Inferred based Distributed Gaming and Nash Equilibrium

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

April 13, 2018

1 Abstract

Since the introduction of Ethereum on 2014 [3], It brought many DApps and DGames, such as CryptoKitty, Etheremon and Decentraland. Most of them were running into issues addressed to the congestion of transaction which is limitation of how Ethereum implemented. Storing on-chain data is too expensive for games needs high interaction, and may cause may problem, one of them is that gaming on distributed system may cause harder to find out a fear nash equilibrium point for numerical designing, and the nonnegligible delay of tranasction caused may limited the interactive rate of game processing.

In this paper, we studied and discussed how should a distributed game works on blockchain based distributed system. First, we discussed distributed nash equilibrium problem on certain condition, then we introduced an infered based system based on an abstraction which formalize blockchain with STMonad and Lens application, further, we discussed how to composed infered system with zero-knowledge proof for game refining.

2 Distributed Nash-Eq

Like traditional Gaming on game theory, we defined DGaming as a series of Gaming Behaviors and Strategis which is Distributed. With Lamport's defination on 1978 [8], 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 seeking out the nash-eq point by analyze it with action matrix, 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.

For example, we discuss a classic game as:

	L	R
U	(1,3)	(-3,0)
M	(-2,0)	(1,3)
D	(0,1)	(0,1)

With Repeated advantage solution, we can simply figure out that the nash-eq strategy is (U, L), but this game is based on a hypothesis that player i shares same knowledge about payoff and strategy space. If players are sensitive on the uncertainty of s_{-1}, they may not choose the rational strategy, like if about %1 players choose R, then D is better than U.

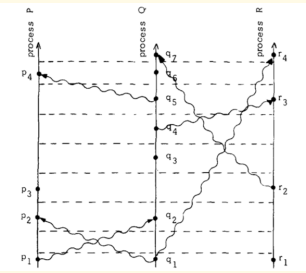

Figure 1: Lamport Timestamp

In distributed game, for example an distributed game based on a paxos based system. Addressed to the nonnegligible delay, player i may upfront know i' and action hysteretic, which may cause the nash-equilibrium become highly dependents on the distributed network. Figure 1 is showing that how events can be trigger and measured on a distributed system. Actually, on a Dgame all players may tried to mining more informations and action as later as they can, the situation may looks like Prisoners Dilemma:

	W	A
W	(1,1)	(-1, 2)
A	(2,-1)	(0,0)

On this game, players can choose from strategy space $Wait, Action$, and as what prisoner's dilemma presented to us, all players may choose W. Which entangled the gaming into a series complicated case, and break perfect information gaming in to imperfect.

Nash equilibrium seeking under communication networks is as tracing increasing attension resently, The distributed equilibrium seeking strategy(DESS) was given by [12]:

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

$\mathrm{T}(\dot{x}^*, 1_N \oplus x^*)$, where x^* presents nash equilibrium. There exist Some seeking work on it with paritially coupled payoff method [11], or Communication Topologies [13] building on certain conditions and developed on the discrete-time stochastic algorithm [**?**] which the players are supposed to be capable of communicating with each other via an undirected and connected communication graph [12].

In this paper, we considered a game (S, f) 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 mix strategy function $\sigma = f(\delta, s)$ which is means how the information is prefect for s_i. And formulated the mix strategy function based on nonnegligible transaction as $= sigma(\delta, \dot{x})$, and we have.

3 Infered Game

An infered Game is that for a game (S, f), all players i shares same knowledge about S and $u_i(s)$. Unlike classic distributed system, Since some formalization of blockchain is based on the separation logic or $\pi calculis$ [2], it can easily be modeled as a time step function, or a FRP issue which can be abstract as [6], where blockchain Status can be also be abstract as $STMonad$ like MAKERDao [10]'s work:

$$time : STMonad_{time} \tag{2}$$
$$at[time]t = t \tag{3}$$

In blockchain, there is only one types of monad, 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 $LENS$ [1] 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. From (2) we know that the distributed nash-equilibrim point is based on the transaction param δ, thus we can assume that *Lens.view*, and *Lens.over* is some paritial function based on δ. We know that if *view* is a pure function so that $\lim \delta$ is 0, which means if a client is well connected to the network, the δ is negligible for that, since for *over* case, you can only shift δ by provide a high *tranasction price* (gas price for ethereum).

