SkyHash: an Opinion Dynamics Model for Hash Consensus over P2P network

Houwu Chen and Jiwu Shu
Department of Computer Science and Technology
Tsinghua University
{chenhw11@mails., shujw@}tsinghua.edu.cn

***Abstract*—Traditional Byzantine consensus does not work in P2P network due to Sybil attack while the most prevalent Sybil-proof consensus at present still can't resist adversary with dominant compute power. This paper proposed a two-layered opinion dynamics model named SkyHash for hash consensus over P2P network. For hash consensus, failures are constrained due to the difficulty to create collision with big hash size, however, we identified DoS attack and extent our model to a DoS-proof one. Simulations show that on the SNAP dataset of the Wikipedia who-votes-on-whom network with reasonable latencies, the network will reach consensus within 45 seconds, and it can also tolerant DoS attack committed by 7% random nodes or 0.9% top influential nodes, where no correct nodes decide on different hashes and 4% nodes refuse to decide, at the cost of 50% reduction of throughput. Comparing to compute power based consensus, our approach can resist any faulty or malicious nodes by unfollowing them. To the best of our knowledge, it's the first work dedicated to hash consensus on P2P network based on opinion dynamics.**

I. Introduction

P2P network is well known on its decentralized nature that increases robustness because it removes single points of failure. Emerging cryptocurrencies(e.g., Bitcoin) demonstrate the demand of consensus over P2P network with decentralization still retained [1]. However, to keep decentralization, no logically central and trusted authority vouches for a one-to-one correspondence between entity and identity, thus makes it difficult to resist Sybil attack, wherein a adversary creates a large number of pseudonymous identities to gain a disproportionately large influence [2]. Traditional Byzantine consensus algorithms that tolerate only a fixed fraction faulty nodes are not useful in P2P network with the presence of Sybil attack [3]. Existing consensus based on compute power can be Sybil-proof but can't resist adversary with dominant compute power [4].

Opinion dynamics is a field where mathematical-and-physical models and computational tools are utilized to explore the dynamical processes of the diffusion and evolution of opinions in human population if each body only only takes local interactions with its contacts [5]. In our previous work, we proposed the *sky* framework to apply opinion dynamics in P2P network for consensus, as well as the *sky* model to maximize performance for binary consensus [6]. However in the scenario of cryptocurrency, each node packs new transactions it receives from the time of last consensus into a block, the whole network need to determine which one of those blocks to agree at, thus the problem is actually consensus on a dynamic set which might be different for each node, and there is no direct way to convert the consensus problem to binary one without breaking decentralization.

In this paper, we proposed *SkyHash*, an opinion dynamics model for hash consensus under the *sky* framework. The model consists of a bit layer and a hash layer, and the bit layer is actually the *sky* model applied in each bit position of hashes and result in a pseudo hash, while the hash layer choose from hashes so that the selected hash has the minimal Hamming distance to the pseudo hash. For hash consensus, failures are constrained due to the difficulty to create collision with big hash size such as 256b, however, we identified DoS attack and extent our model to a DoS-proof one. Simulations show that on the SNAP dataset of the Wikipedia who-votes-on-whom network[6] with reasonable latencies, the network will reach consensus within 45 seconds, and it can also tolerant DoS attack committed by 7% random nodes or 0.9% top influential nodes, where no correct nodes decide on different hashes and 4% nodes refuse to decide, at the cost of 50% reduction of throughput. Comparing to compute power based consensus, our approach can resist any faulty or malicious nodes by unfollowing them. To the best of our knowledge, it's the first work dedicated to hash consensus on P2P network based on opinion dynamics.

II. Related Work

Sybil Attack Resistance One approach to resisting Sybil attack is relying on a certifying authority to perform admission control, which will break decentralization [7]. Another approach is remotely issuing anonymous certification of identity by identifying distinct property of a node, e.g, utilizing geometric techniques to establish location information, but it's unreliable in a network with changing environment [8]. Puzzle computing is also introduced to increase the cost of Sybil attack, such puzzles involve posing a challenge that requires a large amount of computation to solve but is easy to verify [9], however, there's no way to resist Sybil attack if the adversary has dominant computing resources. Sybil prevention techniques based on the connectivity characteristics of social graphs is another direction, because of the difficulty to engineer social connections between attack nodes and honest nodes, this approach is considered to be more robust over other ones [10].

