ACCEPTED FROM OPEN CALL

Reputation Estimation and Query in Peer-to-Peer Networks

Xing Jin, Oracle USA

S.-H. Gary Chan, HKUST

ABSTRACT

Many peer-to-peer systems assume that peers are cooperative to share and relay data. But in the open environment of the Internet, there may be uncooperative malicious peers. To detect malicious peers or reward well behaved ones, a reputation system is often used. In this article we give an overview of P2P reputation systems and investigate two fundamental issues in the design: reputation estimation and query. We classify the state-of-the-art approaches into several categories and study representative examples in each category. We also qualitatively compare them and outline open issues for future research.

INTRODUCTION

In recent years peer-to-peer (P2P) systems have seen enormous success and rich developments over the Internet. Typical applications include file sharing, streaming, Internet telephony, and overlay routing. According to CacheLogic Research, in 2006 P2P traffic accounted for over 72 percent of Internet traffic that year.

In P2P systems cooperative peers self-organize themselves into overlay networks and store or relay data for each other. Many P2P systems work on the assumption of truthful cooperation among peers. However, in the open environment of the Internet, some participating peers may not cooperate as desired. They may be selfish and unwilling to upload data to others, or they may have abnormal actions such as frequent rebooting that adversely affect their neighbors. More seriously, some peers may launch attacks to disrupt service or distribute viruses in the overlay network. We call all these uncooperative, abnormal, or attacking behaviors *malicious actions* and the associated peers *malicious peers*.

Malicious peers may seriously degrade the performance of P2P networks. Liang *et al*. have tracked several attacking behavior in practical P2P file sharing systems [1]. They find that more than 50 percent of copies of popular songs in KaZaa are polluted, meaning that the content downloaded from the network is different from the downloader's expectation (e.g., the content is corrupt and cannot be played, or the content is a different song from the search index metadata). Their study also shows that both structured and unstructured P2P file sharing systems are highly vulnerable if attackers insert massive bogus records to poison search indexes.

To detect malicious peers or reward well behaved ones, a reputation system is often used. In a typical reputation system each peer is assigned a reputation value according to its performance history. Differentiated services are then provided to peers according to their reputation. While the basic idea is simple, a practical system design is not easy. Generally, a P2P reputation system consists of three functional components [2]: collecting information on peer behavior, scoring and ranking peers, and responding based on peers' scores. All these components are nontrivial, especially given the following consideration:

- *Scalability*: A large P2P network may have hundreds of thousands of peers. For example, Skype has several million online users. A reputation system should be highly scalable in terms of peer number.
- *Adaption to peer dynamics*: Peers may join or leave at any time. If reputation information is maintained at peers, peer leaving may lead to information loss. A robust reputation system should take peer dynamics into account.
- *Security*: Malicious peers may endeavor to break down the reputation system so that they can conduct malicious actions without being detected. For example, peers may purposefully leave and rejoin the system with a new identity in order to shed any bad reputation [2]. Clearly, a good reputation system should be secure to resist these adversarial behaviors.

In this article we study two fundamental issues in P2P reputation systems.

Reputation estimation: An estimation method describes how to generate peer reputation based on others' feedback. We classify existing estimation methods into three categories: social network, probabilistic estimation, and game-theoretic model. We select representative examples from each category, and discuss their advantages and limitations. As many estimation methods rely on specific feedback collection mechanisms, we also discuss feedback collection mechanisms when necessary.

Reputation query: Reputation query in P2P

This work was supported, in part, by the Hong Kong Innovation and Technology Fund (ITS/013/08).

networks is not trivial. First, efficient data storage and retrieval is always a challenging issue in P2P networks. Huge amounts of data require distributed storage approaches. Then efficient retrieval becomes nontrivial. Peer dynamics bring more difficulty. Second, reputation data are highly security-sensitive. The reputation of a peer cannot be locally stored at the peer itself, because a dishonest peer may misreport its reputation value in order to gain rewards or avoid punishments. We also need to consider security issues in reputation delivery. In this study we survey the state-of-the-art approaches to reputation storage and retrieval in P2P networks. We classify them into three categories. For each category, we discuss illustrative examples. We also qualitatively compare them and outline possible directions for future research.

There are many other important issues in P2P reputation systems; for example, how to prevent targeted and adversarial attacks? How to interpret reputation? Interested readers may refer to [2, 3] for a comprehensive overview of P2P reputation issues.

