Exploring the Feasibility of Proactive Reputations

Gayatri Swamynathan, Ben Y. Zhao and Kevin C. Almeroth
Department of Computer Science, UC Santa Barbara
{*gayatri, ravenben, almeroth*}*@cs.ucsb.edu*

ABSTRACT

Reputation mechanisms help peers in a peer-to-peer (P2P) system avoid unreliable or malicious peers. In application-level networks, however, short peer life-times mean reputations are often generated from a small number of past transactions. These reputation values are less "reliable," and more vulnerable to bad-mouthing or collusion attacks. We address this issue by introducing *proactive reputations*, a first-hand history of transactions initiated to augment incomplete or short-term reputation values. We present several mechanisms for generating proactive reputations, along with a statistical similarity metric to measure their effectiveness.

1. INTRODUCTION

The growing success of peer-to-peer (P2P) networks makes securing them an increasingly difficult research challenge. Popular P2P applications can support millions of users spread across numerous administrative and network boundaries. The heterogeneous and distributed user population means that at any given time, some peers will be compromised by malicious users using viruses, worms, or application-specific vulnerabilities. These threats pose significant risks to early adopters of next-generation P2P applications such as users of distributed file systems [21, 16], application-level multicast [22, 13] or Internet-scale query engines [2, 17].

The use of reputation systems can help applications preserve correct operation despite the presence of malicious users. A large body of literature has shown their impact on distributed applications in the form of increased levels of cooperation and trustworthiness among peers. Researchers are also using reputations to diagnose complex networking protocols such as the Border Gateway Protocol (BGP) [27]. A reputation system quantifies a peer's trustworthiness as an aggregation of ratings earned from previous interactions. Such interactions can include message forwarding, remote storage, file transfer, or financial transactions.

While reputations have been deployed in online shopping sites such as EBay [14], they are not necessarily a natural fit for the dynamic nature of peer-to-peer networks. Since reputations assess a peer's trustworthiness using historical feedback of its past interactions, longer peer lifetimes lead to more interactions and a more accurate reputation. In P2P networks, however, peers are often short-lived as they periodically exit the application or leave due to failures. This high rate of peer turnover, or churn, means a significant portion of peers will have relatively "short-term" reputations accrued from a small number of past interactions.

For applications that rely on peers for data storage, message forwarding, or distributed computation, choosing a peer based on short-term reputations is highly undesirable. How then can we provide reliable reputation ratings for unknown peers or newcomers? To address this question, we propose the idea of *proactive reputations*. Where traditional reputation systems rely on ratings assigned following transactions performed during the normal execution of an application, proactive reputations allow peers to proactively initiate transactions with one or more peers for the express purpose of generating reputation ratings. The result is a mechanism for quickly generating reliable reputations for new peers or those with short lifetimes.

This paper offers three key contributions. First, we introduce the concept of proactive reputations and describe related research challenges in peer-to-peer networks. Next, we investigate information theoretic metrics to assess the effectiveness of proactive reputation generation. Finally, we propose a set of mechanisms to generate proactive reputations and conduct initial experiments to measure their effectiveness.

The remainder of the paper is organized as follows. We begin by describing the concept of proactive reputations in Section 2. Next, we outline the generation of proactive reputations, and present our methodology in Section 3. We then discuss our performance evaluation in Section 4. Finally, we discuss related work in Section 6 and conclude in Section 7.

2. PROACTIVE REPUTATIONS

In a traditional reputation system, peers assign ratings to others after concluding transactions with them. A peer looking to initiate a transaction, the *initiator*, can use reputations to choose the *candidate peer(s)* with which to interact. Because peers who seek to access another's reputation has no way to directly influence the quality of that reputation, we call this traditional approach *passive*.

Since reputation values are generally aggregates of per transaction feedback values, the "reliability" of a peer's reputation depends very much on the number of past transactions taken into account. In volatile P2P systems, however, the *passive* approach to reputations means peers will often base their interaction choices on reputation values that are "short-term," meaning they are derived from feedback following a relatively small number of past transactions.

In this paper, we propose a *proactive* reputation model for networked systems with verifiable, low-cost transactions. In a proactive reputation system, a peer initiates transactions with a targeted peer for the express purpose of understanding the peer's reliability for future transactions. For example, if Peer X needs to interact with two new peers or peers with "short-term reputations," it can initiate a number of requests for these peers in order to gauge their reliability and trustworthiness. Unlike challenge-response mechanisms where the candidate has a clear incentive to respond correctly, the goal of proactive requests is to blend in with regular application-level traffic to measure the candidate's "normal" response.

