Towards Truthful Feedback in P2P Data Structures

Erik Buchmann, Klemens Böhm, Christian von der Weth

Universität Karlsruhe (TH), Germany
{buchmann|boehm|weth}@ipd.uni-karlsruhe.de

Abstract. Peer-to-Peer data structures (P2P data structures) let a large number of anonymous peers share the data-management workload. A common assumption behind such systems is that peers behave cooperatively. But as with many distributed systems where participation is voluntary, and the participants are not clearly observable, unreliable behavior is the dominant strategy. This calls for reputation systems that help peers choose reliable peers to interact with. However, if peers exchange feedback on experiences with other peers, spoof feedback becomes possible, compromising the reputation system. In this paper we propose and evaluate measures against spoof feedback in P2P data structures. While others have investigated mechanisms for truthtelling recently, we are not aware of any studies in P2P environments. The problem is more difficult in our context because detecting unreliable peers is more difficult as well. On the other hand, a peer can observe the utility of feedback obtained from other peers, and our approach takes advantage of this. To assess the effectiveness of our approach, we have conducted extensive analytical and experimental evaluations. As a result, truthful feedback tends to have a much higher weight than spoof feedback, and collaboration attacks are difficult to carry out under our approach.

1 Introduction

Peer-to-Peer systems (P2P systems) are distributed systems consisting of many nodes in open, coordinator-free communities. Peers typically are known by pseudonyms, which they can replace at little or no cost. P2P systems do not have a central instance that could observe the behavior of peers. Thus, reputation systems [1] to identify and penalize misbehaving peers are crucial building blocks of all kinds of P2P systems.

Reputation systems assign each peer a reputation value, be it positive or negative. A reputation value is an aggregate of positive feedback or complaints from other participants that have observed the behavior of the peer in the past. Clearly, we cannot expect that nodes issue only truthful feedback. A peer may wish to discredit others which have complained about it, or attackers could try to harm nodes by issuing spoof feedback. For instance, [2] has observed similar behavioral patterns at eBay. However, while others have investigated mechanisms for truthtelling recently [3–5], we are not aware of any studies in P2P environments.

This paper proposes and evaluates measures for truthful feedback for one particular kind of P2P system, namely P2P data structures (a.k.a. P2P overlay networks, distributed hash tables, etc. [6]). Such structures let a large number of peers share the data-management and query-processing workload. Designing mechanisms against spoof feedback in P2P data structures is challenging, more than for other P2P systems.

To lookup a data object, several peers must cooperate, and the lookup fails if only one of them is not reliable. With other P2P systems in turn, there typically is only one peer that carries out a service or a well-defined part of it. A related issue is that it is difficult to identify the defector if a lookup request is not processed properly. Consequently, generating truthful feedback is more difficult as well. Further, a lookup in P2P data structures consists of operations that are relatively simple. This means that reputation management must be relatively simple as well so that it does not become disproportionately expensive. Another issue is that P2P data structures have good scalability characteristics, and reputation management must not get in the way of this. In addition to these complications specific to P2P data structures, there are further ’more general’ issues: Each node may change its behavior at any time and can behave differently with different peers. Thus, negative feedback on cooperative nodes and positive feedback on unreliable nodes typically exists. We cannot readily distinguish it from spoof feedback.

This paper makes the following contributions: First, we describe the particular requirements that an approach against spoof feedback in reputation systems for P2P data structures must fulfill. We then describe our approach. It is a characteristic of P2P data structures that a peer can observe the utility of feedback obtained from other peers, and our approach takes advantage of this. For instance, a peer which has forwarded a query to a certain node and has obtained a proper query result may conclude that complaints about that node were wrong and positive feedback was correct. With our approach, each peer uses such clues to derive weighting factors both regarding the issuers of feedback items and the peer the feedback refers to. Second, we provide an evaluation of our approach, both with an analytic model and by means of experiments. The analysis confirms that the differentiation between useful and less useful feedback is effective. We point out the analysis is rather general, i.e., leaves aside the details of the particular underlying reputation system. The experiments address issues which are difficult to examine analytically, such as collusion attacks and dynamicity issues. The experimental results are positive as well. For instance, our approach is effective against collusion attacks in realistic settings. Third, the article features a discussion of the applicability of our measures against spoof feedback to other reputation systems and application scenarios.

The remainder of this article is organized as follows: Section 2 describes the technical background, followed by a description of our approach in Section 3. Section 4 will analyze the approach, Section 5 will evaluate it. Section 6 reviews related work. Finally, Section 7 provides a discussion of the applicability of our approach, and Section 8 concludes the paper.

2 Background

This section briefly reviews the characteristics of P2P data structures and reputation systems and provides a short description of a reputation system which we will use as a basis for our experiments. The section also quantifies the damage spoof feedback may cause in P2P data structures without any countermeasures.

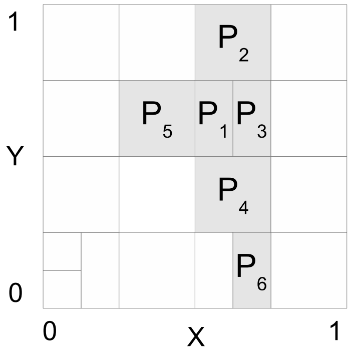

Fig. 1. Two-dimensional CAN.

Content-Addressable Networks P2P data structures (a.k.a. P2P overlay, distributed hashtable) administer huge sets of $(key, value)$-pairs on top of a large physical network. *Content-Addressable Networks (CAN)* [7] are a prominent variant of P2P data structures. Other instances are P-Grid [8], Viceroy [9] or Chord [10] which differ primarily with regard to contact selection and routing topology; cf. Section 6. We point out that the presented measures are independent from the specific P2P data structure. However, when presenting our results, we use a CAN for sample calculations and experiments.