Name	Wiki
Nodes Counts	998
Average Degree	33.33
Diameter	5
Average Path Length	2.34
Density	0.033
Average Clustering Coefficient	0.183
Eigenvector Centrality Sum Change	0.029

Table I
DATASETS PARAMETERS

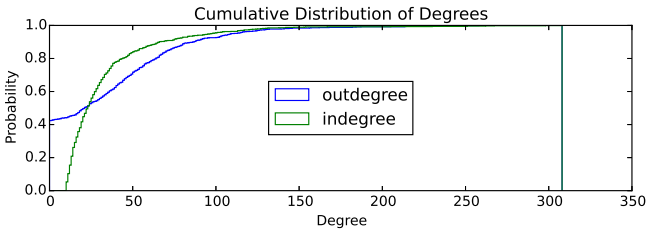


Fig. 1. Degree distribution of the wiki dataset

Cryptocurrency Bitcoin provides Sybil-proof consensus mechanism through an ongoing chain of hash-based proof-of-work(PoW) [1], which is actually a puzzle computing based approach. However, one has dominant compute power can control the network while the rest of the network has no means to resist it, and the proliferation of ASIC miner and mining pools already leads to the monopoly of compute power [11], [4]. Ripple/Stellar [12] also use a relationship based solution to resist Sybil attack similar to ours, however, their algorithm has a major defect that it relies on the assumption that for a node, if 80% of its followees agree on a opinion, then 80% of all nodes agrees on the same opinion, but the assumption only stands when a node follows an overwhelming majority of all nodes. As reported, Ripple/Stellar and other existing solutions like PoS have problem even bigger than PoW [13], [14].

III. THE PROBLEM AND DATASETS FOR EVALUATION

In traditional definition of consensus, each node has a initial value, the consensus problem is to decide upon a common value among all nodes. A node is *correct* if it behaves honestly and without error. Conversely, a node is *faulty*. In a P2P network, an adversary can always isolate some number of correct nodes in eclipse attack [15], hence *almost-everywhere consensus* is the best one can hope for in such networks [16]. Similar to existing definition [17], *almost-everywhere consensus* is defined that up to εn correct nodes in a P2P network does not agree at the common value as the majority of the nodes, where n is the network size, and $\varepsilon > 0$ is sufficiently small. We use the term *opinion* and *value* interchangeably in later sections following the convention of opinion dynamics.

We evaluate our approach on the SNAP dataset of Wikipedia who-votes-on-whom [18] called as the *wiki* dataset in later sections, because it presents trust relationships in the form of votes for administration. We also impose a constraint which can be enforced in P2P client of each correct node that $indegree >= 10$, thus all nodes with followees less than 10 are removed. Parameters of the result network is shown in Table I, and the cumulative distributions of indegrees and outdegrees are shown in Fig. 1.

To facilitate comparing the impact of network size, we also run simulations upon several uniform networks with size of 100, 1000, 5000 and 20000 nodes, where each node has the same degree and connect to each other randomly. Those dataset are named as *uniform-l00*, *uniform–1k*, *uniform–5k* and *uniform–20k* respectively.

IV. OVERVIEW OF THE SKY FRAMEWORK

The *sky* framework is designed to apply opinion dynamics for consensus over P2P network, and its detail is presented in [6]. Each node in the P2P network is owned by somebody and identified by a public key. When the owner of node A trusts the owner of node B, owner of A can set A to follow B in the P2P client, and B is called as *followee* while A is called as *follower*. The network can be abstracted to a directed graph where each peer is a node, and each trust relationship is a directed edge. To ensure connectivity and safety, each correct node is constrained by the P2P client to have at least a minimum number of followees.

Nodes are equally privileged and equipotent participants in the consensus process in any time as ordinary opinion dynamics. However, we introduced the concept of *round* into the consensus process which is commonly used in existing Byzantine consensus but not in opinion dynamics. Starting from a initial state as the first round, each correct node separately determines when to finish its current round and decides its new value following a common rule according to its current value and the values of its followees, and then enters the next round. The common rule is actually the algorithm from the aspect of programming, but to follow the convention of opinion dynamics, we use the term *model* in this paper.

A followee unidirectional broadcasts signed messages to all its followers. We allow a faulty node's signature to be forged by an adversary, thereby permitting collusion among the faulty nodes. Broadcast is implemented by DHT and asymmetric cryptography. For a node as followee, all its followers and itself form a sharing group(known as a "swarm") identified by the followee's public key. Each broadcasted message is signed with the private key of the followee, and the follwers can check the identity and integrity against the followee's public key.

Note here to avoid centralization no global clock or coordinator is used, each node decides how and when to enter next round separately, thus each node may enter the same round in different time. To deal with the problem in asynchronous system pointed by FLP impossibility [19], the framework use a *message filter* as well as a *failure detector* which can make mistakes by erroneously adding nodes to its list of suspects [20]. A node makes its final decision when enough rounds(e.g., 40) passed, and a node may refuse to agree at a hash finally when no hash from its followees is overwhelming.