Proactive reputations are complementary to traditional, passive reputations. An initiator can proactively probe those candidates it is less confident about, while undertaking normal transactions with other peers known to be trustworthy.

Requirements. There are two requirements that must be satisfied for proactive requests to be feasible. First, transactions must have "low cost," and carry uniform application-level "value." A low cost, where cost is measured by the resource overhead consumed per transaction, ensures that utilizing proactive requests does not create significant overhead for their initiators. Uniform value across transactions means proactive transactions have the same "priority" and will be treated similarly as a "typical" transaction. For example, an online auction system such as EBay would not satisfy this requirement. Its transactions vary highly in value, and transactions of low value do not necessarily serve as useful indicators of peer behavior for high-valued transactions. Transactions in cooperative peer-to-peer systems such as structured overlays, in contrast, generally incur low resource costs (bandwidth and processing time), and any variance in "value" across transactions is generally hidden to the request handling peer.

The second key requirement is that transactions must be "verifiable." That is, the initiator peer must have a definitive mechanism for testing whether the transaction was performed properly. In the context of online e-commerce communities, this is analogous to confirming the promised product or payment was received on time. For P2P systems, the initiator can request that a trusted third party verify the transaction result. For message routing, the initiator can route a message to a third party verifier via the candidate peer, and wait for an acknowledgment. For storage, the peer or a trusted party can read the stored data and confirm its contents. The third party verifier can be chosen in two ways. The initiator can choose a trusted party based on existing trust relationships or reputation values. As an alternative, it can exploit the free-cost nature of P2P identities to create a second virtual identity who appears independent from its main identity. This mechanism leverages the Sybil attack [12] on a small scale to improve security.

Benefits. Augmenting a reputation system with proactive reputations has two main benefits. First, from the initiator's perspective, proactive reputations generate a more reliable credibility measure of the target peer. Results from proactive requests are formed from "firsthand" (when using a virtual identity for verification) or trusted observations, and thus are less vulnerable to false ratings or collusion. Second, accruing passive ratings takes time and depends on the target peer's level of interaction with others during the normal execution of an application. Consequently, a significant amount of time may be needed to establish a reliable reputation value. With our proactive model, a peer controls its transaction rate with the target peer, and can generate its reputation quickly (within bounds of detection).

Challenges. We face several major challenges in designing a proactive reputation system. First, for the processing of proactive requests to be fair and non-biased, the receiver must not be able to identify the request initiator. Otherwise, a target peer can tailor its response based on the originator. Therefore, these proactive requests should be anonymous [11, 28]. Second, in order for proactive requests to generate accurate peer behavior, they must be indistinguishable from normal application requests. Requests can come in the form of routing requests, storage directives, or other application-specific actions. Once a malicious peer determines that the purpose of a request is to measure its performance and reliability, it will always process these requests correctly to boost its reputation. Finally, we must design proactive reputations to minimize computation and bandwidth overhead.

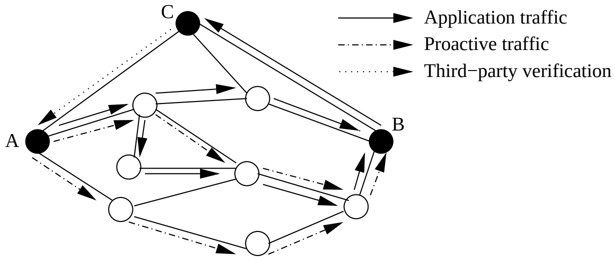

Figure 1: Proactive reputation generation: the initiator node injects proactive requests into the application stream on its way to the candidate node. A trusted 3rd party verifies the transaction outcome and informs the initiator.

In this work, we assume that proactive requests are sent as anonymous messages, and focus on addressing the challenge of hiding proactive requests inside application traffic. In the remainder of the paper, we present an architecture for proactive requests and evaluate its effectiveness at evading detection. While this approach does impose communication and computation overheads on the system, a detailed study of this overhead is the subject of ongoing work.

