星云链技术白皮书

Nebulas Inc.

August 7, 2017

Abstract

白皮书的摘要，让大家看完摘要后，就对星云链要做的事情一目了然。
TODO 是蓝色
注释是红色
引用的例子，如下 GNU pthread [**pth**], PARSEC [**bienia2009parsec**]

Contents

1 简介

1.1 区块链技术简介

简单介绍一下区块链技术的概要，和最流行的开源区块链技术特点。

1.2 商业和技术挑战

当前公有链遇到的技术挑战：智能合约无法被简单搜索；用户以及智能合约的价值贡献无法被评估和计算；PoW 共识速度慢，消耗大量电能，PoS 共识使得拥有大量代币资产的用户获得记账资格，形成马太效应，带来另外的不公平忧虑；目前的激励机制仅仅奖励矿工，有一定的局限性，不能鼓励优秀的智能合约开发者；智能合约一经部署，无法升级，不适应普通的企业应用需求；区块链核心协议如果需要升级，容易引起硬分叉，对区块链社区带来伤害；

1.3 星云链解决的问题

介绍星云链的贡献：NR 排序算法，核心协议的可升级、PoR 共识（出块速度和安全性）、DIP 开发者激励、智能合约升级设计、

1.4 星云链设计基本原则

阐述星云链设计的基本原则：完全兼容以太坊、为区块链设计价值尺度、更快更合理的共识设计、对区块链生态的有效激励、图灵完备可升级的智能合约、可自我演化的区块链、更好用的区块链基础服务。

2 星云链代币 NAS

robin

3 账号和地址

Robin

4 区块和交易

Wenbo

5 Nebulas Rank 算法

5.1 Introduction

Ranking nodes in complex network has been a fundamental concern in various applications. One canonical example is PageRank[**Brin2010**][**page1999pagerank**], which is the core algorithm for Google and other search engines[**langville2011google**]. Besides, by ranking algorithm, people also want to find out the most influential spreaders in epidemic and information network[**doerr2012rumors**][**Kitsak2010**], the most acknowledged scientist by the citation network or co-author network[**walker2007ranking**][**chen2007finding**][**Radicchi2009**], the most important cities in the transportation network[**guimera2005worldwide**], the most important vertices in metabolic network[**ivan2010web**], the top VC firms by co-investment[**Bhat2012**] etc. And when it comes to designing **Nebulas Rank** algorithm for blockchain world, decades of researches have enlightened us with many measurements like degree centrality[**freeman1979set**], eigenvector centrality[**bonacich1972factoring**], Katz centrality[**katz1953new**], PageRank[**Brin2010**], HITS[**kleinberg1999authoritative**], closeness centrality[**sabidussi1966centrality**], betweenness centrality[**freeman1977set**][**freeman1978centrality**][**freeman1991centrality**] etc. Before building our algorithm on the top of them, however, we still need to answer two questions:

1. What properties are embedded in the network?
2. What value should the rank represent?

For the first question, **Nebulas Rank** uses the transaction graph of blockchain system, which is generated by the transaction history during the past period. We compare blockchain transaction graph with others in three aspects. First, as node representing blockchain system accounts and edges representing money transferring between two accounts, basically, the graph is a weighted directed graph, bearing large difference from social network[**Ugander2011**] and similarity with webpage network[] in terms of topology. An asymmetric, i.e. directed, edge represents the imbalanced ability of giving or collecting money between two nodes. And the edge weight is of absolute amount, rather than probabilistic, which also enables disparity of link quality among edges. Directly applying standard algorithms could discard these important information. Second, since the graph reveals trajectory of money flow, it could be presumed that Exchanges are highly ranked, whereas such accounts are willing to swap money with any client. Thus anyone can acquire unlimited links from those existing important nodes without much cost. Along with the anonymity of typical blockchain systems, sybil attackers could make large amount of transferring with a big Exchange account, in order to improve his influence such as PageRank. (It is still hard to receive money from a large number of non-sybil nodes, though.) This is an essentially difference from previous researches' assumptions[]. For example, [] assumes that in an online social network with subscription[], it costs no effort to follow an opinion leader, while attracting a verified opinion leader's attention is not an trivial thing. Third, being different from Bitcoin[], the newly invented blockchain systems such as Ethereum[] introduces "contract" as a new type of account. After a normal account invoking some method of a contract, a sequence of consequent callings will be raised forming part of a call graph. Unlike Bitcoin[] transaction graph, which only contains money transferring, Ethereum call graph/network also represents dynamic programming calling. We believe such network embeds more information and should be useful to measure a DApp or smart contract's value.