A CAN is a distributed system that consists of many nodes (peers). Each peer can issue queries for any data object stored in the CAN, but it is supposed to store data and participate in the evaluation of queries as well. Each CAN node is responsible for a certain zone of the key space, and it knows all neighbors, i.e., peers responsible for adjacent zones of the key space. The key space is an n-dimensional torus of Cartesian coordinates in the unit space. It is independent from the underlying physical network topology. The assignment of zones of the key space to peers results from the CAN construction protocol. A peer which wants to join the CAN finds a random node that is already in the CAN. That node splits its zone, keeping one half and reassigning the other half to the new node.

The key space of the CAN in Figure 1 is two-dimensional. Node P_1 is responsible for Zone ([0.5; 0.5), [0.625; 0.75)) of the key space, i.e., it knows all $(key, value)$-pairs where $key \in$ ([0.5; 0.5), [0.625; 0.75)). The neighbors in the contact list of Node P_1 are Nodes P_2, P_3, P_4, P_5. Since the key space is mapped on a torus, Node P_2 is a neighbor of Node P_6.

Every data object maps to a point in the key space. Accordingly, each operation $(query, insert, update, delete)$ in the CAN refers to a point in the key space. For example, a query is the key of a particular (key, value)-pair, and its result is the value of the pair. Query processing in CAN is a variant of *greedy forward routing*. A node that has issued a query first checks if it can answer the query from its zone. Otherwise, it forwards the query to the neighbor in its contact list whose distance to the query key is minimal. The procedure recurs until the query arrives at the peer that can answer it. The peer then sends the result back to the issuer. In a d-dimensional CAN with N peers, a number of $l = d/4 \cdot N^{1/d}$ participate in the processing of a lookup on average.

Incentives Mechanisms for Cooperation in Structured P2P Systems Research on P2P data structures has tacitly assumed that peers follow the protocol. But participation actually is voluntary, and uncooperative behavior is the dominant strategy (in the economic sense of the word). An uncooperative peer does not follow the protocol of the P2P data structure, e.g., it drops incoming messages. In comparison to P2P systems based on flooding, e.g., Kazaa[1] or gnutella[2], P2P data structures are more vulnerable to uncooperative behavior. Few uncooperative peers can significantly reduce reliability of a P2P data structure. For example, in a CAN consisting of $N = 10,000$ peers with $d = 4$ dimensions, $l = d/4 \cdot N^{1/d} = 10$ peers on average forward a query (cf. [7]). Now suppose that the CAN contains $u = 500$ peers which do not forward any incoming query message. Then the probability to obtain a query result is only $(1 - u/N)^l \approx 60\%$.

FairNet [11] is our proposal for a reputation system that renders uncooperative behavior unattractive. Peers in FairNet, the *feedback issuers*, generate and distribute *feedback items*. Such items are the observations of the feedback issuers regarding a particular transaction. Thus, a feedback item consists of a positive or negative statement and contains the identifiers of the issuer and the peer the feedback refers to (the *feedback subject*). Only peers with a number of positive feedback items above a threshold value are allowed to participate in the P2P data structure. Peers can obtain positive feedback in short time by carrying out proofs of work [12, 13]. A proof of work is a problem that is easy to formulate, and the solution is easy to verify, but solving it requires a lot of resources.

Each peer maintains a private local *reputation repository* for feedback on its neighbors. The repository has a capacity of s feedback items per subject. If an item is added to a repository that is already full, one item in the repository will be replaced. Consequently, as soon as a peer starts to behave unreliably, negative feedback items tend to replace positive ones. Based on the feedback items in its local repository, each peer can derive an individual *reputation value* for each feedback subject. Peers do share feedback: A peer that has observed cooperative behavior of another peer generates a feedback item and stores it in its local repository. The next time the peer sends out a message, it attaches recent feedback items whose subject is a neighbor of the recipient of the message. Therefore the repositories do not only contain feedback generated by the maintainer of the repository, but also feedback forwarded by adjacent peers. In the following, we refer to such intermediate peers as *forwarders*.

Note that the ratio of positive and negative feedback items in a repository on an uncooperative peer does not exactly follow its failure probability. For example, if a peer does not handle 50% of all incoming messages, it does not obtain 50% positive and 50% negative feedback as well. One reason for this is that not only this peer, but also any other peer later in the sequence of forwarders can drop the query. This results in negative feedback on all peers in the sequence.

[1] http://www.kazaa.com

[2] http://www.gnutella.com

The Impact of Spoof Feedback We know from previous work [11] that the measures outlined in the previous subsection are effective even against peers which process messages properly at a variable rate. For example, a peer which does not work off 10% of all incoming queries ends up with more than the double effort compared to a peer that handles all queries properly. (The additional effort is the result of a higher number of proofs of work, in order to remain in the CAN.) However, the experiments show also that dishonest peers issuing spoof feedback can impair the effectiveness of the reputation system significantly.

Suppose that a peer maintains a repository of size s that consists of feedback on only one peer. If feedback is truthful, p_{pos} is the probability that an arbitrary feedback item in the repository is positive. The peer assigned to the repository is deemed reliable if the repository contains at least t positive feedback items, i.e., if $s \cdot p_{pos} \geq t$. Any feedback item has been issued by one of a peers. Now we wonder: How many dishonest peers are necessary to affect the reputation of one peer? Let $x \leq a$ be a number of dishonest peers issuing spoof positive feedback. The overall probability for positive feedback now changes to $\hat{p}_{pos} = \frac{a-x}{a} \cdot p_{pos} + \frac{x}{a} \cdot 1$. Equating this with $s \cdot \hat{p}_{pos} \geq t$, we obtain an estimate of the rate of $\frac{x}{a}$ dishonest peers required to induce positive feedback into the repository such that an uncooperative peer is above the threshold:

$$\frac{x}{a} \geq \frac{\frac{t}{s} - p_{pos}}{1 - p_{pos}} \tag{1}$$

For example, consider a FairNet instance with a repository size of $s = 10$ and a threshold $t = 6$. In this setup each uncooperative peer does not forward or answer 50% of all incoming messages. We know from previous work [11] that this results in a probability of $p_{pos} \approx 0.13$ for uncooperative peers. Equation 1 now tells us that an uncooperative peer is deemed cooperative if at least 54% of the peers it has interacted with issue spoof positive feedback.