3. RESISTING TRAFFIC ANALYSIS

In this section, we describe how to generate proactive requests, and consider a variety of metrics to quantify its success in evading detection. Consider an initiator, A, that wishes to test Candidate B's behavior via proactive requests. As shown in Figure 1, A will forward a number of anonymous messages to B, and enlist the help of a third party verifier, C. Note that C can be a second virtual identity belonging to the same user as A. For simplicity, we assume that each proactive request fits inside a single overlay message.

Peer B can easily detect anonymous proactive requests injected into the network, since they stand out from traffic with associated identities. Once detected, a malicious node can temporarily behave well to boost its reputation. To make proactive requests indistinguishable from normal traffic, we require that all nodes anonymize a portion of the messages they originate. The result is that any node will see a mixed sequence of normal and anonymous messages. An initiator can then *inject* a sequence of anonymous proactive requests that blend with normal traffic. From the perspective of the target peer, it should be very difficult to distinguish statistically whether an anonymous packet is part of a proactive request, or simply part of a peer's anonymized outgoing traffic.

A proactive reputation scheme should be resilient to traffic analysis by the target peer. While the target peer can monitor all transaction requests it observes in the network, it should not be able to distinguish between normal application traffic and the proactive requests. As a result, proactive reputations would not only be able to detect malicious behavior, but also encourage peers to participate honestly at all times.

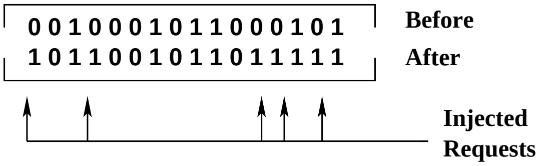

Figure 2: A binary representation of the distribution of anonymous messages inside a finite sized history window. The target node can compare a window of recently seen messages against a reference window to detect proactive requests.

3.1 A Stream Similarity Metric

With this goal in mind, we investigate several metrics that could be employed to accurately assess the generation of proactive reputations. Our objective is to quantify how "random" a stream of open and anonymous requests looks to the target. Assuming that proactive requests are indistinguishable from normal traffic at the application level, we use binary values of 0 and 1 to represent open or anonymous requests received by the target peer. To detect proactive requests statistically, the target peer would maintain a window of recently observed requests, and compare its rate of anonymous versus normal messages against a reference window of "normal" traffic. As shown in Figure 2, if the current window of messages differs substantially from the reference window, the target can assert that it is being probed with proactive requests, and behave differently.

We investigated three different information theoretic measures as possible metrics for a target peer to use in detecting proactive bursts in the transaction stream. These metrics are: *conditional entropy*, *relative entropy* (or *Kullback-Leibler* (KL) distance), and *histogram similarity*. These metrics measure the statistical similarity between two streams, and can be used to estimate the resilience of a stream containing proactive requests against statistical traffic analysis. In each case, the buffer of recent messages is used to gather statistics on the current message stream. A larger buffer provides a bigger dataset and increased accuracy in extracted statistics.

A distribution's entropy is a measure of the randomness it contains. The first metric we examine, conditional entropy, measures the likelihood of predicting the $(N+1)^{th}$ value given the last *N* values. Given the target peer's buffer of recent messages, this metric represents the amount of autocorrelation in the incoming stream, but not the statistical similarity between two streams.

A more appropriate metric is the relative entropy or the Kullback-Leibler (KL) distance, an entropy-based measure of dissimilarity between two probability distributions. The KL-divergence metric, however, is known to lack robustness to small sample set sizes. Given that history buffers at target nodes are finite in size, this metric is unlikely to produce the most efficient detector of proactive requests.

Our ongoing search led us to the related areas of multimedia databases and bioinformatics, where similarity metrics are used to index and retrieve documents, images, musical pieces, and biological sequences [7, 8, 15, 19]. Similarity between data sets is determined using frequency histograms. Histogram similarity metrics include weighted Euclidean distance, square distance, and absolute difference. Smaller values from these measures indicate a higher level of similarity between two streams.

As shown in Figure 2, we need to determine the similarity between two traffic streams observed by the target peer: a normal application traffic without proactive bursts and the current traffic stream possibly injected with bursts of proactive requests. We found histogram similarity metrics to be the best solution currently available, and will discuss their use in more detail in Section 4.