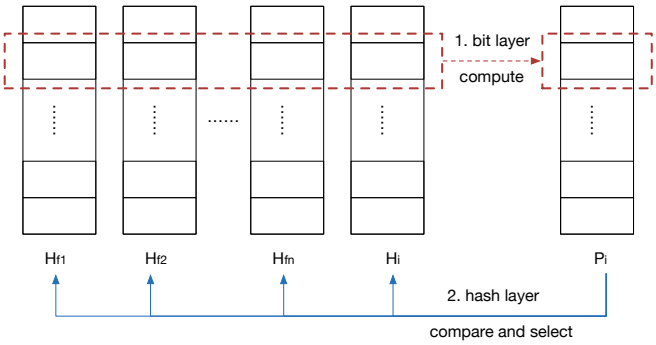


Fig. 2. Layers of the SkyHash model

V. The SkyHash Opinion Dynamic Model

At time t, a node receives all the messages broadcasted by its followees at $t - dt$, then finishes processing the received messages and broadcast its new opinion at t. By designating the opinion of $node_i$ at time t to be $H_i(t)$, the model can be expressed as a function $\mathcal{F}$:

$$H_i(t + dt) = \mathcal{F}(H_i(t), V_i(t)) \tag{1}$$

where $V_i(t) = [H_{f_1}(t), H_{f_2}(t), \dots H_{f_n}(t)]$ and $f_1, f_2, \dots j_n$ are followees of $node_i$. We also denote $H_i = b_{i,1}b_{i,2}\cdots b_{i,k}$, where $b_{i,j} \in \{0, 1\}$ for $j \in [1, k]$ is the value of bit at position j in H_i, and k is the hash size. In the following we also denote $V = \{H_i(t), V_i(t)\}$.

The SkyHash model consists of a bit layer and a hash layer shown in Fig. 2, the bit layer computer a *pseudo hash* P by applying the *sky* model(proposed in [6]) for each bit position separately, and the hash layer choose a hash $newH$ from V such that $newH$ has the minimal Hamming distance to P.

Bit Layer For each node at time t, for each bit position j, we denote $n0_j(t)$ and $n1_j(t)$ to be the count of 0 and 1 in position of each hash in V respectively. The bit layer *sky* model randomly selects one from the following two items:

1) If $n0_i(t) > n1_i(t)$, then set new opinion to 0, and vise versa, while if $n0_i(t) = n1_i(t)$, then select from {0, 1} randomly.
2) If $n0_i(t) > 4 * n1_i(t)$ then set new opinion to 0, while if $n1_i(t) > 4 * n0_i(t)$ then set new opinion to 1. Otherwise set new opinion to 0 with probability of $n0_i(t)/(n0_i(t) + n1_i(t))$ and set new opinion to 1 with probability of $n1_i(t)/(n0_i(t) + n1_i(t))$.

Hash layer Hash layer sub algorithm can be expressed as a function F_h to choose $newH$ from V for each round by $newH = \mathcal{F}_h(P, V)$, where P is the pseudo hash computed from bit layer model. It choose $newH \in V$, such that $\forall H_i \in V, i \in [1, n]$, and $H_i \neq newH$, $dist(newH, P) \leq dist(H_i, P)$, where $dist$ is the function to calculate the Hamming distance between two hashes.

VI. Convergence Analysis

A. Mean Field Analysis

Due to the difficulty to directly analyze the stochastic process of the interactions between every nodes described in Eq. (1), we analyze our opinion dynamics model using *mean field theory(MFT)*, which studies the behavior of large and complex stochastic models by studying a simpler model. Such models consider a large number of small individual components which interact with each other. The effect of all the other individuals on any given individual is approximated by a single averaged effect, thus reducing a many-body problem to a one-body problem [21]. MFT is widely used in opinion dynamics as an effective modeling method [5]. By MFT, the opinion dynamics model can be expressed by a continuous differential equation, and the *round* can be regarded as $dt = 1$ in the corresponding equation shown in Eq. (2).

1) Bit Layer: Since the bit layer *sky* model is already analyzed in detail in [6], only the final mean field equation is introduced in this paper. We denote the densities of bit j in correct nodes to be $c = c0_j + c1_j$ where $c0_j$ and $c1_j$ are the densities of correct nodes with opinion of 0 and 1, and densities of faulty nodes to be $f = f0_j + f1_j + fs_j$ where $f0_j$ and $f1_j$ are the density of faulty nodes with opinion of 0 and 1 and fs_j are the density of faulty nodes without opinions broadcasted. So we have $c + f = 1$. We also denote densities of all nodes(including correct and faulty nodes) with opinion 0 and 1 to be $a0_j$ and $a1_j$ respectively, thus we have $a0_j = (c0_j + f0_j)/(1 - fs_j)$ and $a1_j = (c1_j + f1_j)/(1 - fs_j)$.

