Sky: Opinion Dynamics Based Consensus for P2P Network with Trust Relationships[*]

Houwu Chen and Jiwu Shu

Department of Computer Science and Technology,
Tsinghua University, Beijing, China
{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 opinion dynamics based consensus for P2P network with trust relationships, consisting of the sky framework and the sky model. With the sky framework, opinion dynamics can be applied in P2P network for consensus which is Sybil-proof through trust relationships and emerges from local interactions of each node with its direct contacts without topology, global information or even sample of the network involved. The sky model has better performance of convergence than existing models including MR, voter and Sznajd, and its lower bound of fault tolerance performance is also analyzed and proved. Simulations show that our approach can tolerate failures by at least 13% random nodes or 2% top influential nodes while over 96% correct nodes still make correct decision within 70 seconds on the SNAP Wikipedia who-votes-on-whom network for initial configuration of convergence>0.5 with reasonable latencies. 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 to bring opinion dynamics to P2P network for consensus.

Keywords: opinion dynamics, P2P, Byzantine consensus, Sybil attack

1 Introduction

Emerging cryptocurrencies(e.g., Bitcoin) demonstrate the demand of consensus over P2P network [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

[*] To appear in *The 15th International Conference on Algorithms and Architectures for Parallel Processing* (ICA3PP 2015).

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. Based on observations of existing studies that opinion might converge when nodes only take local interactions [5], we proposed the *sky* framework to apply opinion dynamics in P2P network for consensus, as well as the *sky* model to maximize performance. In the sky framework, each node is identified by its public key, other nodes follow the node if they trust it, during the process of consensus, each node broadcasts opinion to its followers, which then decide new opinions according to their own followees conforming the rules of the sky model. The sky model has better performance of convergence than existing models including MR, voter and Sznajd, and its lower bound of fault tolerance performance is also analyzed and proved. Comparing to compute power based approach, ours enables disarming faulty or potentially malicious nodes by unfollowing them. Theoretic analysis and simulations both show that it can tolerate failures by at least 13% random nodes while over 96% correct nodes still make correct decision for initial configuration with convergence $\geq$ 50%. Simulations also show that on the SNAP dataset of the Wikipedia who-votes-on-whom network [6] with reasonable latencies, it can reach almost-everywhere consensus within 70 seconds and tolerate failures committed by 2% top influential nodes. To the best of our knowledge, it's the first work to bring opinion dynamics to P2P network for consensus.

2 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]. Those approaches don't target at the consensus problem directly but provide good bases.

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 [4, 11]. Ripple [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 and other existing solutions like PoS have problem even bigger than PoW [13, 14].

3 The Problem and Evaluation Datasets

In traditional definition of consensus, specifically binary consensus, each node has a initial value $v_i \in \{0, 1\}$, 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 under an eclipse attack [15], an adversary can always isolate some number of correct nodes 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 agreed on the *wrong* value, where n is the network size, and $\varepsilon > 0$ is sufficiently small, the wrong value is 1 if initially 0 is the majority among all correct nodes, and vise versa. We use the term *opinion* instead of *value* in later sections following the convention of opinion dynamics.

We evaluate our approach on the SNAP dataset of Wikipedia who-votes-on-whom [6] 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 1, and the cumulative distributions of indegrees and outdegrees are shown in Fig. 1.

Table 1. Datasets parameters

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

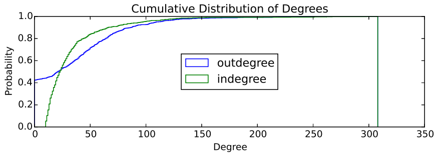


Fig. 1. Degree distribution of the wiki dataset

To facilitate comparing the impact of network size, we also run simulations upon three uniform networks with size of 100, 1000, 5000 nodes, where each node has the same degree and connect to each other randomly. Those dataset are named as *uniform-less*, *uniform* and *uniform-more* respectively.