Parameter	Range	Default
Size of network	50-100	50
# of transactions	100-10000	5000
Proactive burst size	0-70	40
Window Size	50-500	100
Anon. rate (Model 1)	0-70	30

Table 1: Simulation Parameters

3.2 Producing Anonymous Cover-Traffic

We now present a set of mechanisms to shape normal application traffic in order to provide sufficient cover for anonymous proactive requests. For simplicity, we assume a uniform request rate across the nodes in the network.

We now describe three candidate models that determine how nodes in the network anonymize their outgoing traffic. In the first model, each peer in the network anonymizes outgoing traffic at a predefined constant rate. We will investigate the effectiveness of this model for a range of anonymization rates. In our second model, peers in the network vary their rate of anonymous transactions at some predefined time interval. The rate of change per hour is randomly chosen across a predefined range, e.g. 20-80%. Finally, we consider a third and most fine-grained model of traffic anonymization. In this model, peers dynamically define a random number of messages N and a random anonymization rate R for these messages. The peer applies R to the next N incoming messages, and then resets both values.

Our objective in developing these models is to choose the optimal anonymization scheme that will allow a peer to inject the maximum sized burst of proactive requests to a target peer without detection. Therefore, our metric of success is how large of a consecutive burst of proactive requests can be injected before the target peer successfully detects a change in stream statistics.

Our proposed methodology is presented in Figure 1. The source node sends a burst of anonymous proactive requests to the target peer. The requests are routed by the network overlay to the target peer, and mixes with normal application traffic from other peers. Our traffic anonymization models result in a mix of anonymous and open messages seen by the target peer.

4. EVALUATION

In this section, we evaluate our methodology by performing two sets of initial experiments. Our goal is to quantify the effect of specific anonymization models and system parameters on the ability of a candidate peer to detect *proactive requests*.

4.1 Simulation Setup

We have implemented a simulator in NS-2 using OTcl and C++. Table 1 summarizes the main simulation parameters and the range of values tested. These ranges were selected based on their coverage of the likely performance characteristics. In addition to these parameters, networks of nodes were

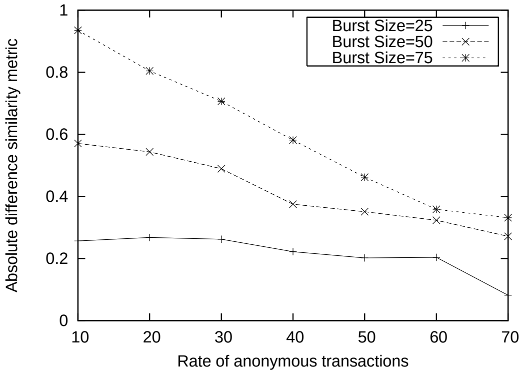

Figure 3: The effect of increasing anonymous rate and burst size on the preset anonymous rate model.

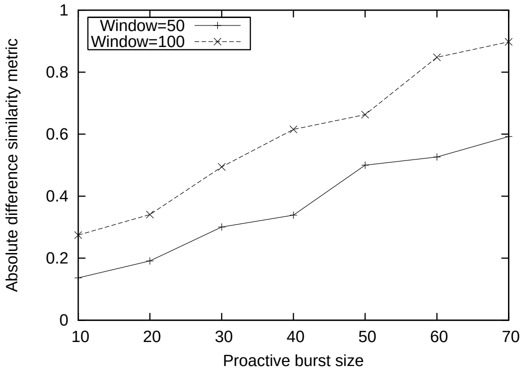

Figure 4: The effect of increasing window sizes on the preset anonymous rate model.

constructed using GT-ITM-based topologies [4]. In these networks, approximately 25% of the nodes transact with the target peer as part of normal application execution. The target peer processes approximately 5,000 transactions in each simulation run. Each data point in our figures represents the average value of three randomized runs.

The main focus of our evaluation is to determine the similarity between two traffic streams observed by a target peer: a normal application traffic stream, and an application traffic stream with injected bursts of proactive requests. We propose three models of application traffic: a preset anonymous rate, a per-hour anonymous rate, and a per-set-of-transactions anonymous rate. A traffic stream is modeled as a a series of binary values, a “0” represents an open transaction and a “1” represents an anonymous transaction.

Out of the three candidates described in Section 3, we choose the Absolute Difference (AD) metric for our experiments. First, the weighted Euclidean distance does not apply to our data set, since we give equal weight to all data values. Second, because our data streams are composed of binary values, the square distance and absolute difference metrics will produce identical results.