3 Measures Against Spoof Feedback

This section motivates the requirements that an approach against spoof feedback in reputation systems for P2P data structures must fulfill. The section also describes our approach with its measures and data structures. We stress that the presented approach does not depend on particular implementations of reputation systems or P2P data structures. Instead, the peers just need to know the nodes which forwarded the feedback, the feedback subjects and the correlation between the feedback and the transaction outcomes; no matter how the implementation handle this. Section 7 provides a discussion on the applicability of our approach.

Approaches Against Spoof Feedback – Requirements Measures to detect and avoid spoof feedback in a reputation system for P2P data structures must meet the following requirements:

Effectiveness Obviously, the most urgent issue is the effectiveness of the detection of spoof feedback. Effectiveness means that it does not pay off to issue spoof feedback in any case. There are two worst-case scenarios that might impair the effectiveness: First, there are situations where the distinction between spoof feedback and truthful feedback is not feasible. If a transaction fails with a probability of 50%, any feedback is correct with a probability of 50% as well. Second, peers may run collusion attacks to feed spoof feedback into the repositories of others. When such an attack takes place, it can be the case that the majority of the peers displays dishonest behavior.

Short response times The time required to adapt to new situations is an important criterion in any P2P system where peers can change their behavior at any time. Peers can gain advantages during the period of time required to detect such changes. This period of time needs to be as small as possible.

Filtering 'wrong' feedback There are several reasons why honest peers can sometimes generate wrong feedback. For instance, a cooperative peer may forward a message to an uncooperative neighbor only once, because it does not know any better as yet. If the peer obtains negative feedback but does not forward to the uncooperative neighbor again, the feedback could be seen as spoof. Thus, our approach should differentiate between spoof feedback and feedback that is wrong in spite of best intentions.

Tamper-resistant design The measures must not introduce new 'holes' which dishonest and/or unreliable peers can exploit. Therefore the measures should rely on local operations as much as possible, in contrast to other peers, which could be dishonest.

Preserving trust relationships that already exist The idea that a peer either generates spoof feedback all the time or not at all is too undifferentiated. For instance, a peer can generate spoof feedback on selected neighbors only, or wrong feedback could be the result of successful attacks. Thus, we strive for an approach that does not break existing trust relationships after having observed wrong feedback items from one forwarder. Instead, it differentiates between useful and spoof feedback from the same forwarder.

Low resource consumption P2P data structures aim to process large numbers of small transactions. It is acceptable that a few transactions get lost due to unreliable peers. On the other hand, a measure against spoof feedback must not slow down the processing of transactions due to excessive resource consumption.

Overview With our approach, each peer individually determines the weight of the feedback. In particular, a peer can assign different weights to each combination of subject and forwarder. The weights depend on the differences or similarities between the transaction outcomes observed and the outcomes predicted by the feedback. In P2P data structures, the feedback is used to identify a reliable peer to forward a query to. Here, the weights ensure that messages go to reliable peers only, even in the presence of dishonest peers issuing positive spoof feedback.

We now explain briefly the rationale behind our design decisions. A peer needs to associate feedback items with the forwarder they have come from. The assignment helps the peer to reduce the impact of spoof feedback and to determine the weight of future feedback coming from that peer. At first sight, we could have associated feedback with the issuer instead of the forwarder. Namely, the issuer is responsible for the feedback it has generated. However, the forwarder is able to manipulate incoming feedback items, and it can decide which feedback is forwarded and which one is not, i.e., apply some kind of censorship. In other words, the receiver of a feedback item can only pin down the last forwarder of the item with certainty, but not the issuer. Further, one might ask why there are separate weights for each forwarder and each feedback subject. This is because a peer which forwards useful feedback on one feedback subject might forward spoof feedback on another one.

Data Structures We now specify the data structures required to implement our approach against spoof feedback on top of an existing reputation system. Our approach introduces two variables individually maintained by each peer, *weighting factors* and *transaction logs*. The log is the history of all recent transactions handled by the peer, i.e., it contains the identifier of a transaction and the peer the query was forwarded to. A peer also maintains a weighting factor $w_{\sigma,\phi}$ in the interval $[0;1]$ for each feedback subject σ and forwarder ϕ.

In addition, two data structures implement the reputation system as described in Section 2, namely *feedback items* and *reputation repositories*. Assuming the presence of such data structures does not restrict the applicability of our approach: [14] indicates that the referred structures are common for most of the P2P reputation systems.

How to Weight Reputation Values When a peer wants to compute the reputation value of a particular node, it first calculates several auxiliary reputation values, based on the feedback from the different forwarders. It then aggregates these auxiliary values using the weighted average. Let P_σ denote the set of all peers that have forwarded feedback for subject σ, and let $r(\sigma, i)$ be a function that computes the auxiliary reputation value for peer σ based on feedback from peer i.[3] The reputation value v then is as follows:

$$v_\sigma = \frac{\sum\limits_{i \in P_\sigma} r(\sigma, i) \cdot w_{\sigma,i}}{\sum\limits_{i \in P_\sigma} w_{\sigma,i}} \tag{2}$$

Equation 2 ensures that feedback with a low weight does not affect the reputation value significantly. Thus, in P2P data structures the messages go to reliable peers only, even in the presence of dishonest peers issuing positive spoof feedback.

Updating the Weighting Factors Having observed the outcome of a transaction, each node can determine the utility of the feedback available to it. For example, negative feedback on a node that has handled the transaction properly has been less informative, therefore the weight assigned to the corresponding (forwarder, subject)-pair shall be decreased.