4 The Sky Opinion Dynamics Framework

We proposed an opinion dynamics framework called the *sky* framework for consensus over P2P network as described in this section.

4.1 Network Constructing

In our framework, 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.

4.2 Consensus Process

Nodes in our framework 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 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 used here shapes *the opinion dynamics model* which will be introduced in section 5. 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.

A node makes its final decision when enough rounds(e.g., 40) passed. A node is *deciding* before making final decision. If a node finally agrees at 0 or 1, then it's *decided*. A node is *confused* if it's considered to be safe at neither 0 nor 1. For each node, by denoting the count of 0 and 1 in its current value and the values of its followees respectively, the final decision follows the following rules:

1. If $n0 > (n0+n1)*T$ then agree at 0 and the criteria to agree 1 is similar. The T constant controls the strategy to be aggressive or conservative. Greater T results that less nodes to agree at wrong opinion but more nodes to be confused. We use $T = 2/3$ in experiments.
2. If can't agree at 0 or 1, then it's *confused*.

4.3 Message Passing

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.

Each message broadcasted by $node_i$ is a tuple of ($nodeid$, $round$, $opinion$, $state$), where $nodeid$ is the id of $node_i$, $round$ and $opinion$ is its current round and opinion, and $state \in \{deciding, decided, confused\}$.

4.4 Message Handling

According to the well known FLP impossibility [18], consensus cannot be solved deterministically with even a single crash failure in an asynchronous system, because of the inherent difficulty of determining whether a process has actually crashed or is only "very slow" [19]. We use a *message filter* and a *failure detector* which can make mistakes by erroneously adding nodes to its list of suspects [19].

For a *node*, the message filter will refuse to accept any new messages if it has already made its final decision, and it will always keep at most one message from a followee with the largest round denoted as $round_{max}$ while $round_{max} \geq node.round$. The filter is applied when a node receiving a new message as well as when a node finish a round after broadcasting opinion to its followees.

The key idea of the failure detector is that each node maintains a followee nodes list as well as a suspect nodes list. A message is a *valid message* for a node marked as *node* if $msg.round \geq node.round$ or $msg.state \in \{decided, confused\}$. For each node, initially all followees are in the followee nodes list, in each round, a followee is moved to the suspect nodes list for the followee nodes list if no valid message from it in message buffer for a long time(failure detector time out), while a node is moved from the suspect nodes list to the followee nodes list when a new valid message from it is received.

With the help of message filter and failure detector, a node can apply the common rule which shapes the opinion dynamics model in the following way:

1. If a node received a message passed through the message filter, then it should check whether to apply the common rule or not.
2. On failure detector timeout event for each round, it should check whether to apply the common rule or not.
3. A node apply the common rule only when its message buffer has messages from all nodes in its followee nodes list.

5 The Sky Opinion Dynamics 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 $v_i(t)$, the model can be expressed as a function $\mathcal{F}$:

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

where $V_i(t) = [v_{f_1}(t), v_{f_2}(t), \dots v_{f_n}(t)]$ and $f_1, f_2, \dots j_n$ are followees of $node_i$. In later sections we designate the count of 0 and 1 in $\{V_i(t), v_i(t)\}$ to be $n0_i(t)$, $n1_i(t)$ respectively.

However, 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)* [20] which is widely used in opinion dynamics as an effective modeling method [5]. By MFT, the opinion dynamics model shaped by the common rule 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).

We denote the densities of correct nodes to be $c = c_0 + c_1$ where c_0 and c_1 are the densities of correct nodes with opinion of 0 and 1, and densities of faulty nodes to be $f = f_0 + f_1 + f_s$ where f_0 and f_1 are the density of faulty nodes with opinion of 0 and 1 and f_s 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 a_0 and a_1 respectively, thus we have $a_0 = (c_0 + f_0)/(1 - f_s)$ and $a_1 = (c_1 + f_1)/(1 - f_s)$.