The rest of the article is organized as follows. In the next section we explore the reputation estimation issue. We then discuss reputation query techniques. We conclude in the final section.

Reputation Estimation

There are mainly three reputation estimation methods in current P2P networks. The first one is the social network, where all feedback available in the network are aggregated to compute peer reputation. The second one is probabilistic estimation, which uses sampling of the globally available feedback to compute peer reputation. The third one is the game-theoretic model, which assumes that peers have rational behavior and uses game theory to build a reputation system. We elaborate on these methods below.

Social Network

Approaches based on the social network can be further divided into two categories: *separated reputation model* and *correlated reputation model*. In a separated reputation model only the direct transaction partners of a peer (e.g., resource provider/downloader or streaming neighbor) can express their opinion on the reputation of the peer. A practical example is the eBay reputation system (although eBay is not a P2P network). After each transaction at eBay, the buyer and the seller rate each other with positive, negative, or neutral feedback. The reputation is calculated at a central server by assigning 1 point for each positive feedback, 0 point for each neutral feedback, and –1 point for each negative feedback. The reputation of a participant is computed as the sum of its points over a certain period. Considering that peers may lie in their feedback, Mekouar *et al*. propose to monitor suspicious feedback [4]. That is, after each transaction between a pair of peers, both peers are required to generate feedback to describe the transaction. If there is an obvious gap between the two pieces of feedback, both are regarded as suspicious. Later on, the more suspicious feedback a peer generates, the smaller weight in reputation computing its feedback has. Similarly, in [5] a peer's reputation is computed as a weighted average of feedback from direct witnesses of its performance. Xiong *et al*. develop a general reputation model, which considers, for example, feedback from peers, the trustworthiness factor of feedback sources, and the transaction context factor for discriminating transaction importance [6]. Almost all separated reputation models can be expressed by this model.

In a correlated reputation model the reputation of a peer is computed based on the opinion of its direct transaction partners as well as third-party peers. In this model a peer, A, who wishes to know the reputation of another peer, B, can ask some peers (e.g., its neighbors) to provide their opinion on B (although some of the peers may not have conducted any transaction with B). A then combines peer opinions to calculate B's reputation. We take EigenTrust as an example [7]. In EigenTrust, whenever a peer conducts a transaction with another peer, they keep reputation values for each other. If there is no direct transaction between two peers, they keep a zero reputation value for each other. Peers then iteratively update the reputation values. Each time peer A wishes to update the reputation of peer B, A asks for B's reputation from all other peers in the system. A then computes a weighted sum of these reputation values and keeps the result as the new reputation of B. In each iteration all peers conduct the above reputation update. The process continues until the reputation values kept at different peers converge. Another example is the network information and control exchange (NICE) reputation model [8]. Each peer holds the reputation of its transaction partners according to the quality of transactions. All peers further form a trust graph based on reputation values. Later on, an overlay path between two peers is selected as the most trustworthy path between them in the trust graph.

The correlated reputation model is more like our real social network, where third-party peers can express their opinion on a peer. But it costs more to collect and aggregate third-party opinion. For example, EigenTrust takes a long time to wait for reputation values to converge.

The correlated reputation model is more like our real social network, where third-party peers can express their opinion on a peer. But it takes more cost to collect and aggregate third-party opinion. For example, EigenTrust takes a long time to wait for reputation values to converge.

Probabilistic Estimation

This approach uses sampling of the globally available feedback to compute peer reputation. It usually relies on some assumptions on peer behavior. For instance, it may assume that a peer is trustworthy with a certain but unknown probability. And when sharing its own experience with others, a peer may lie with some, again unknown, probability [9]. It then uses probabilistic estimation techniques to estimate all unknown parameters. Many estimation methods may be used. Despotovic *et al*. use maximum likelihood estimation, which assumes that peers do not collude [9]. Mui *et al*. use Bayesian estimation, which uses only direct interaction among peers and does not use third-party opinion [10].

