibrium seeking under communication networks is as tracing increasing attension resently, Some seeking work is about paritially coupled payoff method [5], or Communication Topologies [7]. In this paper, we considered distributed Nash equilibrium with a set of player $\mathbb{N} = 1, 2, ..., N$, under a distributed network which has nonnegligible delay δ, and Denote $f_i(x)$ as the payoff function of player i, where $X = [x_1, x_2, ..., x_n]^T$, $x_i \in S$ is action of player i. And we define a measurement function $m = f(\delta, i)$ which is means how the information is prefect for player i.

The distributed equilibrium seeking strategy(DESS) was given by [6]:

$$\dot{x} = k_i \frac{\partial f_i}{\partial x_i}(y_i), i \in \mathbb{N} \tag{1}$$

$(\dot{x}^*, 1_N \oplus x^*)$, where x^* presents nash equilibrium. There are some descendent work based on it, but most of them are analytically studied based on certain condition, ignores nonnegligible communication or transaction cost, which is non-ignorable in blockchain and other distributed system. In blockchain case, we can compose measurement function and DESS together, then the strategy can be present as $m(\delta, \dot{x})$. And obviously, there should be existed some relationship between Nash-eq point and δ, such as:

$$m(\delta_a, \dot{x}) > m(\delta_b, \dot{x}) \vdash a > b \tag{2}$$

3 Infered perfect information

Unlike classic distributed system, Since some abstraction of blockchain is based on the separation logic or *πcalculis* [**?**], it can easily be seems as a time step function, or a FRP issue which can be abstract as [2], where blockchain Status can be also be abstract as $STMonad$ like MAKERDao [4]'s work:

$$time : STMonad_{time} \tag{3}$$
$$at[time]t = t \tag{4}$$

Unlike other issues we met in pure functional world, there should be only one types of monad in blockchain, the STMonad, $STMonad$ is a Monad type for status, thus except status changes, all other functions should be pure which means no side-effect, which means we can build an infered system or a $LENS$ [**?**] above it. A Lens should have two property, *over*, and *view*.

```
data Lens a b = Lens
    { view :: a -> b
    , over :: (b -> b) -> (a -> a)
  }
```

A concrete example based on Solidity is, for each functions, if it's a pure function which means nor cost gas or cause side-effect on blokchain it can be infered based on ABI, it's a *view*, otherwise it's a *over* which means you have to send a transaction to modified the status of chain.

4 Pure Functional Solidity

blalbal

References

[1] Control.lens.tutorial.

[2] Linear types can change the blockchain!

[3] Vitalik Buterin. A next-generation smart contract and decentralized application platform, 2014.

[4] Conal Elliott and Paul Hudak. Functional reactive animation. In *International Conference on Functional Programming*, 1997.

[5] Leslie Lamport. Time, clocks and the ordering of events in a distributed system. pages 558–565, July 1978.

[6] Mikael Brockman Nikolai Mushegian, Daniel Brockman. Reference implementation of the decentralized dai stablecoin issuance system, 2 2018.

[7] L. Wang, S. Liang, and Y. Hong. Discrete-time algorithm for distributed nash equilibrium seeking of a class of aggregative games. In *2017 36th Chinese Control Conference (CCC)*, pages 11325–11330, July 2017.

[8] M. Ye and G. Hu. Distributed nash equilibrium seeking by a consensus based approach. *IEEE Transactions on Automatic Control*, 62(9):4811–4818, Sept 2017.

[9] M. Ye and G. Hu. Distributed nash equilibrium seeking in multiagent games under switching communication topologies. *IEEE Transactions on Cybernetics*, PP(99):1–10, 2017.