By designating the derivative of c_0 on t to be dc_0/dt which is actually the change speed of c_0, we can have Eq. (2) where s_1 is the probability that a node flips from opinion 1 to opinion 0, and s_0 is the contrary.

$$\frac{dc_0}{dt} = -\frac{dc_1}{dt} = c_1 s_1 - c_0 s_0 \tag{2}$$

We adapt the paradigmatic majority rule(MR) model, and then proposed the *sky* model by incorporating the MR model with a simulated annealing(SA) model we proposed.

5.1 Majority Rule Model

Traditional *majority rule(MR)* model needs to select a group each time and then make all of the nodes in the group conform the majority opinion of the group, however, there's no such group in the sky framework. We adapt the MR model by regarding each node and all of its followees as a group, but instead of making all of the nodes inside the group to have the majority opinion, we just let the node itself to have that opinion without its followees changed. The rule is shown as following:

1. If $n0_i(t) > n1_i(t)$, then set new opinion to 0, and vise versa.
2. If $n0_i(t) = n1_i(t)$, then select from $\{0, 1\}$ randomly.

We specify the mean indegree and outdegree of a node to be D. According to the first rule, a node flips from opinion 1 to opinion 0 only when the count of its followees with opinion of 1 is less than $D/2$, and vice versa, and according to the second rule, when the count of its followees with opinion of 1 equals to $D/2$, it has probability of $1/2$ to flip, thus for Eq. (2), we can have the following equation:

$$\begin{cases} s_1 = F(\frac{D}{2} - 1; D, a_1) + \frac{1}{2} d(\frac{D}{2}; D, a_1) \\ s_0 = F(\frac{D}{2} - 1; D, a_0) + \frac{1}{2} d(\frac{D}{2}; D, a_0) \end{cases} \tag{3}$$

where $F(k; n, p)$ 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.

5.2 Simulated Annealing Model

The *simulated annealing(SA)* model we proposed provides nodes the ability to escape from their current opinion at some probability while keep stable if $n1_i(t)/n0_i(t)$ or $n0_i(t)/n1_i(t)$ is big enough for a node, as shown in the following:

1. 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.
2. 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 $n1_i(t)/(n0_i(t) + n1_i(t))$.

With the notations same as the previous section, for Eq. (2), we can have the following equation:

$$\begin{cases} s_1 = F(0.2D; D, a_1) + \sum_{i=0.2D}^{0.8D} d(i; D, a_1)(\frac{D-i}{D} + \frac{1}{2D}) \\ s_0 = F(0.2D; D, a_0) + \sum_{i=0.2D}^{0.8D} d(i; D, a_0)(\frac{D-i}{D} + \frac{1}{2D}) \end{cases} \tag{4}$$

5.3 Sky Model

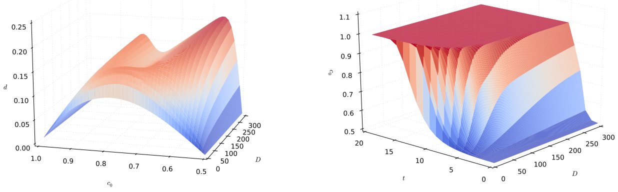
For each node, the *sky* model randomly selects a rule from the rules corresponding to the *MR* model and the *SA* model, thus dc_0/dt is a linear combination of that of the MR and the SA model as the the following equation, where $d_g c_0/dt$ is Eq. (2) with Eq. (3) and $d_s c_0/dt$ is Eq. (2) with Eq. (4):

$$\frac{dc_0}{dt} = \frac{d_g c_0}{dt} * ratio + \frac{d_s c_0}{dt} * (1 - ratio) = \frac{1}{2}(\frac{d_g c_0}{dt} + \frac{d_s c_0}{dt}) \tag{5}$$

6 Convergence of the Sky Model

Under the assumption that all nodes are correct, 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 $c_0 \geq 0.5$.** 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. 2a and Fig. 2b respectively. From Fig. 2a 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.,sky c_0 strictly increases with time t. This conclusion can also be proved mathematically, but it won't be presented here due to lack of space in this paper. From Fig. 2b 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 $D = 5$, c_0 can still converge to 1 in about 10 rounds.