For the second question, **Nebulas Rank** aims to measure the value of users, and smart contracts [] in blockchains. For normal accounts, we define value by two aspects: **Liquidity**, which stresses the ability to control digital assets flow of high quality; **Propagation**, which focus more on the spreading influence. For smart contracts, we also consider **Interoperability** as a measurement. There are three-fold purposes in **Nebulas Rank**: 1) to be a good metric for blockchain accounts and smart contract search engine; 2) to provide a trustful criteria in PoS[**poelstra2015stake**] consensus protocol, where only high ranked nodes should be eligible to become a validator; 3) DIP[TBA]. In this version of white paper, we does not include smart contracts and DIP, as they are of another independent interest and would be presented later. So next all we will discuss is about normal accounts ranking only.

As revealed in [**Borgatti2005**], most centrality measures can be classified by their type of network flow. From the dimension of diffusion mechanism, traffic flow can spread by different kinds of duplication or transfer. Another spectrum is the trajectory of flow, which can be either paths, trails or walks. Essentially, blockchain transaction graph is the trajectory of money exchange, which falls into the classification of "transfer walk". Imagine an amount of money enters the network. Then the owner node divides the money and transfers it to its neighbors or keep it with different probabilities, More specifically, the money is divisible, imperishable and non-replicable and each step of walk is random due to the limit of local information. So with the context of "transfer walk", we roughly represent the Liquidity by the amount of flows a node "controls" and the Propagation by the amount of flux over a node. Additionally, however, although the graph represents money exchange, it may as well be seen as other flow types which might outputs better ranking. For example, information flows with "parallel replication along paths", since one may also argue that accounts transfer money because of knowing each other and exchanging information.

Following are our solutions to the challenges described above.

First, in order to turn a list of transaction records into a graph, we keep the transferring value as the edge weight and embed temporal information into node's property. We assign each directed edge's weight as the sum of largest K transactions amounts from source account to target account during the past T days. Then we reduce each edges' weight by its target's "coinage" (see §5.3.3) and outgoing amount(see §5.3.6) as well as an encouragement function (see §5.3.4). By this mean, a directed weighted graph can be generated and some undesired activities can be mitigated:

- By setting a time window of T days and treating transactions, old and new, impartially, a node is encouraged to keep active all the time;
- Because at most K transactions between each two nodes are counted, simply transferring over edges repeatedly does not improve links' quality;
- In order to get higher coinage, money needs to stay in place for a while, which slows down sybil attacks such as transferring money along loops or interact frequently with Exchange service node
- Accounts are encouraged to transfer out enough amount of money in order not to get their in-links reduced, which contributes to better money circulation.

We will talk about details of transaction processing method in §5.3.

Second, with graph generated, we measure each node's importance based on Weighted LeaderRank algorithm[**Chen2013**][**Li2014**]. LeaderRank is a simple variant of PageRank, which adds a ground node into the network and connect the ground node with each non-ground node. It substitutes PageRank's damping factor by links throughout ground node, which is said to be more effective in computing, robust against manipulations and noise than PageRank algorithm[**Chen2013**]. The intuition for both LeaderRank and PageRank is random walk and Markov Chain. By PageRank, from each node, the probability of jumping to an arbitrary node is the same (or equal to 1 if there is no out links [**Kim2002**]). Whereas by LeaderRank[**Li2014**][**Chen2013**], different nodes adopt different arbitrary transition probabilities. For example, we could allow a node with more in-links to receive more from the arbitrary transition. This is more plausible in the context of blockchain, since an account with little money transferred in is less trustable. Also, imagine a node receiving much money but hardly spend it. We suppose such account has more "surplus value" and assign it with more arbitrary outgoing probability. We will talk about the details of LeaderRank scheme in §5.4.