By using a small portion of the globally available feedback, the probabilistic model has lower cost in feedback collection than the social net-

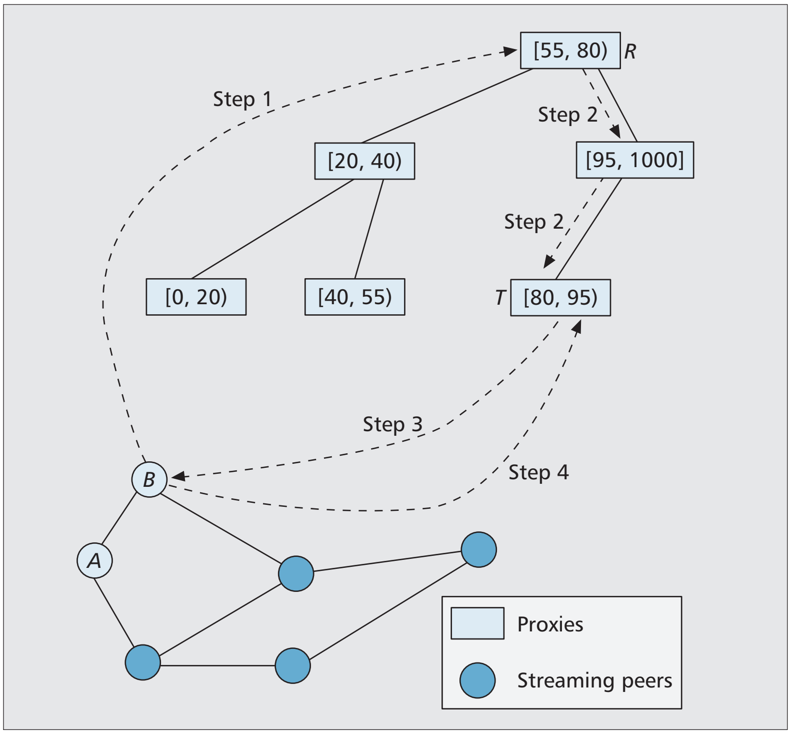

Figure 1. *Process of submitting a report about a streaming peer* A *by its child* B *for the first time in a proxy-based approach (from [5]). Step 1)* B *sends* A*'s IP address to* R*. Suppose that* A*'s IP address is represented by a numerical value 88. Step 2)* R *searches in the binary tree to identify the proxy that manages 88 (*T *in this case). Step 3)* T *responds to* B *with its certificate. Step 4) After verifying the trustworthiness of* T, B *sends its report about* A *to* T.

work approach. On the other hand, the social network approach can use a complicated reputation model, and is robust to a wide range of malicious actions. But the probabilistic model can be applied to only simple reputation models (due to the difficulty in probabilistic estimation) and is effective against only a few kinds of malicious actions. The performance of the two models has been compared in [11]. It has been shown that the probabilistic model performs better for small malicious populations, while the social network approach is better when most peers are malicious.

Game-Theoretic Model

Different from the above two approaches, the game-theoretic model assumes that peers have rational behavior and uses game theory to build a reputation system. Rational behavior implies that there is an underlying economic model in which utilities are associated with various choices of peers and that peers act so as to maximize their utilities. Fudenberg *et al*. present a game-theoretic framework to offer certain characterizations of the equilibria payoffs in the presence of reputation effects [12]. But the work assumes that a central trusted authority does feedback aggregation, which may not be scalable to large-scale P2P networks.

Reputation Query

In this section we discuss techniques for reputation query in P2P networks.

Centralized and Partially Centralized Structures

The simplest solution is to use a powerful server to keep the reputation of all peers. For example, eBay uses a central server to collect and keep all users' reputations. Feedback from users is sent to and stored at the server. A query of a user's reputation is also sent to and answered by the server. Similar approaches have been used in [13].

A centralized approach is easy to implement and deploy. Security of a central server is much easier to achieve than that of distributed components in a distributed approach. Furthermore, in a centralized approach, reputation management is independent of peer joining and leaving, which greatly simplifies reputation retrieval. However, a centralized approach is not scalable to large P2P networks. Also, the server forms a single point of failure, making the system vulnerable.

To address the limitations of the centralized approach, a partially centralized approach, which uses a set of servers instead of a single server, has been proposed. Mekouar *et al*. propose a malicious detector algorithm (MDA) to detect malicious peers in KaZaa-like systems [4]. KaZaa is a partially centralized P2P file sharing system with a set of supernodes. Each ordinary peer is attached to a unique supernode. MDA assumes that supernodes are all trustworthy and maintain reputation information for ordinary peers. All evaluation results about a peer are maintained at its attached supernode. Supernodes can then enforce differentiated service to peers according to their reputation.