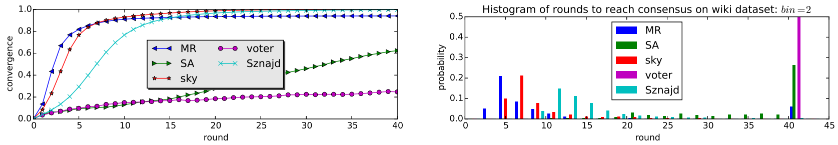
We simulate the sky model on the wiki dataset for 1000 runs starting with $c_0 = c_1 = 0.5$, where *convergence* is defined as $cvg = |c_0 - c_1|/(c_0 + c_1)$, note here the network may also agree at 1 instead of 0. Some of other existing opinion dynamic models besides the MR model can also be adapted to our framework,



(a) Derivative of c_0 on t (b) Evolution of c_0

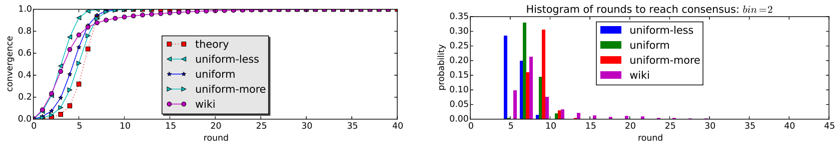
Fig. 2. Numeric analysis of the sky model

the *voter* model can be adapted by that for each node the opinion of a random selected node from all of its followees is chosen, and the *Sznajd* model can be adapted by that for each node if two randomly chosen followees have the same opinion, then the node set its opinion to that opinion otherwise nothing happens. The convergence and rounds to converge for all the models on the wiki dataset are show in Fig. 3 . Note **round 41 means the network failed to reach consensus within 40 rounds in that run**, and also each bin of the histogram is 2.



(a) Convergence (b) Rounds to reach consensus

Fig. 3. Simulation of all the models on the wiki dataset



(a) Convergence (b) Rounds to reach consensus

Fig. 4. Simulation of the sky model on different datasets

From Fig. 3b , we can see that for the sky model, probability of rounds needed to reach consensus decrease asymptotically when greater than 10, and all of the runs can reach consensus within 40 rounds. In contract, all of the other models have some runs can't reach consensus within 40 rounds. Some of runs of the MR model can never reach consensus, and simulation shows the network may stuck in a stable state where both the nodes with opinion of 0 and 1 exist, but they never change in later rounds. Majority of the runs of the SA model will not reach consensus in 40 rounds, simulations shows that the network may vastly change in each round without steady change direction of convergence, and escape in a tiny probability from the state to the track with convergence steadily increased in each round.

The sky model on all the datasets are show in Fig. 4 . From Fig. 4a , we can see that for the sky model on each dataset, simulation result of the sky model approximately fits theoretical analysis. Rounds to converge grows with nodes count(denoted as N), and approaches more closely to theoretical result when N is larger, that's because mean field equation works best when $N \to \infty$, thus $\forall N$ the theoretical result is in fact the theoretical lower bound. From Fig. 4b , we can see that for the sky model, all the runs on all datasets can reach consensus within 40 rounds. Most of the runs can reach consensus quickly in about 10 rounds.

7 Fault Tolerance of the Sky Model

7.1 Sybil Attack

Sybil attack resistance analysis is straightforward. See Fig. 5, where node marked by A is the current node deciding its new opinion, and A decide its opinions according to opinions broadcasted by its followees including correct nodes C and faulty nodes F while nodes S are Sybil nodes. Because of the difficulty for S to make A trust S which is actually **controlled by A rather than S**, there are no directed links from S to A, so Sybil nodes take no effect when A is deciding its new opinion. Collusion among F and S does not help the attack, because the contribution to the decision of A is still the same with F without S.