By designating the derivative of $c0_j$ on t to be $dc0_j/dt$ which is actually the change speed of $c0_j$, we can have Eq. (2) where $s1_j$ is the probability that a node flips from opinion 1 to opinion 0, and $s0_j$ is the contrary.

$$\frac{dc0_j}{dt} = -\frac{dc1_j}{dt} = s1_j c1_j - s0_j c0_j \tag{2}$$

We specify the mean indegree and outdegree of a node to be D, $F(k; n, p)$ is the *cumulative distribution function* and $d(k; n, p)$ is the *probability mass function* for k successes in binomial distribution of n trials with probability p, then Eq. (2) can be written as following:

$$\frac{dc0_j}{dt} = \frac{s_m 1_j + s_s 1_j}{2} c1_j + \frac{s_m 0_j + s_s 0_j}{2} c0_j \tag{3}$$

Where $s_m 1_j$, $s_m 0_j$ can be calculated from Eq. (4), and $s_m 1_j$, $s_m 0_j$ can be calculated from Eq. (5).

$$\begin{cases} s_m 1_j = F(\frac{D}{2} - 1; D, a1_j) + \frac{1}{2} d(\frac{D}{2}; D, a1_j) \\ s_m 0_j = F(\frac{D}{2} - 1; D, a0_j) + \frac{1}{2} d(\frac{D}{2}; D, a0_j) \end{cases} \tag{4}$$

$$\begin{cases} s_s1_j = F(0.2D; D, a1_j) + \sum_{i=0.2D}^{0.8D} d(i; D, a1_j)(\frac{D-i}{D} + \frac{1}{2D}) \\ s_s0_j = F(0.2D; D, a0_j) + \sum_{i=0.2D}^{0.8D} d(i; D, a0_j)(\frac{D-i}{D} + \frac{1}{2D}) \end{cases} \tag{5}$$

2) Hash Layer: To analysis the hash layer model, we start from answering this question: for hash length of k, density of each bit with value 0 in pseudo hash is p_p, and density of each bit with value 0 in D candidate hashes is p_h, then what's the final density(denoted as p_f) of each bit with value 0 in the selected hash when we select the hash from the candidates with the minimal Hamming distance to the pseudo hash?

Due to capacity of this paper, the result is given directly by the following equation without intermediate reasoning:

$$p_f(D, k, p_h, p_p) = \sum_{l=0}^{k}\sum_{m=0}^{k}\sum_{n=0}^{k} prob(l, m, n), n \le l, n \le m \tag{6}$$

where:

$$\begin{aligned} prob(l, m, n) = &\frac{l+m-2n}{k} \\ &\cdot \sum_{u=0}^{D} p_{eq} p_{gt}^{j} (p_{eq} + p_{gt})^{D-u-1} d(m; k, p_p)^{1-D} \end{aligned} \tag{7}$$

Here $d(m; k, p_p)$ is the possibility mass function described above, and

$$\begin{cases} p_{eq} = p_{dists}[l, m, n] \\ p_{gt} = \sum_{x=l}^{k}\sum_{y=0}^{k} p_{dists}[x, m, y] \end{cases} \tag{8}$$

and $p_{dists}[l, m, n]$ is calculated by the following equation:

$$\begin{aligned} p_{dists}[l, m, n] = &(\binom{k}{n} \sum_{y=0}^{k} (p_p(1-p_h))^y) \\ &\cdot (\binom{k-n}{m-n} \sum_{y=0}^{k} (p_p p_h)^{m-n}) \\ &\cdot (\binom{k-m}{l-n} \sum_{y=0}^{k} ((1-p_p)p_h)^{l-n}) \\ &\cdot ((1-p_p)(1-p_h))^{k-m-(l-n)} \end{aligned} \tag{9}$$

Then considering the bit layer and hash layer models as a whole, we can have the following equation:

$$\frac{dc0}{dt} = p_f(D, k, c_0, (c_0 + dc0_j) - c_0 \tag{10}$$

where p_f is described in Eq. (6), and $dc0_j/dt$ is described in Eq. (3).

However, Eq. (10) is inaccurate because it models the case that bit in one position is uncorrelated with bit in any other positions for a hash in mean field. But in fact the correlation between different bits of a hash increases along with the increasing of convergence.