Note that supernodes in KaZaa are self-elected from ordinary peers and may not be fully trustworthy. One approach uses predeployed proxies instead of supernodes for reputation maintenance [5]. In this approach each peer is attached to a unique proxy according to its IP address. Correspondingly, each proxy is responsible for a certain IP range, and proxies are organized into a binary search tree based on the IP ranges they maintain. Each peer periodically generates reports about its streaming neighbors. All reports about a peer are sent to its attached proxy. A query about a peer's reputation is also forwarded to and answered by the peer's attached proxy. Figure 1 shows the report submission process in this approach [5]. Each circle in the figure is a streaming peer, and each quadrangle is a deployed proxy. Numbers in a quadrangle indicate the IP range maintained by the proxy (here numerical values are used to represent IP addresses). Suppose that streaming peer *B* is streaming peer *A*'s child in the streaming overlay, and *B* prepares to submit a report about *A*'s performance. If *B* has not sent any report about *A* before, *B* first sends *A*'s IP address to a random proxy (which is *R* in the figure). *R* then searches in the tree to identify the proxy whose range covers *A*'s IP address (*T* in this case). *T* then sends a response message to *B* as well as its certificate of trustworthiness (issued by a trusted certification authority). After *B* verifies the trustworthiness of *T*, it sends its report about *A* to *T*. In the following, *B* will directly send reports about *A* to *T*.

Two important issues in partially centralized approaches are efficient search and load balancing among multiple supernodes/proxies. First, each peer should be attached to a unique supernode or proxy. In MD, this is done by KaZaa's built-in mechanism. If a P2P network does not have such a built-in mechanism, this is not easy. Suppose each proxy is responsible for a certain range of peers. Given any peer in the system, we need to quickly identify the proxy responsible for it (e.g., for reputation update or query). If the number of proxies is small, simple flooding can be used for search. Otherwise, a more complicated overlay structure should be built among proxies (e.g., the binary search tree in [5]). Second, loads for reputation management should be evenly distributed among supernodes/proxies. MDA does not consider this issue as it uses the KaZaa built-in mechanism to attach peers to supernodes. In [5] a dynamic load redistribution method has been proposed to balance loads among proxies.

Compared to centralized approaches, partially centralized approaches have significantly improved system scalability. However, in order to serve a large P2P network, a large number of supernodes or proxies may be needed, which leads to high implementation and maintenance costs.

As compared to centralized approaches, partially centralized approaches have significantly improved system scalability. However, in order to serve a large P2P network, a large number of supernodes or proxies may be needed, which leads to high implementation and maintenance costs.

STRUCTURED OVERLAY

Another class of approaches uses distributed hash table (DHT) to store and search for peer reputation. In DHT each peer is assigned a unique peer ID, and each object is hashed to a key in the same space of peer IDs. The peer with ID equal to the hashed key is responsible for storing the location of the object (or the object itself). With a hashed key of an object, a query for the object is routed through peers in DHT to the peer that is responsible for the object. Each peer in DHT maintains a routing table for routing messages.

We take PeerTrust as an example [6]. It adopts P-Grid as the underlying DHT network. It also uses a system-wide hash function `Hash`, which maps one peer ID to another. Suppose that peer *p* has an ID, *ID(p)*. Whenever *p* has a transaction with another peer, *q*, *p* generates a report about *q* and sends it to the peer with ID `Hash(ID(q))` through DHT routing. The reputation of *q* is then stored and maintained at the peer with ID `Hash(ID(q))`, which is called the reputation manager of *q*. Queries of a peer's reputation are also forwarded to its reputation manager through DHT routing. In this way, peer reputation is distributedly stored in the system.

This approach has several advantages. First, peer reputation is distributedly stored and computed at the reputation managers. There is no need for a central server or supernodes. Second, a peer's reputation manager is determined by a universal hash function, which cannot be selected by the peer itself. This reduces the possibility of collusion between a peer and its reputation manager.