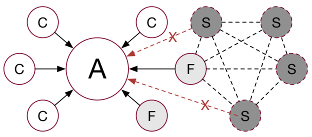


Fig. 5. Sybil attack

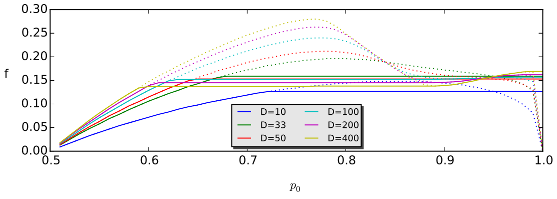


Fig. 6. Critical points

To compromise the network, creating new Sybil nodes or relationships between them are useless, instead, the adversary should attract more correct nodes to follow the nodes controlled by him. Experiments presented in later sections even show that for the same number of trust relationships from correct nodes to faulty nodes, the smaller the faulty nodes number is, the stronger the attack is.

7.2 Lower Bound

Because the model is symmetric on binary opinion 0 and 1, **it's sufficient to only track the case that $c_0 \geq c_1$.**

According to our definition of *almost-everywhere consensus*, a successful consensus process should fulfill the following requirements:

1. If c_0 is far greater than c_1(e.g., $c_0 \geq 2c_1$), then at least $(1-\epsilon)$ proportion of correct nodes should agree at 0.
2. Else at least $(1-\epsilon)$ proportion of correct nodes should agree at the same opinion which is either 0 or 1.

Under Byzantine failures, a faulty node can behave arbitrarily or even collude with other nodes. Different behavior of faulty nodes contributes differently to the evolution of c_0 in the mean field equations. Here are some scenarios:

1. All faulty nodes left the network at $t-dt$, then at t we have $a_0 = c_0/(1-f)$ and $a_1 = c_1/(1-f)$.
2. This scenario is not about failure, but about the dynamic characteristic of P2P network. Same number of correct nodes with opinion 0 joins the network at $t-dt$, then at then at t we have new c_0 denoted as c_0' with $c_0' = 2c_0/(1+c_0)$ together with the corresponding $c_1' = c_1/(1+c_0)$, $a_0' = 2c_0/(1+c_0)$ and $a_1' = c_1/(1+c_0)$.
3. All faulty nodes always broadcast 1 at $t-dt$, then at t we have $a_0 = c_0$ and $a_1 = c_1 + f$.
4. All faulty nodes broadcast 1 to half of their followees and 0 to the other half at $t-dt$, then at t we have $a_0 = c_0 + f/2$ and $a_1 = c_1 + f/2$
5. Faulty nodes broadcast opinion randomly chosen from 0 and 1 at $t-dt$, then at t we also have $a_0 = c_0 + f/2$ and $a_1 = c_1 + f/2$
6. Faulty nodes broadcast 1 when they should broadcast 0 and vise versa at $t-dt$, then at t we have $a_0 = c_0 + f_0'$ and $a_1 = c_1 + f_1'$, where f_0' and f_1' can be calculated according to the mean field equations similar to Eq. (2).

Note that the first two examples show how the agreement evolves in dynamic network, also topology or global view of the network are not involved in our model, and consensus emerges from local interactions of each node with its direct contacts. Failures can't be enumerated exhaustively and they can mix in a network, but since $\max(\frac{c_0+f_0}{1-f_s}) = \max(\frac{c_0+f_0}{c_0+c_1+f_0+f_1}) = c_0 + f$ when $f_0 = f$ and $f_1 = f_s = 0$, and $\min \frac{c_0+f_0}{1-f_s} = c_0$ when $f_0 = f_s = 0$ and $f_1 = f$, the following constraint always stands:

$$\begin{cases} a_0 \in [c_0, c_0 + f] \\ \quad a_0 + a_1 = 1 \end{cases} \tag{6}$$

Lemma 1 (If the network can tolerate any failures committed by given faulty nodes, it must agree at 0).