LeaderRank is a measurement of the node flux. Intuitively, money flows in and out from one node in a continuous and dynamic pattern, and LeaderRank observes the steady amount of money passing by a specific node. From another perspective, mode flux means more control. The LeaderRanks algorithm matches both our goals of measuring **Liquidity** and **Propagation**. Although some other betweenness based algorithm such as flow betweenness[] and random walk betweenness (aka. current flow betweenness)[] may be more suitable to represent the flow controlling ability i.e. **Liquidity**, such measurements are quite computational intensive. So we don't adopt them for now. Besides, there are some more thinkings during the design of **Nebulas Rank**. For example, the network clustering may also be helpful to lower rank spam nodes[]. But it is also defective to overexploit the clustering effect. All issues will be discussed at §5.6.

~~Our experiment results shows that ......~~

The rest of paper is organized as follows. §5.2 introduces the related works. Then in §5.3, we define the network topology and weight based on blockchain transactions. And in §5.4, LeaderRank with schemes designed for **Nebulas Rank** is introduced. In §5.5, we show our experiment results. And Finally we give all discussions and conclusions in §5.6

5.2 Related Works

Centrality, the core ranking index, is a most studied concept in network science since decades ago[**newman2010networks**]. There are a body of literatures introducing various centralities, including degree centrality[**freeman1979set**], eigenvector centrality[**bonacich1972factoring**], Katz centrality[**katz1953new**], closeness centrality[**sabidussi1966centrality**], betweenness centrality[**freeman1977set**][**freeman1978centrality**][**freeman1991centrality**][**noh2004random**][**newman2005measure**], PageRank[**Brin2010**], HITS[**kleinberg1999authoritative**], SALSA[**Science2001**], etc. It is fundamental to clearly classify these measurements by a unified framework. **Borgatti2005** adopts a network flow based view to classify the centrality measurements by two categorical dimensions: material flowing by parallel duplication, serial duplication and transfer; and trajectory following geodesics optimum, path, trail and walk. **Borgatti2006** propose a unified framework with four dimensions from the perspective of graph theory. **Lu2016** review representative centrality algorithms and classified them into those only based on structural information, those driven by Markov dynamics, those by looking at the effect of removing nodes, those with dynamics-sensitivity and those trying to identify more than one node. With a hierarchical understanding of centrality algorithms, we are able to choose appropriate strategy according to the network scenarios. **Nebulas Rank**'s scenario is the money exchange flow network mentioned in [**Borgatti2005**].

Since Bitcoin[**Nakamoto2008**] system released in 2009, researchers have done some statistical and empirical analysis on Bitcoin's transaction graph[**Ron**][**Haslhofer**][**NielKondor2014**][**Baumann2014**], and some use the transaction graph structure to discuss anonymity in Bitcoin[**Meiklejohn2013**][**Ober2013**][**pham2016anomaly**][**Fleder2015**][**Fer**
After other cryptocurrencies emerged and become popular, transaction graph analysis is conducted with more blockchains[**Chang2017**][**Anderson2016**]. **Nebulas Rank** adopts their transaction graph concept, i.e. Entity Graph in [**Tschorsch2015**], with minor revisions. That is, each account, or set of accounts belonging to the same people, is mapped as a node. And each directed edge represents the intensity of transferring between two accounts. Actually before blockchain system like Bitcoin was invented, scientists have tried to study some financial networks among banks and global trading entities[**propper2008towards**][**Boss2004**][**Serrano2007**][**Bech2008**][**Fagiolo2009**][**Morten2006**][**Boss2004a**][**Krempel2002**][**Serrano2003**]. Comparing with blockchain transaction networks, these early studied finical networks are defined not only by transferring activities, but also by lending-based relationship. Moreover, the scale of these networks is much smaller. To conclude, there is rarely research work proposing custom ranking method for large scale transaction graph, especially blockchain transaction graph.