However, this approach has some security concerns. First, reputation managers may misbehave by providing false or random data when answering a query. Majority voting has been used to address this. That is, a DHT network can be configured to have multiple replicas responsible for the same key, or multiple hash functions can be used to map each peer to multiple reputation managers [6]. When a peer searches for the reputation of another peer, it finds all the replicas responsible for the key and uses a voting scheme to compute the final result. However, voting cannot guarantee obtaining the correct decision and does not completely address the problem. As shown in [11], simple collusion can seriously affect the result of voting. Second, a reputation report or query is delivered between its generator and the reputation manager by DHT routing. A malicious peer in the delivery path may modify, intercept, or discard the report or query. PeerTrust has proposed to encrypt messages in order to prevent data modification during delivery [6]. But it cannot prevent data discarding during routing. In summary, DHT-based approaches cannot guarantee secure reputation computing and delivery.

Furthermore, DHT has its own limitations. Since peers are highly dynamic in P2P networks, a reputation manager may unexpectedly leave the system. Then the data maintained by it are no longer available. In addition, load balancing mechanisms that abide by DHT storage and routing methods are complicated, especially in dynamic networks. DHT also has its own security threats and vulnerabilities, and there are many targeted attacks on its routing scheme, data placement scheme, IP mapping scheme, and so on.

UNSTRUCTURED OVERLAY

XREP uses a polling algorithm to choose reliable resource in Gnutella-like file sharing networks [14]. It consists of four operations: resource searching, vote polling, vote evaluation and resource downloading (Fig. 2). The first operation is similar to searching in Gnutella. A peer broadcasts to all its neighbors a `Query` message containing the search keywords. When a peer receives a `Query` message for which it has a match, it responds with a `QueryHit` message, as shown in Fig. 2a. In the next operation, upon receiving `QueryHit` messages, the query initiator selects the best matching resource among all possible choices. It then polls other peers using an encrypted `Poll` message to enquire about their opinion of the selected resource or the resource provider. In XREP each peer maintains information on its own experience with the resource and other peers. Upon receiving a `Poll` message, each peer checks its experience data. If there is any information about the resource or the provider indicated by the `Poll` message, the peer sends its vote to the polling peer with an encrypted `PollReply` message, as shown in Fig. 2b.

In the third operation the polling peer collects a set of votes and evaluates the votes. It first decrypts the votes and discards corrupt ones. Then it analyzes voters' IPs and detects cliques of dummy or controlled votes. After that, it randomly selects a set of votes and directly contacts the voters with a `TrustVote` message. Each contacted voter is required to send a `VoteReply` message for vote confirmation. This

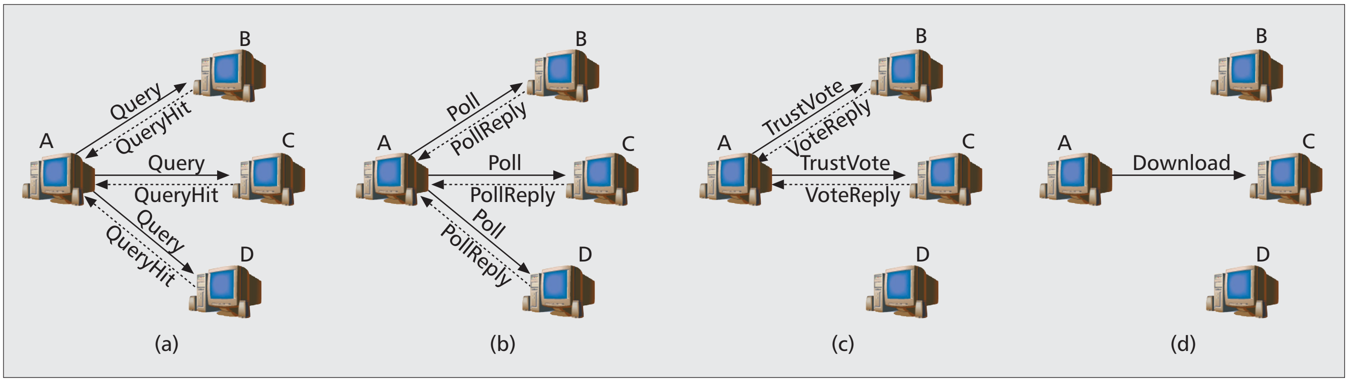

Figure 2. *Operations in XREP: a) resource searching; b) vote polling; c) vote evaluation; d) resource downloading.*

forces attackers to pay the cost of using real IPs as false witnesses. After this checking process, the polling peer can obtain the reputation of the resource or provider. Based on the reputation value, the polling peer can either download the resource, as shown in Fig. 2d, or discard the resource and repeat the voting process on another resource.