*For a network with c_0, c_1 and f given at time t to be $c_0(t)$, $c_1(t)$ and $f(t)$, if it can tolerate **any** failures committed by faulty nodes, then it must agree at 0.*

Proof. For the case that $c_0(t)$ that is far greater than $c_1(t)$, it stands according to the almost-everywhere consensus requirements stated above. For the else case, if some failures can stop it to agree at 0, then according to the requirements it must agree at 1, s.t. $\exists$ time $t' > t$ and $c_0(t') \leq \varepsilon(1-f)$. Because of the continuity of c_0 on t, must $\exists t''$, s.t $t' > t'' > t$, $c_0'' \in [c_0(t'), c_0(t)]$, $c_0(t'') = c_1(t) < c_0(t)$ and $c_1(t'') = 1 - f - c_0(t'') = c_0(t'')$. But according to the symmetric property of the model, the failures must also be able to stop it to agree at 1 from time t'', thus leads to contradiction.

Lemma 2 (For given f, greater c_0 tolerate failures equally or better).

For a network with given f, if at two times t' and t''(no relationship between t' and t'' assumed), s.t $c_0(t') < c_0(t'')$, and for $t > t'$, network can tolerate any failures, then it can also tolerate any failures for $t > t''$.

Proof. If for $t > t'$ and the network can tolerate any failures, then according to Lemma 1, it must agree at 0. We then discuss in two cases. For $c_0(t'') \leq \varepsilon(1-f)$, because of the continuity of c_0 on t, $\exists t'''$ s.t. $c_0(t''') = c_0(t'') \in [c_0(t'), \varepsilon(1-f)]$ and $c_1(t''') = 1 - f - c_0(t''') = c_1(t'')$, thus the network can tolerate any failures for $t > t'''$, we can then conclude the network can also tolerate any failures for $t > t''$. For $c_0(t'') > \varepsilon(1-f)$, if it can't reach consensus successfully, then must $\exists t''' > t''$ s.t $c_0(t''') \in [c_0(t'), \varepsilon(1-f)]$, but it's already known that for $t > t'$ s.t $c_0(t) \in [c_0(t'), \varepsilon(1-f)]$ it can tolerate any failures, thus leads to contradiction.

Lemma 3 (If tolerate smaller a_0, then tolerate greater a_0).

For a network with given f, c_0 and c_1, if at two times t' and t''(no relationship between t' and t'' assumed), s.t $a_0(t') < a_0(t'')$, and for $t > t'$, network can tolerate any failures, then it can also tolerate any failures for $t > t''$.

Proof. From Eq. (5) we can see that given other parameters, dc_0/dt strictly increases with a_0(note that $a_1 = 1 - a_0$), then $c_0(t' + dt) < c_0(t'' + dt)$. According to Lemma 2, it can also tolerate any failures for $t > t''$.

Theorem 1 (Lower bound of fault tolerance).

For any network with known faulty nodes and initial states of correct nodes, thus c_0, c_1 and f are given, if the network can tolerate the failure that all the faulty nodes always output 1, it can tolerate any other failures.

Proof. According to Lemma 3, and Eq. (6), if a network can tolerate failure with $a_0 = c_0$ together with $a_1 = c_1 + f$, then it can tolerate any other failures. And $a_0 = c_0$ together with $a_1 = c_1 + f$ is exactly the case all the faulty nodes always output 1, thus the theorem stands.

7.3 Fault Tolerance Performance

Because of the constraint that $c_0 + c_1 + f = 1$, it's not convenience to study the performance threshold of fault tolerance on C_0 directly. However, we can translated the threshold question to a new one: if at time t a network with $c_0 = p$ has no faulty nodes, then uniformly choose f proportion of all the nodes(including opinion with 0 and 1) to be faulty, what's the max value of f the network can tolerate?

$f_{critical}$ is the *critical point* for p if $f_{critical}$ fulfill the following two requirements:

1. $\forall f < f_{critical}$, when $t \to \infty$ and under any failures, $c_0/(1-f) \geq 1 - \varepsilon$.
2. $\nexists f'$ such that f' fulfill the previous requirement while $f' > f_{critical}$.