The most relevant work with **Nebulas Rank** is NEM[**nem**]'s Proof-of-Importance scheme. It adopts NCDawareRank[**Nikolakopoulos2013**], which exploits the clustering effect of network topology, as the ranking algorithm, with clustering algorithm based on SCAN algorithm[**xu2007scan**][**shiokawa2015scan**][**chang2017mathsf**]. And **Fleder2015** uses PageRank[**Brin2010**][**page1999pagerank**] as an assisting metric to discover interesting addresses and analyze their activities. However, both NCDawareRank and PageRank are ranking algorithms for webpage network. As we already mentioned in §5.1, blockchain transaction graph is very different from webpage network. And although community structure does exist in transaction graph and should be helpful to handle with spam nodes, it does not suit the consensus purpose mentioned in §5.1. Because in order to compute "unforgeable" node importance, accounts controlled by a single "real world" entity should be guaranteed to be mapped to the same cluster. However it is key difficulty to connect blockchain world with the "real world", and thus there is no proper objective definition for clustering problem. Therefore current clustering algorithms cannot provide meaningful and trustful result. Moreover, [**Fleder2015**]'s work does not provide an automated framework to identify important nodes. Instead, it still needs manual analyzing with the help of PageRank, which does not match **Nebulas Rank**'s context.

The algorithm we choose is LeaderRank[**Chen2013**][**Li2014**]. It is a simple variant of PageRank[**Brin2010**][**page1999pagerank**]. In PageRank, initially every node gets one unit rank value. Then at each iteration, every node distributes its rank value equally to its directed neighbors. To deal with dangling node problem, there is a damping factor, where every node distribute a specific proportion of its rank value to all nodes equally in the network. **Chen2013** propose a simple yet effective modification on PageRank's damping factor and call it LeaderRank. Then **Li2014** extend LeaderRank to weighted case and further improve its performance. By weighted LeaderRank[**Li2014**], an additional ground node is added and a bidirectional link is added between every node and ground node. Every edge targeting to ground node is of same weight and every edge from ground node is weighed positively proportional to target node's in-degree. LeaderRank is more resistant against manipulation and noisy data than PageRank[**Chen2013**][**Li2014**][**Lu2016**]. In terms of computation, LeaderRank can be seen as PageRank with one more node and set damping factor to be zero. And thus it is easy to implement and very scalable. We will modify LeaderRank a little and discuss more on its weighting scheme at §5.4.

5.3 Transaction Graph

5.3.1 Transactions

The input data for **Nebulas Rank** are all the transaction records, i.e. token transferring, during the past T days, denoted by a set of tuples:

$$T_{xs}^{all} = \{(s, t, \tau, a), \tau = Today - T \ldots Today\}$$

, where s, t and a are the source account, target account and amount of an transfer, respectively.

Further, we filter transactions, providing that self transfer and zero amount transfer are excluded:

$$T_{xs} = \{(s, t, \tau, a) | s \neq t \wedge a > \Phi \wedge (s, t, \tau, a) \in T_{xs}^{all}\}, \Phi = 0 \tag{1}$$

5.3.2 Transactions Aggregation

Based on transactions defined above, we construct the directed weighted transaction graph $G = (V, E, W)$, where node set, edge set and weight on edges are denoted by V, E and W respectively. We also denote that $N = |V|$ and $M = |E|$.

Each vertex $v \in V$ represents one individual account's address. Each edge represents the transferring intensity between two accounts. Consider $e = (s, t) \in E$, this edge is directed, and naturally, the weight of it should be determined by all related transactions, i.e. $(s, t, \tau, a) \in T_{xs}$. To compute edge (s, t)'s weight, we take the sum of top K amounts out of all related transactions:

$$w_e = \sum_{i=1}^{K} a_i, s.t. a_i \in \{a | (s, t, \tau, a) \in T_{xs}\} \wedge a_1 \geq a_2 \ldots \tag{2}$$

By this mean, the link between two nodes is bi-directed and asymmetric, with top K transactions along each direction aggregated to become the weight. This is different from NEM, which all transfers amounts between two nodes are aggregated into one unilateral edge's weight[**nem**]. We presume NEM's solution is vulnerable to manipulations, since only a simple triangle loop will enhance edges' weight into infinity. We will show the advantage of our aggregation method in §**??** ~~experiment confirming why doing so~~

5.3.3 Temporality Embedding

We noticed that the transactions happens with timestamps. So we try to embed this temporal information as a property of nodes. For each account, we calculate its coinage by the following pseudo-code.