3) Numeric Analysis: To analysis the convergence, we only consider the case where all nodes are correct while the case with faulty nodes are analyzed later. Thus we can have $a_0 = c_0$ and $a_1 = c_1$. Because the model is symmetric on binary opinion 0 and 1, and $c_0 + c_1 = 1$, it's sufficient to only track c_0 and consider $c_0 \ge 0.5$. According to the mean field equations, dc_0/dt(a.k.a. the change speed of c_0) and $\int \frac{dc_0}{dt} dt$ (a.k.a c_0) are demonstrated in Fig. 3a and Fig. 3b respectively for $D \in \{8, 16, 32, 64, 128, 256\}$. From Fig. 3a we can see that $\forall c_0 \in (0.5, 1)$ and $\forall D > 0$, change speed of c_0 is always positive, i.e., c_0 strictly increases with time t. From Fig. 3b we can see that network with greater degree D will converge more quickly. We can also see that with a tiny deviation of c_0 from 0.5, even when $D = 16$, c_0 can still converge to 1 within 40 rounds.

B. Simulations

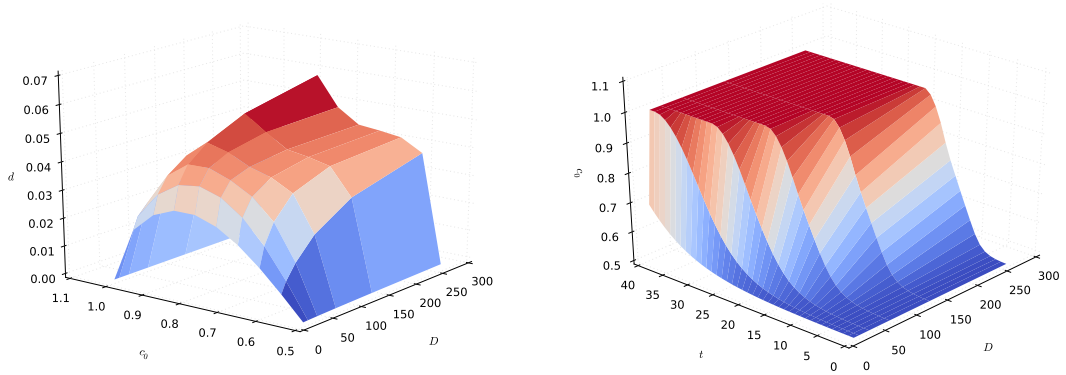
To study the convergence performance, we simulate the model on the uniform–1k dataset with several hash sizes as shown in Fig. 4a, as well as on all the datasets with hash size of 256b as shown in Fig. 4b. The vertical axis is the *density of the top hash* which is the hash with the most number of nodes agrees at the time. Fig. 4a shows that greater hash size leads to quicker convergence. Fig. 4b demonstrates that for all the uniform dataset, rounds to converge increases with node count and approaches to the theoretical result, that's because mean field equation works best when $N \to \infty$.

The impact of correlations between different bit positions on the uniform–1k dataset are shown in Fig. 4c which shows the case that initially all the nodes randomly choose from 2, 16, 128 and 1024 hashes respectively. In all this cases, densities of each bit position with value 0 and 1 are both 0.5, but the correlations between different bit positions are different. Fig. 4c shows that round needed to reach consensus decreased with greater correlations. Follow this observation, we can assume that in an accurate theoretical result, round needed to reach consensus should be smaller than the one shown in Fig. 4b.

VII. Fault Tolerance

Sybil attack of the *sky* framework is already analyzed in our previous work [6], thus in this paper we focus on non-Sybil failures. Under Byzantine failures, a faulty node can behave arbitrarily, it may not run according to the opinion dynamic model, immune to the hashes broadcasted by its followees and even colludes with other nodes.

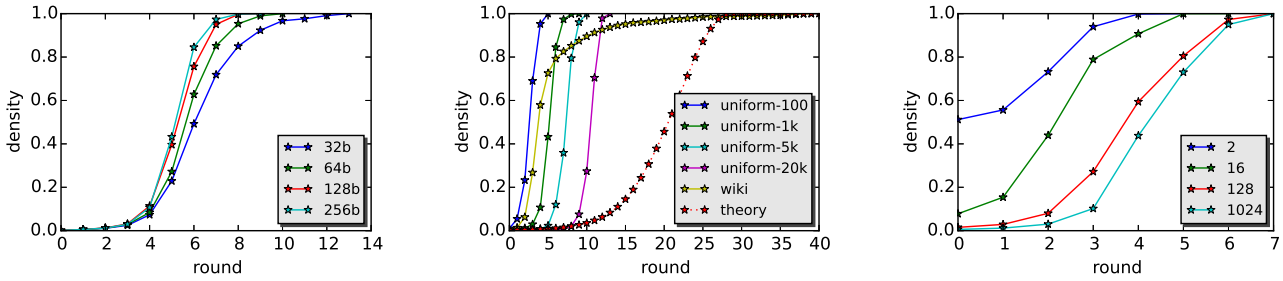
It's impractical to analyze against all possible failures, however, *time* related failures such as stop failure or delay is already handled by the sky framework based on failure detector, we only analyze *value* related failures here. For big hash size such as 256b, it's impossible at present to elaborate data so that its hash is same as a given value(aka. hash collision), thus a faulty node can't broadcast arbitrary hashes as its will, and its ability to compromise the network is also