Following the definition of *critical point*, according to Theorem1 for $\varepsilon = 0.05$, by iterating on the mean field equation, critical points can be plotted in Fig. 6, where solid lines are critical points. There are also dotted lines where at each point $dc_0/d_t = 0$. From the figure we can see that $\forall D \in [10, 400]$, as long as $p \geq 0.75$, the network can tolerate any failure with $f \leq 0.13$.

8 Experiment

According to existing studies, latency between peers in DHT is mostly between 50 to 1000 ms [21]. 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.

Since for a network with c_0 far greater than c_1(e.g., $c_0 \geq 2c_1$), reaching consensus at 0 is successful, but that at 1 is failed, we define *signed convergence* as the Eq. (7), thus only if a network survive from failures, *signed convergence* will equal to *convergence* defined earlier.

$$cvg = (c_0 - c_1)/(c_0 + c_1) \tag{7}$$

To measure final decision of correct nodes, we also define *decision* as the following equation:

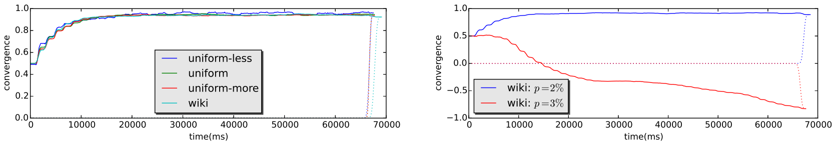
$$decision = |d_0 - d_1|/(d_0 + d_1) \tag{8}$$

where d_0 and d_1 is the count of correct nodes which have final decision on opinion 0 and 1 respectively.

We experiment on network started with $cvg = 0.5$ and $f = 13\%$ while faulty nodes always output 1, then in all decided correct nodes(agree at 0 or 1), for all dataset correct deciding(agree at 0) is about 96%, and uniform datasets have almost the same performance regardless their network scale, as shown in Fig. 7a.

But for the wiki datasets, we also concern the tolerance of failures by collusion of *top $n\%$ influential nodes*, defined as the first $n\%$ nodes by sorting all nodes in descendant order on the count of a node's followees. Simulation shows that for the target $\varepsilon < 5\%$, the algorithm can tolerate failures by 2% top nodes on the wiki dataset, as shown in Fig. 7b, where the red dotted lines are the case of failed to reach the goal under failures commited by 3% top nodes. In all decided correct nodes(agree at 0 or 1), correct deciding(agree at 0) is about 96.8%.

Comparing failures committed by random nodes and top influential nodes, we also find that **more centralized trust relationships leads to more powerful ability to compromise the network** even when the total numbers of trust relationships participated in the collusion are the same, and in theory analysis we have already known that the effect of a specific failure depends on the density of trust relationships for correct nodes to faulty nodes. For the wiki dataset, the total trust relationships is 33256, and for 13% random nodes, the trust relationships involved is about 4323, while for top 3% nodes, the trust relationships



(a) By random nodes (b) By top influential nodes

Fig. 7. Convergence under failures

involved is only 2155, in contrast that the network can survive in the former but no in the later. Even excluding the factor that lots of trust relationships are among the faulty nodes which has no effect for correct nodes in the 13% random node case, the result also supports the finding.

9 Future Work and Conclusion

Although our approach can successfully runs over the wiki dataset, it also shows the consensus speed degrades comparing to the uniform dataset, as existing studies show that community strength impacts the performance [22]. The relationships between our model and community strength need to be studied further. Fault tolerance performance will degenerate when starting with convergence ≤ 0.5 as we can see from Fig. 6. However, layered on this work, hash value consensus has been developed by us which can tolerate failures well in any case utilizing the premise that hash collision is impossible when hash size is big enough.

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 consensus.