~~formula~~ ~~Defined as C_v normalized by max~~

The intuition of coinage is ~~insights~~.

Besides, we conjecture that reducing each transaction's contribution according to its block height, like NEM does[**nem**], encourages users to postpone their transferring until the last day of period, which will cause unnecessary confusion. Instead, **Nebulas Rank** treats each transaction equally, which encourages every account to keep active all the time.

We will talk about the coinage exploitation in §5.3.5. And we will show the advantage of our solution in §**??** ~~experiments confirming the effect of coinage~~.

5.3.4 Encouragement Function

~~formula and intuition~~ ~~defined as B_v normalized by max~~

We will talk about how we apply the encouragement function in §5.3.5. And we will show its advantage in §**??** ~~experiments confirming the effect of encouragement function~~.

5.3.5 Exploiting Nodes' Property

We defined two node properties C_v and B_v in §5.3.3 and §5.3.4, respectively. Then we reduce each edge's weight by its target node's properties:

$$w_{(.,v)} \leftarrow w_{(.,v)} \times ln(1 + \frac{C_v + B_v}{2}) \tag{3}$$

活跃币龄(Active Coinage)

活跃币龄：在当前周期内，某个地址新接收和新发出的币为活跃币，这些币的币龄为该地址的活跃币龄

```
def calculate(cur, genesis, nodes):
    for node in nodes:
        # cal coinage according to the active coins
        if node.balance < 0:
            coinage = (genesis.no - cur.no) * node.balance
            node.coinage += coinage
            node.balance = 0
        else:
            node.coinage += node.balance

def cal_coinage(nodes, transactions):
    # first block
    genesis = transactions[0].block
    for trans in transactions:
        # next block
        if cur.no < trans.block.no
            # cal coinage till current block
            calculate(cur.no, genesis.no, nodes)
            cur = trans.block
        # record balance
        edge.from.balance -= edge.coins
        edge.to.balance += edge.coins
    # cal coinage till current block
    calculate(cur, genesis, nodes)
```

5.3.6 Mitigating Dormant Effect

Consider a node receiving a large amount of money but does not spend any. This node forces its money to be "dormant " and prevents money from being circulated, which contradicts with **Nebulas Rank**'s purpose(§5.1). Thus we need to mitigate this dormant effect. In detail, we consider 1-hop local information of each node, limiting the amount of its in-transfers by the total amount of its out-transfers:

$$w_{(.,v)} \leftarrow \frac{w_{(.,v)}}{\sum_u (w_{(u,v)})} min\{\sum_u w_{(v,u)}, \sum_u w_{(u,v)}\}. \tag{4}$$

Intuitively, such restricting method is reasonable: 1) Imagine two phases for a piece of blockchain token. First it is made out of thin air, which is as the reward of the system. Second it is either circulated around the whole network, which almost never stops being transferred from accounts to accounts, or enters dormant state, which the last owner does not spend it out. Formula 4 does not affect the first phase, as edges weights can only be reduced as in-links. And in the second phase, accounts are encouraged to spend enough money in order to improve their in-link quality. 2) From the perspective of money flow, only circulated money should be counted. Nodes with dormant money does not control much of the network flow. That is, deleting these nodes does not affect interactions among other nodes. So formula 4 conforms with **Nebulas Rank**'s Liquidity value (§5.1).

We will show the how **Nebulas Rank** benefits from mitigating dormant effect in §**??**.

~~will this affect wgc????~~

5.4 LeaderRank

We build our scoring algorithm based on LeaderRank[**Li2014**][**Chen2013**]. It does minor modification on the famous PageRank algorithm[**Brin2010**][**page1999pagerank**]. The ranking algorithm can be understood as a Markov chain. States are nodes. Transition probability is proportional to the weight of some node's out-edge. By PageRank, if there is no out-edge from some node, aka Dangling Node[**Kim2002**], the strategy from transition probability from corresponding node is that evenly among all other pages

It can be described by an iterative process:

$$R_j^{t+1} = \sum_u R_j^t \times \frac{w_{(j,i)}}{\sum_k w_{(j,k)}} \tag{5}$$

~~explain~~

We add one ground node $\mathcal{G}$ into the network, and double link it with every other node. First every non-ground node gives and receives an amount of "altruist" money to the the ground node: $\forall v \in V$, $w_{(\mathcal{G},v)} = \beta \times mean(W)$, $w_{(v,\mathcal{G})} = \alpha \times mean(W)$. Then every non-ground receives an amount of "bonus" money from the ground node: $\forall v \in V$, $w_{(\mathcal{G},v)} \leftarrow w_{(\mathcal{G},v)} + \gamma \sum_u w_{(u,v)}$. Then ~~mathmatical proof to adopt power method~~.