(a) Derivative of c_0 on t

(b) Evolution of c_0

Fig. 3. Numeric analysis for hash size of 256b



(a) Convergence on the uniform-1k dataset with several hash sizes

(b) Convergence on all datasets with hash size of 256b

(c) Impact of correlations between different bit positions on the uniform-1k dataset

Fig. 4. Simulation of the SkyHash model without faulty nodes

constrained. Experiments(not presented in this paper) shows that faulty nodes broadcasting random hashes in each round has little impact, similar to our previous research in binary consensus [6]. However, also similar to the result of the previous research that faulty nodes might mislead the consensus by colluding to keep broadcasting one value(e.g. 0) in binary consensus, denial of service attack might be committed as we introduced in the following. Another potential attack vector is producing partial hash collision, i.e., elaborate data so that part of the hash is exactly what an adversary wants even it can't elaborate the whole hash, but we have not find a way to exploit it yet, especially considering the *DoS-proof model* introduced below.

A. Denial of Service

Each hash represents a block with a number of transactions packed in it, though the transactions are ensured to be valid or the hash will be refused by correct nodes, a group of malicious nodes can still selectively choose which transactions to be packed and if the group can have the network always agree at hashes from itself, in this case valid transactions may not be served forever, and those type of attack is called denial of service(DoS) attack.

Experiment not presented also shows that when each correct node proposes a hash generated by itself, even 0.5% malicious nodes colluding together to always broadcast the same hash can make the network agree at the hash they proposed, thus successfully commit DoS attack.

A naive idea on dealing with DoS attack is to make each correct node refuse any hash whose corresponding data does not include unserved transactions it seen. but it's unreliable because each node may receive an individual transaction in different time. However, our previous research shows that lower bound of the faulty tolerance performance for binary consensus shows that for the density of correct value for a bit is near 0.5, the fault tolerance performance is poor, and the performance increases with the density [6]. Based on the observation, we proposed the idea to resist DoS attack by leveraging the strength from the adversary itself to cultivate competitive rivals in correct nodes, and more powerful adversary leads to more powerful rivals, thus increase the density of non-DoS hash.

DoS-proof model For each correct node, it consists of two phases, a *reverse* phase and a *normal* phase. Given round threshold R, a node is in the *reverse* phase when $r <= R$, the hash layer of reach node's round chooses $newH \in V$, such that $\forall H_i \in V, i \in [1, n]$, and $H_i \neq newH_i$, $dist(newH, P) \geq dist(H_i, P)$, while when $r <= R$, a node

behaves exactly the same as the case without DoS-proof.

Simulation on the uniform–1k dataset with DoS attack from 11%, 15% and 20% nodes is shown in Fig. 5 with round threshold $R = 15$ in the DoS-proof model. Green lines are the cases that the network succeeds to resist the DoS attack, where all correct nodes agrees at a hash which is not the hash(called as *DoS hash*) proposed by DoS attack nodes. Red lines are the cases that the network fails to resist the DoS attack, where all correct nodes agrees at the DoS hash. Solid lines are the density of the hash proposed by the DoS attack nodes, and dashed lines are the density of the top hash(may be DoS hash or non-DoS hash) defined before.

The network will survive in DoS attack by less than 15% nodes, where all nodes still agrees on the same valid hash in each run, but 50% of the runs will agrees on the hash proposed by the DoS attack nodes, thus the throughput will decrease to 50% of the case without DoS attack. Fig. 5a and Fig. 5b demonstrate the oscillation of the density of DoS hash, and the heavier the attack, the smaller range the oscillation, until the oscillation is unobvious thus in all runs the network will always agree on the DoS hash as in Fig. 5c.

VIII. Experiment

Since to the best of our knowledge, this paper is the the first work to bring opinion dynamics to P2P network for hash consensus, there is no previous work to compare by experiments, this paper only presents the experiment of our *SkyHash* model.

According to existing studies, latency between peers in DHT is mostly between 50 to 1000 ms [22]. In our experiment, we employ a simply latency model that the time for each message to be delivered conforms gauss distribution of ($\mu = 500$, $\sigma = 500$) with lower cutoff of 50 and no upper cutoff which means a message may never be lost in a small probability even if the node broadcasts it is correct, we also set $timeout = 2000$ for the failure detector and round threshold $R = 15$ in the DoS-proof model.