P2P data structures are dynamic systems where the peers are free to change their behavior at any time. Thus, there should be weights that allow to focus on recent transactions. In addition, single stochastic occurrences should not impact the weights. This –

[3] In FairNet, the reputation value is the number of positive feedback items in the repository that refers to the peer in question.

and the fact that it does without additional data structures – motivates the use of the exponential moving average over time to adapt the weights to new observations. Factor z with $0 \leq z \leq 1$ specifies the importance of recent information, i.e., larger values of z prefer new values. Let $a(F_{\sigma,\phi}, \theta)$ be a function to express the correlation between the transaction result θ and the set of feedback items $F_{\sigma,\phi}$ with Subject σ forwarded from Peer ϕ. The new weight w' is derived from the old weight w as shown in Equation 3.

$$w'_{\sigma,\phi} = (1 - z) \cdot w_{\sigma,\phi} + z \cdot a(F_{\sigma,\phi}, \theta) \tag{3}$$

In FairNet the transaction results and the feedback items are binary: a query is either answered or not, and the number of positive feedback items about a particular peer can only be above the threshold t or below. Let T^{pos}, T^{neg} be the sets of all successful and unsuccessful transactions, respectively. We now can develop the following correlation function $a(F_{\sigma,\phi}, \theta)$:

$$a(F_{\sigma,\phi}, \theta) = \begin{cases} 1 & \text{if } (\theta \in T^{pos} \wedge |F_{\sigma,\phi}| \geq t) \vee (\theta \in T^{neg} \wedge |F_{\sigma,\phi}| < t) \\ 0 & \text{if } (\theta \in T^{pos} \wedge |F_{\sigma,\phi}| < t) \vee (\theta \in T^{neg} \wedge |F_{\sigma,\phi}| \geq t) \end{cases} \tag{4}$$

Other reputation systems may depend on measures that express more sophisticated correlations between the transaction outcomes observed and the feedback. However, our experiments in Section 5 will show that a relatively simple solution leads to remarkably positive results already.

4 Analysis

This section provides an analysis of the measures proposed. The analysis is independent from the underlying reputation system and data structures. On the other hand, the analysis (not the experimental evaluation) is based on various assumptions. We will discuss the impact of these assumptions later in the paper. First, transaction processing takes place in rounds. In every round, each node issues one query and forwards or answers l queries on average. In addition, we assume that the system is in steady state, and the load of query processing and message forwarding is equally distributed among all nodes. Our formal analysis further assumes that uncooperative and dishonest peers are evenly distributed over the key space, i.e., there is not any cluster of neighboring peers that are unreliable and/or dishonest. Finally, we assume that the underlying reputation system handles the creation and distribution of feedback as follows: At the end of a transaction, each forwarder will be informed about its outcome. If a query remains unanswered, each forwarder generates negative feedback with the next forwarder in the sequence as feedback subject. In the other case, these peers generate positive feedback. The generated feedback will then be forwarded to all neighbors of the feedback subject.

The analysis only refers to the measures against untruthful feedback, not to the P2P data structure and the reputation system together with these measures. Hence, the analysis uses the quality of the feedback available, the frequency of successful transactions in the P2P data structure etc. as external parameters. In particular, the characteristics of P2P data structures are represented by two values: A query will not be processed successfully with probability g, and the processing of each query requires the cooperation

Parameters of the data structure	*Symbol*
Probability of an unsuccessful transaction	g
Number of peers that have to cooperate to process one transaction	l

Parameters of the reputation system	*Symbol*
Probability of positive feedback	p_{pos}
Number of feedback items in the repository of one peer	s
Threshold for the number of positive feedback items for a reliable peer	t

Parameters of the countermeasure	*Symbol*
Ratio of spoof feedback provided by dishonest peers	b
Smoothing factor of the Exponential Moving Average	z

Table 1. Parameters used in the analysis

of l peers that are not observable from the outside. In order to model the reputation system we use the following parameters: A peer forwards and handles transactions of another one only if it has at least t positive feedback items in its repository whose subject is the peer in question. The repository has the capacity to store s feedback items per subject. As a result of feedback generation in the reputation system, p_{pos} is defined to be the probability that an arbitrary feedback item issued by a honest peer on a reliable subject is positive. The values of p_{pos} then depend on the reputation system.[4]

The rate of spoof feedback b in the reputation system is the input variable of our analysis. A value of $b = 0$ denotes an honest peer that disseminates truthful feedback only, while dishonest peers forward spoof feedback at a rate of $b > 0$. Finally, the factor z specifies the smoothness factor of the Exponential Moving Average and can be customized according to the preference for newer values. Table 1 lists all parameters used in the analysis.

To examine the impact of spoof feedback, we first determine the expected average values of the weights. A dishonest peer issues spoof feedback with a rate of b and accurate feedback with a rate of $(1-b)$ that is positive with probability p_{pos}. Equation 5 gives the probability that an arbitrary feedback item generated by a dishonest node is positive.

$$p_{pos}^{dis} = \begin{cases} b \cdot 1 + (1-b) \cdot p_{pos} & \text{for spoofed positive feedback} \\ b \cdot 0 + (1-b) \cdot p_{pos} & \text{for spoofed negative feedback} \end{cases} \tag{5}$$

If at least t feedback items in a repository with s items are positive, the maintainer of the repository deems the peer reliable. Each honest peer (characterized by $b = 0$)

[4] See [11] where the value of p_{pos} is derived in one specific reputation system.

generates positive feedback with probability p_{pos}. Therefore, it happens with a certain probability p_t that an honest peer generates less than t positive feedback items on a reliable node. In consequence, other nodes could suspect the peer to disseminate spoof feedback and reduce the weight of its feedback. The share of positive feedback items in a repository follows a binomial distribution. Equation 6 now calculates the probability p_t for a repository containing less than t positive feedback items:

$$\begin{aligned} p_t &= P(\text{Number of positive feedback in the repository} < t) \\ &= \sum_{i=0}^{t-1} \tbinom{s}{i} \cdot (p_{pos})^i \cdot (1 - p_{pos})^{s-i} \end{aligned} \tag{6}$$