Let $Ha(j)$ represent the histogram bin value of the j consecutive 1s in the application traffic. That is, $Ha(1)$ would be the frequency count of single 1s in the traffic stream; $Ha(2)$ would be the frequency count of two consecutive 1s in the stream; and so on. Similarly, let $Hp(j)$ represent the histogram bin value of j consecutive 1s in the application stream with proactive bursts. We define the Absolute Difference (AD) metric as:

$$AD = \sum_{j=1}^{N} \mid Ha(j) \text{ - } Hp(j) \mid \tag{1}$$

The maximum number of histogram bins is represented by N. This maximum number of bins would be equal to the window size at the target peer because the target could observe a stream of consecutive 1s equal in length to the window size. The smallest values of the absolute difference metric represent the best similarity of the two data streams. Therefore, a good proactive reputation generation scheme is one that ensures low absolute difference values.

4.2 Simulation-Based Experiments

Our first set of experiments evaluates the preset anonymous rate model. In this model, each peer in the network transacts anonymously at a predefined rate, X. This rate varies from 10% to 70% of the total number of transactions.

Figures 3 and 4 illustrate the effect of increasing the rate of anonymous transactions, burst size, and window size on the preset anonymous rate model. As seen in Figure 3, small proactive bursts are better hidden than large bursts. Additionally, small bursts are better hidden as the number of anonymous transactions increases. With an anonymous rate greater than 70%, small proactive bursts show a near zero absolute difference value, indicating that they go essentially undetected. On the other hand, large bursts perform poorly with low anonymous transaction rates, but are better hidden as the amount of anonymous traffic increases. As expected, hiding proactive requests is much easier among a high percentage of anonymous transactions.

The next experiment evaluates the effect of the size of a target peer’s buffer window size. A larger buffer window should give the target peer a better chance of identifying traffic as active probes.

Figure 4 illustrates the effect of varying the window size on the preset anonymous rate model. In this experiment, we employ two window sizes of 50 and 100 transactions. We maintain a 30% anonymous rate for the experiment. For each window size, there is an increase in the absolute difference value as the burst size increases. This result occurs because a 30% anonymous rate is able to hide small bursts but is not effective for large bursts. With an increase in window size, the absolute difference values between the normal application stream and the proactive stream increases. With a larger buffer, a target peer is better able to detect that it is being probed for reputation assessment. We observe similar results when varying the window size for the per-hour and per-transaction-set traffic models.

Our final experiment compares the three models of generating application traffic. This comparison is conducted with respect to the size of proactive bursts. The traffic generated by the three models, before proactive bursts are inserted, is modeled as follows. For the preset anonymous rate model, each node in the network overlay generates approximately 30% anonymous traffic. For the per-hour and per-transaction-set anonymous rate models, the rate varies randomly between 20% and 80% per hour, and per set of transactions, respectively.

Given the three anonymization models, we examine the effect of varying the length of proactive bursts injected into the network from 10 to 70 messages. The target peer main-

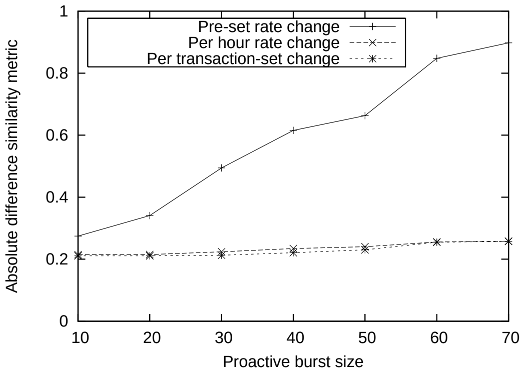

Figure 5: A comparison of the three application traffic models.

tains a window size of 100 transactions. The experiment proceeds by having network overlay nodes conduct normal transactions, open and anonymous, as modeled by the application traffic models. Proactive requests targeted towards a specific peer are routed by overlay nodes, and reach the target depending on the network topology and background traffic.

As seen in Figure 5, the per-hour and per-transaction-set models perform significantly better than the preset model. As observed earlier, a fixed anonymous rate of only 30% performs poorly with increasing burst sizes. Thus, higher burst sizes result in a more dissimilar stream than lower burst sizes. The second and third models, however, continue to perform well, even with higher proactive burst sizes. This result occurs because both models are more dynamic than the first model. Each peer generates anonymous transactions at a different rate, and varies this rate over each hour or over a specific set of transactions.