Approaches based on unstructured overlays have similar limitations to DHT-based ones. Messages may be intercepted or blocked during transmission, and voting is vulnerable to collusion among peers. Therefore, no secure reputation computing or delivery can be guaranteed. Furthermore, searching or voting on an unstructured overlay is based on flooding, which incurs heavy traffic in the network. For example, in XREP `Poll` messages are broadcast throughout the network each time a peer needs to find out the reputation of a resource or a provider.

COMPARISONS

We compare the above reputation query techniques in Table 1 and elaborate on the results below.

A centralized approach requires a central server for reputation storage, and a partially centralized approach relies on supernodes or predeployed proxies. On the contrary, approaches based on structured and unstructured overlays rely on peers to manage reputation and do not require additional facilities. Specifically, in DHT-based approaches a peer's reputation is maintained at its reputation manager, which is computed by a universal hash function. In approaches based on unstructured overlays, peers often locally hold the reputation of their transaction partners.

Based on different storage mechanisms, the approaches have different reputation search methods. In a centralized approach a reputation query is directly sent to the server. In a partially centralized approach a query is first sent to a supernode, which forwards the query to the target supernode. In a DHT-based approach DHT routing is used to route queries. In an approach based on unstructured overlays, flooding is often used, which may consume much network bandwidth.

Among these approaches, the centralized one has the poorest scalability, while the DHT-based one is the most scalable. The partially centralized approach has better scalability than the centralized one, but still relies on predeployed proxies or supernodes and is not fully scalable. The approach based on unstructured overlays does not need any central component; however, it is not as scalable as the DHT-based one because of its high bandwidth consumption in reputation search.

The centralized and partially centralized approaches are robust to peer dynamics. In these approaches reputation values are stored at a server or supernodes, which are often highly stable. In the DHT-based approach the leaving of a reputation manager will lead to the loss of data stored at it. Fortunately, DHT itself has some mechanisms to keep high data availability under peer churn. In the approach based on unstructured overlays there is little protection against data loss due to peer leaving. It may encounter high data loss in the presence of peer churn.

Regarding security, the centralized and partially centralized approaches are the most secure if assuming the server and supernodes are fully trustworthy. In these approaches reports or queries are directly sent to the server or supernodes, and there are no third-party peers in delivery paths. On the contrary, the approaches based on structured or unstructured overlays cannot guarantee secure reputation computing or delivery. In these approaches a reputation maintainer may be malicious and provide forged data, and a delivery path may contain malicious peers. Although there are many methods for improving system security (e.g., encryption/decryption or voting), none of them can guarantee 100 percent security.

CONCLUSION

In this article we investigate two key issues in P2P reputation systems, reputation estimation and query. We discuss representative examples in the literature and compare them from multiple aspects. There are many other research issues in P2P reputation systems, such as anonymity. In many applications, users may only be willing to participate if a certain amount of anonymity is guaranteed. But most existing reputation systems have sacrificed anonymity in order to provide secure underlying protocols, where each peer holds a unique certificate, and

Approach	Deployment requirement	Reputation storage	Reputation query	Scalability	Adaptation to peer dynamics	Security		
						Trustworthiness of reputation maintainer	Message modification in transmission	Message discarding in transmission
Centralized	A central server	The server	Direct server access	Low	High	Full	No (no overlay relay)	No (no overlay relay)
Partially centralized	A set of supernodes or proxies	Supernodes or proxies	Search among supernodes or proxies	Medium	High	Full	No (no overlay relay)	No (no overlay relay)
Structured overlay	No	Peers (computed by a hash function)	DHT search	High	Medium	No guarantee	No (addressed by encryption)	Possible
Unstructured overlay	No	Peers (e.g., transaction partners)	Flooding	Medium	Low	No guarantee	No (addressed by encryption)	Possible

Table 1. *Comparisons between various reputation query techniques.*

peers use the certificates to authenticate each other. Other interesting issues may include analyzing security threats and studying reward/punishment mechanisms.

References

[1] J. Liang *et al.*, "Pollution in P2P File Sharing Systems," *Proc. IEEE INFOCOM*, 2005.