In order to determine the same probability for dishonest peers, we change the value in Equation 6 from p_{pos} to p_{pos}^{dis}, as shown in Equation 7:

$$p_t^{dis} = \sum_{i=0}^{t-1} \binom{s}{i} \cdot (p_{pos}^{dis})^i \cdot (1 - p_{pos}^{dis})^{s-i} \tag{7}$$

The expected average weight now is the probability that a transaction is not successful and that a repository contains less than t positive feedback items plus the probability of the opposite case. Equation 8 determines the weight of feedback issued by honest peers, Equation 9 does so for dishonest ones.

$$w^{hon} = p_t \cdot g + (1 - p_t) \cdot (1 - g) \tag{8}$$

$$w^{dis} = p_t^{dis} \cdot g + (1 - p_t^{dis}) \cdot (1 - g) \tag{9}$$

A participant in a P2P system is free to change its behavior at any time and with any frequency. For example, one peer might work hard to obtain a high standing in the eyes of others and try to disseminate spoof feedback afterwards. Thus, the time needed to adapt to new behavior is crucial. This time can be quantified as the number k of repository updates needed to adapt w to a new ratio of spoof feedback b. The weights are updated according to Equation 3. Therefore, k is a function of the smoothing factor z of the exponential moving average. Let a_k be the correlation $a(F_{\sigma,\phi}, \theta)$ between the transaction result and the set of feedback items at time k. We can now rewrite Equation 3 to Equation 10.

$$\begin{aligned} w_k &= z \cdot a_k + z \cdot (1-z)^1 \cdot a_{k-1} + z \cdot (1-z)^2 \cdot a_{k-2} + \cdots + (1-z)^k \cdot a_0 \\ &= z \cdot a_k + a_0 \cdot (1-z)^k + z \cdot \sum_{i=1}^{k-1} a_{k-i} \cdot (1-z)^i \end{aligned} \tag{10}$$

The initial parameter a_0 is the value of w before the change in the behavior. To ease the calculation, the number of positive and negative feedback items in the repository and the rate of unsuccessful transactions after the change is assumed as constant, i.e., $a_1, a_2, \cdots, a_k$ are equal. Now Equation 10 is a geometric sequence and can be solved and rewritten to obtain the value of k, as shown in Equation 11.

$$k = \left\lceil \frac{\log(\frac{w_k - a_k}{a_0 - a_k})}{\log(1-z)} \right\rceil \tag{11}$$

5 Evaluation

Having described the fundamentals of our approach independent from a concrete implementation of a P2P data structure or a reputation system, we will now evaluate the effectiveness with numeric methods and by means of experiments. Our intention with this section is to confirm that truthful feedback tends to have higher weight than spoof feedback even in worst-case settings, that the reputation system adapts quickly to changes in the behavior of nodes, and that it is effective against collaboration attacks. In order to obtain expressive results, the evaluation is based on FairNet.

Weights for Truthful and Spoof Feedback The first question is whether spoof feedback may obtain a higher weight than honest feedback. We accomplish this by interpreting the formulae of the analysis. To do so, we use realistic values taken from a FairNet instance consisting of $10,000$ peers organized in a four-dimensional keyspace. In this setup, u is the rate of uncooperative peers. Each uncooperative peer does not forward or answer 50% of all incoming query. The rationale behind a failure rate of 50% is to analyze a 'more difficult' setting – even in the presence of spoof feedback, a completely uncooperative peer would be quickly discovered. In contrast, a failure rate of 50% is a worst-case scenario for settings with a small number of uncooperative peers u, because *any* feedback item is wrong with probability 50%. In this setting the probability for each uncooperative node to obtain positive feedback is $p_{pos} \approx 0.13$.[5] I.e., it is less than its failure probability of 50%. This is because FairNet generates more negative than positive feedback.

In settings with a large fraction u of uncooperative peers, the number of truthful positive feedback in the repositories goes against zero, as do the probabilities of successful transactions. Thus, it is easy to detect spoof positive feedback. On the other hand, if the probability of a successful transaction is about 50%, spoof feedback cannot be detected. Therefore, in a setting with few queries properly processed and a small p_{pos}, we expect the weighting factors to be smaller on spoof feedback, compared to a setting where p_{pos} is smaller than the probability of successful queries. However, we can already declare success if the weights of spoof feedback *never* are above the ones of nodes following the protocol.

We now determine the weight of a dishonest peer for a ratio of $b = 0$ to $b = 1$ spoof positive feedback generated on an uncooperative feedback subject. Figure 2 graphs the result of our analysis. The figure confirms our expectations: For values such as $u = 10\%$ or below, i.e., in our worst-case scenario with 50% successful transactions, the weight of honest feedback ($b = 0$) is only slightly larger than the one of a peer issuing spoof positive feedback only ($b = 1$). In contrast, in settings with a large number of uncooperative peers and a high rate of unsuccessful transactions, i.e., with high certainty regarding the accuracy of feedback, the weight of honest feedback is significantly larger than the weight of spoof feedback. Summing up, the analysis so far has shown that our approach assigns higher weights to honest feedback in any case. However, the difference between truthful feedback and spoof feedback might be small in worst-case settings. In

[5] Determining the probability p_{pos} in FairNet requires a complex algebraic model which we omit here for the lack of space. See [11].

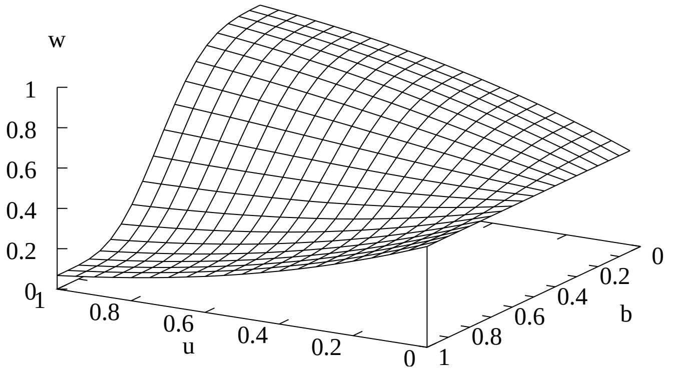

Fig. 2. Average weights for different shares of dishonest peers and spoof feedback.

these settings however, truthful and spoof feedback would lead to the same decisions. We will address the applicability of our measures under worst-case conditions by means of realistic experiments later on.