Finally, we can make two interesting observations from these results. First, we were moderately surprised to see that the per-transaction-set did not perform significantly better than the per-hour model. It seems that the mixing of streams prior to arrival at the target produced sufficient statistical variance to cover the proactive requests. Second, both per-hour and per-transaction-set models saw little change across different sized proactive bursts. While these results are generally positive, we are running more detailed experiments in order to better understand their underlying factors.

5. DISCUSSION

While this paper discusses the high level concepts related to proactive reputations, the details of a number of issues remain the focus of ongoing work.

Minimum Anonymity Required. In this paper, we assume the availability of a fully anonymous routing layer and focus on the challenge of hiding proactive requests inside application traffic. One interesting question is, how much anonymity is required to satisfy our requirements for evading detection? While prior work on anonymity measures anonymity against powerful colluding attackers, we require a much weaker level of anonymity. Given the number of network messages, a target peer cannot expend significant resources to determine the source of a single request. Therefore, simple forwarding through one or more relay peers (who can be secondary identities for the same physical user) should suffice.

Integration into Global Reputations. Proactive reputations augment traditional global reputation systems with on-demand, first-hand transaction feedback. They can provide guidance to peers interested in interacting with a target peer, but are specific to the peer who initiated the requests. Ideally, they should be integrated into the global reputation for the target peer so that other peers can benefit from a more reliable global reputation value.

An interesting question is how to integrate proactive reputations with global reputations while avoiding vulnerability to collusion. For example, a colluding peer could offer strong support for a malicious peer in the form of positive proactive reputations. We believe that integration should be handled on a per-peer basis, where a peer, A, interested in the reliability of peer, B, can access both B's global and per-peer proactive reputations, and use its own discretion in discarding or using any of the proactive values.

6. RELATED WORK

Several reputation systems have been proposed to discourage maliciousness and motivate trustworthiness and cooperation in P2P networks [1, 9]. Protecting these systems against the Sybil attack [12] remains a significant challenge [6]. Some solutions address the problem of false ratings and dynamic peer personalities [3, 25, 26]. Finally, controlled anonymity has been shown to avoid peer discrimination [10].

Research on similarity-based data retrieval and indexing has led to metrics that measure similarity between documents, images, musical pieces, and biological sequences [7, 8, 19]. Multimedia databases use histogram-based similarity metrics for image retrieval [15], while entropy-based metrics are used to determine stationarity in Internet measurements [18].

Finally, extensive work exists on the subject of anonymous communication. A majority of these projects use the Chaum-Mix [5] model, including Onion Routing [20], Tor [11], and most recently, Cashmere [28]. In addition, P5 [23] and Herbivore [24] use the dining cryptographer model.

7. CONCLUSIONS

High churn rates in dynamic networks pose a serious challenge to the adoption of reputation systems that depend on long-term state for accuracy. In this paper, we propose a novel approach of quickly generating reliable reputations through the use of proactive transaction requests. By blending in with ordinary application-level traffic, these verifiable requests provide a first-hand estimation of the reliability and trustworthiness of unknown peers. To ensure that these requests are treated like normal application requests, we anonymize them and provide cover traffic by anonymizing a portion of normal application traffic. We use a stream similarity metric to evaluate the effectiveness of these approaches and conduct initial experiments to measure their resistance to detection. For low window sizes, our results demonstrate that bursts of proactive requests blend in well and are nearly undetectable under traffic analysis by target peers.

We note that this work focuses on the feasibility of adopting long-term reputation systems for high-churn networks, and does not address other shortcomings of general reputation systems. These include vulnerability to collusion attacks, as well as attacks based on dynamic peer behavior, where an attacker behaves well in order to build a sufficient reputation to launch a single focused attack. Addressing these vulnerabilities using proactive reputations is the topic of ongoing research.

REFERENCES

[1] ABERER, K., AND DESPOTOVIC, Z. Managing trust in a Peer-2-Peer information system. In *Proc. of CIKM* (November 2001).

[2] BHARAMBE, A. R., AGRAWAL, M., AND SESHAN, S. Mercury: supporting scalable multi-attribute range queries. In *Proc. of SIGCOMM* (Portland, OR, September 2004).