Fig. 6 exhibits the experiment results on the wiki dataset. Green lines are the cases that the network succeeds to resist the DoS attack, where all correct nodes agrees at a hash which is not the hash proposed by DoS attack nodes. Red lines are the cases that the network fails to resist the DoS attack, where all correct nodes agrees at the hash(called as DoS hash) proposed by DoS attack nodes. The vertical axis is the density of the top hash which is the hash with the most number of nodes agrees at the time.

Under the *SkyHash* model the wiki dataset can survive under DoS attack committed by 7% random nodes or 0.9% top influential nodes defined as the first 0.9% nodes by sorting all nodes in descendant order on the count of a node's followees, however, the throughput will decrease 50% even when the network survives. In all the cases that the network survives, correct nodes can always reach almost-everywhere consensus within 45 seconds without correct nodes agree at different values, while under DoS attack by 7% random nodes, 1.5% nodes refuse to agree at any values, and under DoS attack by 0.9% top influential nodes, 4% nodes refuse to agree at any values.

Proportion of compute power	Success probability
50%	100%
40%	50.398%
30%	13.211%
20%	1.425%
10%	0.024%

Table II
BITCOIN ATTACK SUCCESS PROBABILITY

As we introduced in Section II, Bitcoin's PoW is the best Sybil-proof consensus at present, but it is a different mechanism to our work and not comparable directly in Fig. 6. Through the automatic adjustment of the difficulty of PoW, Bitcoin generates a block in about 10 minutes, and a fully confirmed consensus need 6 blocks thus needs about 1 hour. However if a single node or a group of nodes has a large proportion of compute power, it can compromise the network and create a fork. Table II shows the probability of success attack for 6 blocks confirmations [23]. If one adversary in Bitcoin has a threatening compute power, the whole network can't do anything to resist it because the power is controlled by the adversary itself, while in our approach a node's power is controlled by its followees, thus a node can be unarmed by unfollowing it.

IX. Discussion and Conclusion

The mean field equation to analyze the hash layer model is still inaccurate, and due to the complexity of the SkyHash model we did not found a way to theoretically analyze the threshold of fault tolerance performance as our previous work on the bit level model. Even we've already identified the DoS attack, there might still be other types of attack not covered.

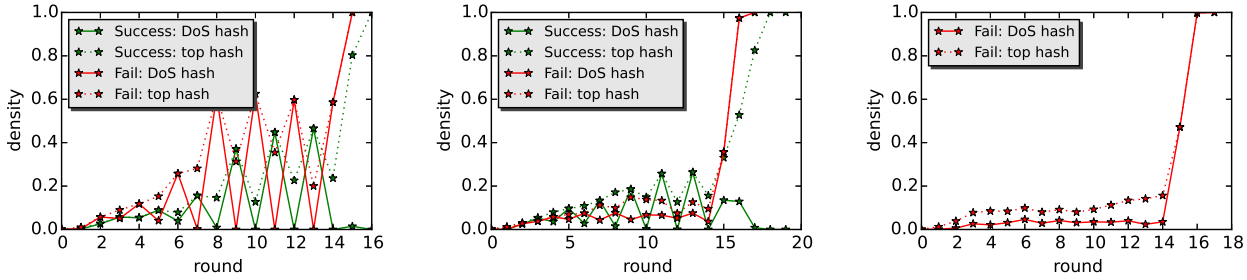
Although our approach can successfully runs over the wiki dataset, it also shows the convergence performance degrades comparing to the uniform dataset, and existing studies show that community strength impacts the performance [24].

Sybil-proof consensus is still an open problem, and even the most prevalent Sybil-proof consensus at present still have a big problem that it can't resist adversary with dominant compute power. Opinion dynamics based approach presented in this paper is a new attempt to circumvent the problems of existing solutions. Theoretical and experimental result reveals that it has acceptable performance and the ability to resist any faulty or malicious nodes by unfollowing them. To the best of our knowledge, it's the first work to bring opinion dynamics to P2P network for hash consensus.

References

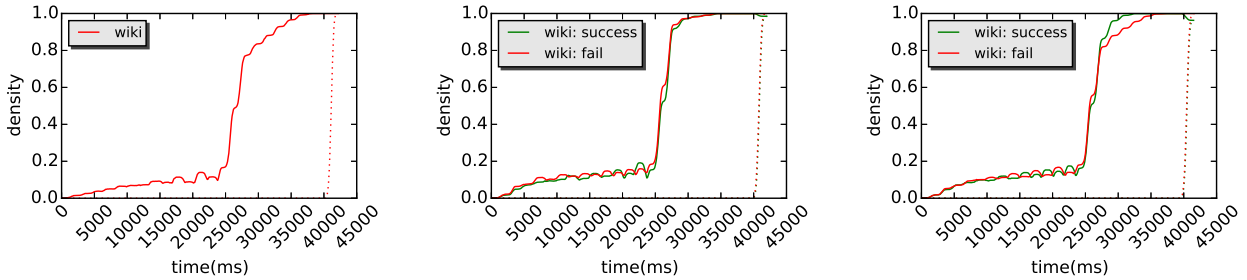
[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." http://www.bitcoin.org/bitcoin.pdf, 2009.