Dynamicity As a next step, we want to determine the number of updates k needed to adapt a weight w to a new ratio of spoof feedback b. The setup of our simulation is similar to the one used for Figure 2, i.e., the system consists of 10,000 peers in a four-dimensional topology. In order to have expressive results, the setup contains 50 uncooperative peers which do not handle 80% of incoming transactions.

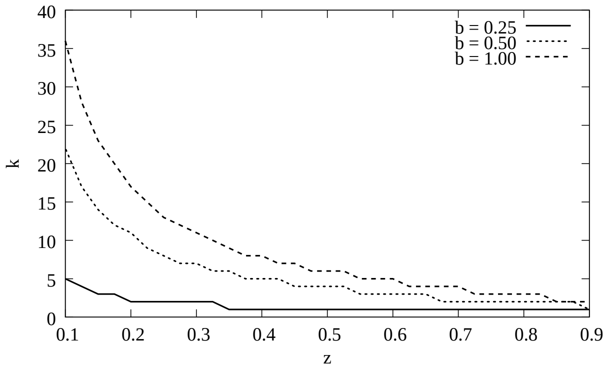

Fig. 3. Number of interactions to update the weight depending on the smoothing factor z.

Equation 9 provides an estimate of w, depending on the ratio of spoof feedback b. Figure 3 graphs the number of updates k needed to decrease the weight of a truthful repository ($b = 0$) to 99% of the weight of a dishonest repository.[6] The number of updates is shown in comparison to the smoothing factor z and for three different ratios of spoof feedback. The exponential moving average replaces old information at a constant rate z. This explains why it requires more updates k to adapt to a repository with $b = 1$ in comparison to one with $b = 0.25$, as shown in Figure 3. However, in P2P data structures a peer usually interacts with its neighbors frequently. The example calculation of Section 2 has shown that it requires around 10 interactions between neighboring peers to forward one query from the issuer to the peer that can actually answer it in a setting with 10,000 peers and a four-dimensional key space. Thus, even though the value of $k = 35$ at $z = 0.1$ and $b = 1$ might seem to be large, it actually tells us that the weights are adjusted within less than four rounds. Larger smoothing factors shorten this period of time even more.

Robustness Against Collaboration Attacks The last question that we have to address is: How useful is our approach in the presence of peers running a collaboration attack? In particular, does it pay off for a group of dishonest peers to 'boost' the reputation of an uncooperative peer by issuing and forwarding spoof feedback? A series of experiments addresses these questions. Our experimental setup consists of 1,000 peers in a setup where each peer has 26 neighbors. 50 uncooperative peers ignore 50% of the incoming queries. x dishonest peers surround each uncooperative peer. These x peers try to push the standing of the uncooperative one by disseminating spoof positive feedback items at a rate b. In other words, they generate honest feedback at a rate $(1 - b)$. In a series of 625 experiments, we varied the number of attackers from $x = 0$ to 25 and changed the ratio of spoof feedback from $b = 0$ to 1. As outlined in Section 2, only the neighbors can observe the behavior and generate feedback on a peer. Thus, the experiments with $x = 25$ identify extreme settings where dishonest nodes almost completely surround the uncooperative peer.

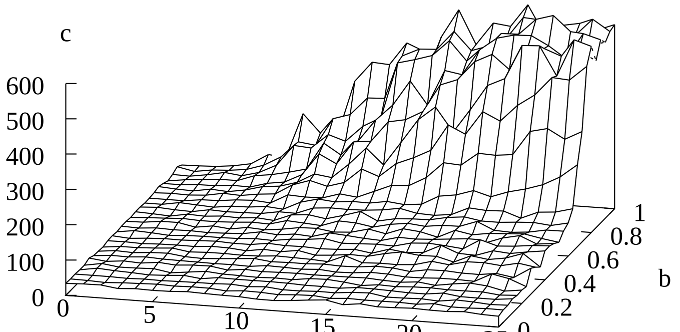

Fig. 4. Unhandled transactions in FairNet.

[6] Because of the exponential moving average, the weights asymptotically converge to the expected value. Hence, we are satisfied with a conformance of at least 99%.

Each experiment consists of 200,000 queries. The numbers are taken after an initialization period that allows the reputation system to reach a steady state. We measured the number c of queries dropped per round by all uncooperative peers, i.e., the total number of unanswered queries caused by 50 uncooperative peers, 'supported' by up to 950 dishonest peers issuing spoof positive feedback. Figure 4 shows the results of our experiments without our measures. It indicates that uncooperative peers drop a small fraction of queries even without the involvement of any dishonest peers ($x = 0$). This is because our experimental setup does not include data replication, and queries referring to keys in the zones of uncooperative peers are answered with a probability of 50% only. Except for this phenomenon, the reputation system does well without countermeasures against spoof feedback, even in the presence of dishonest peers. Only collaboration attacks where more than one third of the neighbors of an uncooperative node issue significantly more than 70% spoof feedback increase the number of unanswered queries. On the other hand, there have already been distributed attacks on the Internet with thousands of 'zombie computers' compromised by viruses and directed by a single attacker. Similar attacks on P2P data structures are conceivable as well. Thus, measures against spoof feedback are still necessary.