Thus we build a probability matrix based on the graph with ground node. $H_{(N+1)\times(N+1)} = h_{ij}$, $\forall i, j \in V \cup \{\mathcal{G}\}$, $h_{ij} = w_{ji}/\sum_k w_{jk}$. It holds that $\forall v \in V \cup \{\mathcal{G}\}$, $\sum_u h_{uv} = 1$.

Using power method as follows, we are able to compute the rank for every node:

Initially we have $P^0 = \mathcal{R}^{n\times 1}$, $\forall v \in V, P_v^0 = 1/n, P_{\mathcal{G}}^0 = 0$; At each iteration $i = 1, 2 \ldots$, $P^i = H \times P^{i-1}$; After convergence, we get P^*, then distribute ground node's rank among every other node evenly, i.e. $\forall v \in V$, $P_v^* \leftarrow P_v^* + \frac{P_{\mathcal{G}}^*}{N}$.

~~explain~~

the out link weight from each node to ground node is of the same amount, causing different damping factors for each node, i.e. the less a node's out degree is, the more likely it jumps to an arbitrary node. Considering receiving from this kind of arbitrary jumping, PageRank actually gives high probability to nodes with lower in-degree, which may be overestimate the quality of new nodes.[formula support here] But Weighted LeaderRank[**Li2014**] assigns more probability to nodes with more in-degree. In the context of blockchain transaction, new accounts are more likely to be sybil nodes, thus the latter is more plausible. However, there could also be more weighting scheme for LeaderRank. ~~Consider a node with large incoming value but little outing value. Such node is preventing money from flowing, so it is explainable to distribute its "surplus value", namely incoming value minus outing value, to the ground~~

~~node.~~ In our design, each node sends and receives ”basic flow” of same amount to and from the ground node. And each each node also receives ”bonus flow”, which depends on the node’s in-degree, from the ground node. The rank in steady state represents the expected amount of money flow through every node. We will talk about the detail of our Weighted LeaderRank in

5.5 Experiments

5.5.1 Ethereum Stats

~~degree - avg neighbor degree and dynamics; hhi~~

5.5.2 Exploitability of 1-hop Local Information

5.5.3 Noise Resistance

5.5.4 Sybil Attack Resistance

~~all with comparison~~

5.6 Discussions And Conclusions

6 PoR 共识算法

jingchan

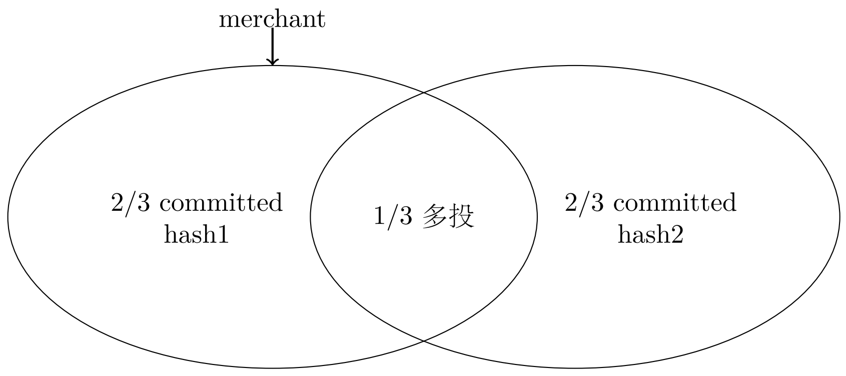

Figure 1: 双重支付攻击示意图

7 DIP 开发者激励协议

Shangshu

8 Nebulas Force

wenbo

9 智能合约

Wenbo

10 星云链基础服务及开发工具