[3] BUCHEGGER, S., AND BOUDEC, J. L. A robust reputation system for P2P and mobile ad-hoc networks. In *Proc. of P2PEcon* (June 2004).

[4] CALVERT, K. L., DOAR, M. B., AND ZEGURA, E. W. Modeling internet topology. *IEEE Communications 35*, 6 (June 1997).

[5] CHAUM, D. L. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM 24*, 2 (1981).

[6] CHENG, A., AND FRIEDMAN, E. Sybilproof reputation mechanisms. In *Proc. of P2PEcon* (August 2005).

[7] CILIBRASI, R., AND VITANYI, P. M. B. Clustering by compression. *Transactions on Information Theory 51*, 4 (2005), 1523–1545.

[8] CILIBRASI, R., VITANYI, P. M. B., AND DE WOLF, R. Algorithmic clustering of music based on string compression. *Computer Music Journal 28*, 4 (2004), 49–67.

[9] DAMIANI, E., ET AL. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proc. of CCS* (November 2002).

[10] DELLAROCAS, C. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proc. of EC* (Oct. 2000).

[11] DINGLEDINE, R., MATHEWSON, N., AND SYVERSON, P. Tor: The second-generation onion router. In *Proc. of USENIX Security Symposium* (August 2004).

[12] DOUCEUR, J. The sybil attack. In *Proc. of IPTPS* (March 2002).

[13] DUNAGAN, J., HARVEY, N. J. A., JONES, M. B., KOSTIC, D., THEIMER, M., AND WOLMAN, A. Fuse: Lightweight guaranteed distributed failure notification. In *Proc. of OSDI* (San Francisco, CA, December 2004).

[14] EBAY. ebay home page, http://www.ebay.com, 2005.

[15] FURHT, B., AND SAKSOBHAVIVAT, P. A fast content-based multimedia retrieval technique using compressed data. In *Proc. of SPIE MSAS* (Nov. 1998).

[16] HAEBERLEN, A., MISLOVE, A., AND DRUSCHEL, P. Glacier: Highly durable, decentralized storage despite massive correlated failures. In *Proc. of NSDI* (Boston, MA, May 2005).

[17] HUEBSCH, R., ET AL. Querying the internet with PIER. In *Proc. of VLDB* (Sept. 2003).

[18] KRISHNAMURTHY, B., MADHYASTHA, H. V., AND VENKATASUBRAMANIAN, S. On stationarity in internet measurements through an information-theoretic lens. In *Proc. of NetDB* (April 2005).

[19] LI, M., CHEN, X., LI, X., MA, B., AND VITANYI, P. The similarity metric. In *Proc. of SODA* (January 2003).

[20] REED, M. G., SYVERSON, P. F., AND GOLDSCHLAG, D. M. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications 16*, 4 (May 1998).

[21] RHEA, S., ET AL. Pond: The OceanStore prototype. In *Proc. of FAST* (April 2003).

[22] ROWSTRON, A., KERMARREC, A.-M., DRUSCHEL, P., AND CASTRO, M. SCRIBE: The design of a large-scale event notification infrastructure. In *Proc. of NGC* (November 2001), pp. 30–43.

[23] SHERWOOD, R., BHATTACHARJEE, B., AND SRINIVASAN, A. P5: A protocol for scalable anonymous communication. In *Proc. of IEEE Symposium on Security and Privacy* (Oakland, CA, May 2002).

[24] SIRER, E. G., GOEL, S., ROBSON, M., AND ENGIN, D. Eluding carnivores: File sharing with strong anonymity. In *Proc. of ACM SIGOPS European Workshop* (Leuven, Belgium, September 2004).

[25] SWAMYNATHAN, G., ZHAO, B. Y., AND ALMEROTH, K. C. Decoupling service and feedback trust in a peer-to-peer reputation system. In *Proc. of AEPP* (July 2005).

[26] XIONG, L., AND LIU, L. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering 16*, 7 (2004).

[27] YU, H., REXFORD, J., AND FELTEN, E. A distributed reputation approach to cooperative internet routing protection. In *Proc. of NPSec* (November 2005).

[28] ZHUANG, L., ZHOU, F., ZHAO, B. Y., AND ROWSTRON, A. Cashmere: Resilient anonymous routing. In *Proc. of NSDI* (May 2005).