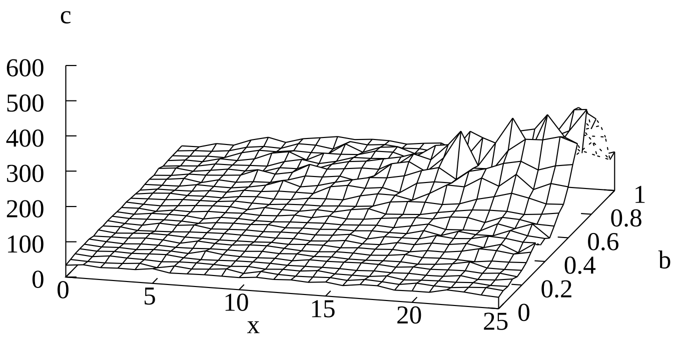

Fig. 5. Unhandled transactions in FairNet with weights.

But how do the experimental results change with our approach? To investigate this, we replayed our series of experiments with our measures activated. Figure 5 contains the results. The figure shows two findings: The impact of the attacks has been largely reduced, and the number of collaborators required for a successful attack has increased considerably. At least 17 attackers have to disseminate more than 80% spoof positive feedback to increase the number of unanswered queries significantly. Moreover, the maximal number of messages lost resulting from $x = 25$ absolutely dishonest collaborators is approximately one third of the one observed in the experimental setup without our approach. Other experiments with spoof negative feedback on cooperative feedback subjects (omitted here for lack of space) yield similar results. Summing up, repository weights are an effective countermeasure against collaboration attacks in reputation systems for P2P data structures.

6 Related Work

This section reviews approaches that are related to our measure against spoof feedback in reputation systems for P2P data structures. The section starts with a (very) short outline of related P2P data structures, followed by a review of P2P reputation systems. An overview on truthtelling mechanisms that are not specific to P2P systems concludes.

P2P Data Structures. P2P data structures address a core issue in data management: administering of huge sets of $(key, value)$-pairs under a high rate of parallel transactions. The various approaches [7–10] differ primarily with regard to *contact selection* and *path selection*, i.e., which are the peers a node can communicate with and forward messages to. The topology of the key space is closely related to contact selection and path selection. Common topologies include hypercubes (e.g., CAN [7]), rings (Chord [10]), virtual search trees (P-Grid [8]), and butterfly networks (Viceroy [9]); see [6] for an analysis of the impact of the topology on the characteristics of the data structures.

P2P Reputation Systems. All of these approaches assume that nodes readily follow the protocol. We think that this is not realistic. Reputation systems allow the peers to deal with unreliable nodes by collecting, distributing and aggregating feedback on the behavior of the participants in the past. One of the first reputation systems based on P2P data structures is [15]. The approach is based on complaints, i.e., negative feedback. Each peer stores the feedback it has generated in a global repository that is accessible by all peers. A peer assigned with more negative feedback than the global average is deemed unreliable. As a measure against spoof feedback, the approach proposes to check not only the number of complaints on the peer in question, but also the reputation of the peers which issued the complaints. But this does not help against a compromised global repository and comes with a large overhead. *EigenTrust* [16] is an approach to reputation systems that is based on a distributed eigenvector computation. The approach uses a P2P data structure to store a global trust vector. For each pair of peers, the trust vector contains a normalized reputation value, based on the number of satisfying and unsatisfying transactions. In order to avoid spoof feedback the reputation value of each peer is recursively weighted with the reputation of its 'observers'. However, cooperative peers are not forced to provide truthful feedback in settings such as ours. Another assumption that does not hold in P2P data structures is that an initial set of users is known to be trustworthy. *PeerTrust* [17] derives trust values from the satisfaction earned by each transaction, the credibility of the participating peers, the context of the transactions and community-specific issues. Similar to the other approaches, the trust model of PeerTrust depends on a secure, global data structure that stores feedback. Spoof feedback is addressed with a credibility factor derived from the assumptions that uncooperative peers tend to disseminate spoof feedback and cooperative peers usually issue truthful feedback. These assumptions may fail in the presence of groups of colluding peers which strive for 'strategic' goals, e.g., discrediting other nodes. A comparison of other P2P-based reputation systems is shown in [14].

It is challenging to secure global data structures against dishonest peers. A peer which wants to influence the reputation system could try to insert spoof feedback, tamper with feedback items it is supposed to forward and manipulate feedback in its local zone. FairNet [11], our reputation system for P2P data structures, avoids these vulnerabilities by introducing mechanisms that work on local data structures. In particular, the peers maintain local repositories and exchange feedback with every message that is sent out to another peer. With local repositories, an attacker that wants to modify a certain reputation value is forced to compromise the repositories of many peers. However, local repositories without further countermeasures may still fall prey to spoof feedback.

Truthtelling Mechanisms. In addition to mechanisms designed for certain reputation systems, others have investigated approaches to incentivize truthtelling. The approaches do not depend on a specific implementation. *CONFESS* [3] aims at eliciting truthful feedback in buyer-seller situations. The idea is that buyers who appear repeatedly will build a reputation for truthtelling in equilibrium. The authors formally prove the effectiveness of the mechanism under the given assumptions. However, their solution is not readily applicable to our setting, for two reasons. First, CONFESS requires a central instance that all participants deem trustworthy. This is different from P2P architectures. Another issue is that uncertainty/subjectivism is not part of the model, at least currently: If a seller behaves cooperatively, the buyer will always notice this. If the seller does not, the buyer will notice this as well. Our approach in turn does without this assumption. The rate of such errors is an endogenous parameter of our approach.

Other proposals, e.g., *Bayesian Truth Serum* [5] and *Peer-Prediction* [4], pursue a different (i.e., not reputation-based) approach to the same problem, albeit in a slightly different setting. They compare the probability distribution of truthful answers to other probability distributions (the one of the answers of all participants in the case of Peer-Prediction, and the one predicted to be the distribution of the answers of all participants in the case of the Bayesian Truth Serum). This comparison allows to maximize the expected payoff of truthful answers, as formally shown in the respective publications. Unlike CONFESS, it does so without requiring repeated interactions. However, both approaches are not applicable to our setting as well. First, Peer-Prediction requires that the probability distributions of answers (of truthful feedback, to translate this to our setting) is known; Bayesian Truth Serum in turn requires that peers come up with an estimate of this distribution. Another issue is that, in spite of the name of one of the approaches, they are not Peer-to-Peer. More specifically, it is unclear how to implement them in an environment consisting of only the peers (and no other instances that could act as coordinators etc.). Finally, to the best of our knowledge, there have only been few experiments evaluating these approaches [18].