[2] S. Marti and H. Garcia-Molina, "Taxonomy of Trust: Categorizing P2P Reputation Systems," *Comp. Net.*, vol. 50, no. 4, Mar. 2006, pp. 472–84.

[3] S. Ruohomaa, L. Kutvonen, and E. Koutrouli, "Reputation Management Survey," *Proc. IEEE ARES '07*, Apr. 2007, pp. 103–11.

[4] L. Mekouar, Y. Iraqi, and R. Boutaba, "Peer-to-Peer's Most Wanted: Malicious Peers," *Comp. Net.*, vol. 50, no. 4, Mar. 2006, pp. 545–62.

[5] X. Jin, Q. Xia, and S.-H. G. Chan, "Building a Monitoring Overlay for Peer-to-Peer Streaming," *Proc. IEEE GLOBECOM '06*, Nov. 2006.

[6] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-based Trust for Peer-to-Peer Electronic Communities," *IEEE Trans. Knowledge Data Eng.*, vol. 16, no. 7, July 2004, pp. 843–57.

[7] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," *Proc. WWW '03*, 2003, pp. 640–51.

[8] R. Sherwood, S. Lee, and B. Bhattacharjee, "Cooperative Peer Groups in NICE," *Comp. Net.*, vol. 50, no. 4, Mar. 2006, pp. 523–44.

[9] Z. Despotovic and K. Aberer, "Maximum Likelihood Estimation of Peers Performance in P2P Networks," *Proc. P2PEcon '04*, June 2004.

[10] L. Mui, M. Mohtashemi, and A. Halberstadt, "A Computational Model of Trust and Reputation," *Proc. IEEE HICSS '02*, Jan. 2002, pp. 2431–39.

[11] Z. Despotovic and K. Aberer, "P2P Reputation Management: Probabilistic Estimation vs. Social Networks," *Comp. Net.*, vol. 50, no. 4, Mar. 2006, pp. 485–500.

[12] D. Fudenberg and D. Levine, "Reputation and Equilibrium Selection in Games with a Patient Player," *Econometrica*, vol. 57, no. 4, 1989, pp. 759–78.

[13] S. Jun, M. Ahamad, and J. Xu, "Robust Information Dissemination in Uncooperative Environments," *Proc. IEEE ICDCS '05*, June 2005, pp. 293–302.

[14] E. Damiani *et al.*, "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks," *Proc. ACM CCS '02*, Nov. 2002, pp. 207–16.

Biographies

XING JIN [M] (xing.jin@oracle.com) received his B.Eng. degree in computer science and technology from Tsinghua University, Beijing, China, in 2002, and his Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology (HKUST), Kowloon, in 2007. He is currently a member of technical staff in the Systems Technology Group at Oracle, Redwood Shores, California. His research interests include distributed information storage and retrieval, peer-to-peer technologies, multimedia networking, and Internet topology inference. He is a member of Sigma Xi and the IEEE Communications Society Multimedia Communications Technical Committee.

S.-H. GARY CHAN [M] (gchan@cse.ust.hk) received his B.S.E. degree (Highest Honor) in electrical engineering from Princeton University, New Jersey, in 1993, with certificates in applied and computational mathematics, engineering physics, and engineering and management systems, and his M.S.E. and Ph.D. degrees in electrical engineering from Stanford University, California, in 1994 and 1999, respectively, with a minor in business administration. He is currently an associate professor with the Department of Computer Science and Engineering, HKUST, and an adjunct researcher with Microsoft Research Asia, Beijing. His research interests include multimedia networking, peer-to-peer technologies and streaming, and wireless communication networks. He is a member of Tau Beta Pi, Sigma Xi, and Phi Beta Kappa. He served as a Vice-Chair of the IEEE Communications Society Multimedia Communications Technical Committee from 2003 to 2006. He was a Guest Editor for *IEEE Communications Magazine*, Special Issue on Peer-to-Peer Multimedia Streaming (2007), and *Springer Multimedia Tools and Applications*, Special Issue on Advances in Consumer Communications and Networking (2007). He was Co-Chair of the Multimedia Symposium for IEEE ICC 2007. He was Co-Chair of the workshop on Advances in Peer-to-Peer Multimedia Streaming at ACM Multimedia 2005, and the Multimedia Symposia for IEEE GLOBECOM 2006 and IEEE ICC 2005.