References

1. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. `http://www.bitcoin.org/bitcoin.pdf` (2009)
2. Douceur, J.R.: The sybil attack. In: Revised Papers from the First International Workshop on Peer-to-Peer Systems. pp. 251–260. IPTPS '01, Springer-Verlag, London, UK, UK (2002)
3. Aspnes, J., Jackson, C., Krishnamurthy, A.: Exposing computationally-challenged Byzantine impostors. Tech. Rep. YALEU/DCS/TR-1332, Yale University Department of Computer Science (Jul 2005)
4. Courtois, N.T., Bahack, L.: On subversive miner strategies and block withholding attack in bitcoin digital currency. CoRR abs/1402.1718 (2014)

5. Castellano, C., Fortunato, S., Loreto, V.: Statistical physics of social dynamics. Rev. Mod. Phys. 81(2), 591–646 (2009)
6. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data (Jun 2014)
7. Castro, M., Druschel, P., Ganesh, A., Rowstron, A., Wallach, D.S.: Secure routing for structured peer-to-peer overlay networks. SIGOPS Oper. Syst. Rev. 36(SI), 299–314 (2002)
8. Bazzi, R.A., Konjevod, G.: On the establishment of distinct identities in overlay networks. In: Proceedings of the Twenty-fourth Annual ACM Symposium on Principles of Distributed Computing. pp. 312–320. PODC '05, ACM, New York, NY, USA (2005)
9. Borisov, N.: Computational puzzles as sybil defenses. In: Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing. pp. 171–176. P2P '06, IEEE Computer Society, Washington, DC, USA (2006)
10. Alvisi, L., Clement, A., Epasto, A., Lattanzi, S., Panconesi, A.: SoK: The evolution of sybil defense via social networks. In: Proceedings of the 2013 IEEE Symposium on Security and Privacy. pp. 382–396. SP '13, IEEE Computer Society, Washington, DC, USA (2013)
11. Cawrey, D.: Are 51% attacks a real threat to bitcoin? http://www.coindesk.com/51-attacks-real-threat-bitcoin/ (June 20 2014)
12. Schwartz, D., Youngs, N., Britto, A.: The Ripple protocol consensus algorithm. https://ripple.com/files/ripple_consensus_whitepaper.pdf (2014)
13. Kim, J.: Safety, liveness and fault tolerance—the consensus choices stellar. https://www.stellar.org/blog/safety_liveness_and_fault_tolerance_consensus_choice/ (2014)
14. Poelstra, A.: A treatise on altcoins. https://download.wpsoftware.net/bitcoin/alts.pdf (10 2014)
15. Singh, A., Castro, M., Druschel, P., Rowstron, A.: Defending Against Eclipse Attacks on Overlay Networks. In: Proceedings of the 11th Workshop on ACM SIGOPS European Workshop. EW 11, ACM, New York, NY, USA (2004)
16. Dwork, C., Peleg, D., Pippenger, N., Upfal, E.: Fault Tolerance in Networks of Bounded Degree. In: Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing. pp. 370–379. STOC '86, ACM, New York, NY, USA (1986)
17. Augustine, J., Pandurangan, G., Robinson, P.: Fast Byzantine Agreement in Dynamic Networks. In: Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing. pp. 74–83. PODC '13, ACM, New York, NY, USA (2013)
18. Fischer, M.J., Lynch, N.A., Paterson, M.S.: Impossibility of distributed consensus with one faulty process. J. ACM 32(2), 374–382 (1985)
19. Chandra, T.D., Toueg, S.: Unreliable failure detectors for reliable distributed systems. J. ACM 43(2), 225–267 (1996)
20. Mean field theory. http://en.wikipedia.org/wiki/Mean_field_theory

21. Dabek, F., Li, J., Sit, E., Robertson, J., Kaashoek, M.F., Morris, R.: 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. pp. 7–7. NSDI'04, USENIX Association, Berkeley, CA, USA (2004)
22. Gargiulo, F., Huet, S.: Opinion dynamics in a group-based society. EPL (Europhysics Letters) 91(5), 58004 (Sep 2010)