7 Discussion

The experiments presented so far have acknowledged that our countermeasure can be used with CAN and FairNet. But it remains to be discussed if our countermeasure are applicable to other reputation systems and application scenarios as mentioned in Section 6. Unlike many other approaches, FairNet does not depend on global data structures. Instead, the peers manage and exchange feedback locally with each interaction. However, our approach does not depend on the location where the feedback is stored.

Instead, the peers just assign weights to the nodes which forward the feedback, according to the correlation between that feedback and the transaction outcomes observed. Thus, our approach is applicable to each reputation system where nodes exchange feedback items, e.g., [15, 19] or (with some changes in the architecture) [17].

Our approach relies on mechanisms to detect spoof feedback with little resource consumption. The downside is that the approach requires several interactions before adapting to the behavior of a node, according to our evaluation. Thus, our approach requires reputation systems characterized by a high throughput of feedback. However, this is generally the case in systems such as P2P data structures, and it is an attribute of many fields of application, e.g., semantic web or distributed search engines.

The experiment on collaboration attacks has indicated a significant improvement of the reliability in the presence of many peers issuing spoof feedback at a high rate. But the experiment has also shown that the countermeasure cannot prevent any transaction from being forwarded to unreliable peers. Therefore, our approach is only applicable in settings with many 'inexpensive' transactions where a few messages may get lost.

8 Conclusions

Spoof feedback is an important issue in any kind of reputation systems. Dishonest participants may wish to discredit others or try to take advantages from disseminating spoof feedback. The problem becomes even more difficult in distributed reputation systems for P2P data structures. Such settings are characterized by a high throughput of feedback and complex collaboration models with peers that cannot be observed from one instance. In this paper we describe the requirements that a reputation system for P2P data structures must fulfill and propose our new approach for truthful feedback. The approach takes advantage of the fact that each peer can observe the utility of feedback obtained from others after having observed the outcome of a transaction. The peers derive weighing factors of (feedback forwarder, feedback subject)-pairs.

We evaluate our approach with an analytic model and by means of extensive experiments. The analysis confirms that the differentiation between useful and less useful feedback is effective, irrespective of the particular implementation of the reputation system. The experimental evaluation demonstrates the applicability of our approach in realistic settings. It shows a significant reduction of the impact of collusion attacks where more than 90% of the peers issue spoof feedback.

References

1. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation Systems. Communications of the ACM (CACM) **43** (2000)
2. Khopkar, T., Li, X., Resnick, P.: Self-Selection, Slipping, Salvaging, Slacking, and Stoning: The Impacts of Negative Feedback at eBay. In: Proceedings of the 6th ACM Conference on Electronic Commerce (EC'05). (2005) 223–231
3. Jurca, R., Faltings, B.: CONFESS: Eliciting Honest Feedback without Independent Verification Authorities. Proceedings of the 6th International Workshop on Agent Mediated Electronic Commerce (AMEC'04) (2004)

4. Miller, N., Resnick, P., Zeckhauser, R.: Eliciting Informative Feedback: The Peer Prediction Method. Management Science **51** (2005) 1359–1373
5. Prelec, D.: A Bayesian Truth Serum for Subjective Data. Science **306** (2004) 462–466
6. Gummadi, K., Gummadi, R., Gribble, S.D., Ratnasamy, S., Shenker, S., Stoica, I.: The Impact of DHT Routing Geometry on Resilience and Proximity. In: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'03). (2003)
7. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A Scalable Content-Addressable Network. In: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'01). (2001)
8. Aberer, K.: P-Grid: A Self-Organizing Access Structure for P2P Information Systems. In: Proceedings of the 9th International Conference on Cooperative Information Systems (CoopIS'01). (2001) 179–194
9. Malkhi, D., Naor, M., Ratajczak, D.: Viceroy: A Scalable and Dynamic Emulation of the Butterfly. In: Proceedings of the 21th ACM Symposium on Principles of Distributed Computing (PODC'02). (2002) 183–192
10. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'01). (2001)
11. Böhm, K., Buchmann, E.: Free Riding-Aware Forwarding in Content-Addressable Networks. International Journal on Very Large Data Bases (VLDB) (2006)
12. Back, A.: Hashcash - A Denial of Service Counter-Measure. http://hashcash.org (2002)
13. Jakobsson, M., Juels, A.: Proofs of Work and Bread Pudding Protocols. In: Proceedings of the 4th International Conference on Communications and Multimedia Security (CMS'99). (1999)
14. Dewan, P., Dasgupta, P.: Securing P2P Networks Using Peer Reputations: Is There a Silver Bullet? In: Proceedings of the 2nd IEEE Consumer Communications and Networking Conference (CCNC'05). (2005)
15. Aberer, K., Despotovic, Z.: Managing Trust in a Peer-2-Peer Information System. In: Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM'01). (2001)
16. Garcia-Molina, H., Schlosser, M.T., Kamvar, S.D.: The EigenTrust Algorithm for Reputation Management in P2P Networks. Proceedings of the 12th International World Wide Web Conference (WWW'03) (2003)
17. Xiong, L., Liu, L.: PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. IEEE Transactions on Knowledge and Data Engineering (TKDE) **16** (2004)
18. Prelec, D., Weaver, R.G.: Truthful Answers are Surprisingly Common:Experimental Tests of the Bayesian Truth Serum. In: Proceedings of the Conference on Econometrics and Experimental Economics (CEEE'06). (2006)
19. Cornelli, F., Damiani, E., di Vimercati, S.D.C., Paraboschi, S., Samarati, P.: Choosing Reputable Servents in a P2P Network. In: Proceedings of the 11th International World Wide Web Conference (WWW'02). (2002) 376–386