[2] J. R. Douceur, "The sybil attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, (London, UK, UK), pp. 251–260, Springer-Verlag, 2002.



(a) DoS attack by 11% nodes (b) DoS attack by 15% nodes (c) DoS attack by 20% nodes

Fig. 5. Simulation of DoS-proof model on the uniform-1k dataset



(a) Without DoS attack (b) DoS attack by 7% random nodes (c) DoS attack by 0.9% top influential nodes

Fig. 6. Experiments on the wiki dataset

[3] J. Aspnes, C. Jackson, and A. Krishnamurthy, "Exposing computationally-challenged Byzantine impostors," Tech. Rep. YALEU/DCS/TR-1332, Yale University Department of Computer Science, July 2005.

[4] N. T. Courtois and L. Bahack, "On subversive miner strategies and block withholding attack in bitcoin digital currency," *CoRR*, vol. abs/1402.1718, 2014.

[5] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," *Rev. Mod. Phys.*, vol. 81, no. 2, pp. 591–646, 2009.

[6] H. Chen, "Sky: an opinion dynamics framework and model for consensus over p2p network," *CoRR*, vol. abs/1501.06238, 2015.

[7] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 299–314, 2002.

[8] R. A. Bazzi and G. Konjevod, "On the establishment of distinct identities in overlay networks," in *Proceedings of the Twenty-fourth Annual ACM Symposium on Principles of Distributed Computing*, PODC '05, (New York, NY, USA), pp. 312–320, ACM, 2005.

[9] N. Borisov, "Computational puzzles as sybil defenses," in *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*, P2P '06, (Washington, DC, USA), pp. 171–176, IEEE Computer Society, 2006.

[10] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi, "SoK: The evolution of sybil defense via social networks," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, (Washington, DC, USA), pp. 382–396, IEEE Computer Society, 2013.

[11] D. Cawrey, "Are 51% attacks a real threat to bitcoin?." http://www.coindesk.com/51-attacks-real-threat-bitcoin/, June 20 2014.

[12] D. Schwartz, N. Youngs, and A. Britto, "The Ripple protocol consensus algorithm." https://ripple.com/files/ripple_consensus_whitepaper.pdf, 2014.

[13] J. Kim, "Safety, liveness and fault tolerance—the consensus choices stellar." https://www.stellar.org/blog/safety_liveness_and_fault_tolerance_consensus_choice/, 2014.

[14] A. Poelstra, "A treatise on altcoins." https://download.wpsoftware.net/bitcoin/alts.pdf, 10 2014.

[15] A. Singh, M. Castro, P. Druschel, and A. Rowstron, "Defending Against Eclipse Attacks on Overlay Networks," in *Proceedings of the 11th Workshop on ACM SIGOPS European Workshop*, EW 11, (New York, NY, USA), ACM, 2004.

[16] C. Dwork, D. Peleg, N. Pippenger, and E. Upfal, "Fault Tolerance in Networks of Bounded Degree," in *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, (New York, NY, USA), pp. 370–379, ACM, 1986.

[17] J. Augustine, G. Pandurangan, and P. Robinson, "Fast Byzantine Agreement in Dynamic Networks," in *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*, PODC '13, (New York, NY, USA), pp. 74–83, ACM, 2013.

[18] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection." http://snap.stanford.edu/data, June 2014.

[19] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, no. 2, pp. 374–382, 1985.

[20] T. D. Chandra and S. Toueg, "Unreliable failure detectors for reliable distributed systems," *J. ACM*, vol. 43, no. 2, pp. 225–267, 1996.

[21] "Mean field theory." http://en.wikipedia.org/wiki/Mean_field_theory.

[22] F. Dabek, J. Li, E. Sit, J. Robertson, M. F. Kaashoek, and R. Morris, "Designing a DHT for low latency and high throughput," in *Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1*, NSDI'04, (Berkeley, CA, USA), pp. 7–7, USENIX Association, 2004.

[23] "Majority attack - Bitcoin Wiki." https://en.bitcoin.it/wiki/Majority_attack, 2015.

[24] F. Gargiulo and S. Huet, "Opinion dynamics in a group-based society," *EPL (Europhysics Letters)*, vol. 91, p. 58004, Sept. 2010.