So since classic distributed gaming may transfer the gaming from perfect information to imperfect. But if it's blockchain based, since *over* is working as eventual consistency [4] but the status of blockchain can be seems as strong consistency [**?**]. On *view* phase of gaming, when player i make decision on $STMonadtime_0$, the information of $Monat$ is perfect. For phase *over*, players may choose to pay high price as strategy to get more payoff, since it's based on consistency information of blockchain, we can also say that the *over* phase is also perfect information gaming. So for a gaming we wished to have a fixed nash equilibrium point, it should be designed as all data in public and no private status.

4 Zero Knowledge Gaming

Such as we can formalize blockchain as a Monad (A monoid in the category of endofunctors), We can also describe abstractly the cryptology part of bitcoin as a cyclic subgroup of elliptic curve group over a primer finite field, which is also the the key of zk-SNARKs. There is an concrete implementation in our game for providing zero-knowledge proofing based on the infered system and hash algorithms, It's works like other blockchain implementation does [5]. But what further more is, we tried developed algorithms based on Abstract Algebra Type and Isomorphism or Bijection mapping.

The Abstract Algebra Type made an allowance for playing high abstract part of calculators in a more intuitionistic way. Such as when we try to implement ethereum's pubkey/privatekey algorithm, we can just simply write as [7]:

```
def pubkey(priv: CF) -> ECG:
    return ECG(G @ priv)
```

Where CF and ECG is Finite Field Cyclic Group and Elliptic Curve Group of Secp256k1. By the property of cyclic subgroup, we can easily improve the zero knowledge gaming as some endomorphism feature.

$$A + B = X \tag{4}$$
$$A + C = X \tag{5}$$
$$C \neq X \tag{6}$$

5 More Concrete

In the Go!hamster, the game we designed based on this work. We abstract the gaming activity as mining, and designed a series of numerical system, which can maintain the value of user actor by a fixed and well-designed nash equilibrium pointed.

5.1 Decentralized Marking

The ERC20 standard was extended with a built-in transparen exchange, which makes all gaming actor can be limited inside the blockchain or contract. And based on ERC721, we designed a reward system to keep player can get payoff if they act around the nash-eq point. For Exchangeable ERC20 and so on the Exchangeable ERC721, all matching stuffs is based on infered system, which means players can check the full depth of ticker for determining the best price they should pay for tokens.

Functions of smart contract are separated into Functor and Monad. Functors are working based on pre-infered system, and Monads are based on infered system, which is actually presented a sequence of status.

5.2 Infered Token Balance

On out implementation of Gaming Token, we designed an infered based auto expended rate. The expended rate is based on δ and *time* of blockchain, which is bounded on the nash equilibrim of what the token price should be during the game processing.

5.3 Gaming Versus Mining

The most obviously case of nash-eq case on blockchain world is bitcoin mining [9], we can easily build an game matrix model on bitcoin mining behavior, the relationship is between market force, difficult rate, and hash rate. And in game Go!hamster, actors can pay game token for wining reward, more token they paid, higher rewarding rate they got, which is working as a simulation of how bitcoin mining works.

6 Conclusion

Addressed to the strong consistency property of blockchain, A game can have a certain nash equilibrium point, if the side-effects and pure-functions are in separation. Though there is alot of limitation on Ethereum and solidity itself, a real-time game, like RPG or Action game can be still build on a monad based and zero-knowledge based infered system.

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] Roger Wattenhofer Christian Decker, Jochen Seidel. Bitcoin meets strong consistency.

[5] Christina Garman Eli Ben-Sasson, Alessandro Chiesa. Zerocash: Decentralized anonymous payments from bitcoin.

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

[7] Ryan J. Kung. Klefki is a playground for researching elliptic curve group based cryptocoins.

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

[9] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system.

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

[11] 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.

[12] 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.

[13] 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.