

Reputation Management for DHT-based Collaborative Environments*

Natalya Fedotova, Luca Veltri
University of Parma, Italy

Abstract — This article addresses a problem of integration of reputation management mechanisms and lookup processes in DHT-based peer-to-peer (P2P) networks with collaborative nature. We present a possible scenario of application of some reputation techniques to lookup and data retrieval processes in a P2P network based on the popular Kademlia algorithm.

Unlike most of existing reputation management systems for DHT-based P2P environments, we don't use a number of successful downloads as the main instrument to evaluate a trustworthiness of peers, but we propose to consider any type of interactions between nodes. This choice is motivated by the fact that file sharing is not a unique service supported by DHT-based P2P systems. There are other application areas for P2P (e.g. collaborative and distributed computing) where it is important to consider such aspects as a risk factor, a "stay on-line" time or a number of requests without reply.

Keywords-component — Peer-to-peer, Distributed Hash Tables, reputation mechanisms, security.

I. INTRODUCTION: BACKGROUND AND MOTIVATION

Distributed Hash Tables (DHT) are building block for realization of many peer-to-peer (P2P) applications and services, such as distributed file systems, file sharing, instant messaging, collaborative and distributed computing. DHTs provide a very efficient instrument for lookup, routing, data storage and sharing mechanisms [1,2].

However, DHT-based P2P networks represent a particular environment vulnerable to some specific threats and attacks. Generally, these attacks are caused by malevolent behaviour of some network nodes and they are aimed at routing and lookup processes [3]. The self-organizing nature of DHTs enables some countermeasures against several effects of such malicious activity. For instance, in the case of iterative lookup (Kademlia, Chord) incorrect lookup routing attacks can be avoided by checking the progress of lookup at each step; storage and retrieval attacks can be prevented by replication of files using multiple hash functions; Sybil attacks cannot be excluded in such environment, but a lookup efficiency can be improved by parallel routing (issuing several lookup requests at a time) [4].

Anyway, these countermeasures are "momentary" solutions that don't resolve the problem of polluting routing tables by malicious contacts. Therefore, it is reasonable to use gained experiences to render a network community more secure. Once a malevolent activity has been detected, reputation techniques should be applied in order to "clear" routing tables from contacts that have evinced malicious or inconsistent behaviour to avoid them in the future.

Recently, a number of reputation management techniques for P2P networks has been proposed by different researchers. In our previous work [5] we presented a detailed analysis of applicability of several existent reputation evaluation techniques as protection from some types of attacks in DHT-based P2P networks. The analyzed techniques are not designed for DHT-based P2P networks, so none of them represents a universal solution for such systems. At the same time, different reputation management instruments used by these techniques could be a quite effective in some particular cases in a DHT environment.

Normally, reputation techniques proposed specifically for DHT-based networks are heavily based only on evaluation mechanisms of successful and unsuccessful downloads, since they regard only file sharing P2P applications [6,7]. However, as we have mentioned above, P2P technology also supports instant messaging, collaborative applications, distributed computing, etc.

In the last few years P2P systems have been successfully used for sharing computation under various distributed computing projects like FightAIDS@Home, Genome@Home, Seti@Home, United Devices Cancer Research Project and others [8]. These projects represent a public-resource computing that relies on personal computers with excess capacity, such as idle CPU time, to resolve some complex research problems. Public-resource computing is an aspect of the peer-to-peer paradigm, even if it uses a grid technology to realize its tasks.

Currently, such systems approach a DHT nature. Some steps of computational processes become completely independent from central servers: calculations results of some node are stored in the network and retrieved by a successor that use them in its own part of the task; if a peer leaves a network while processing a work unit, the work unit is eventually sent to another peer that becomes responsible for it (like in DHT data storage systems when a node becomes responsible for resources of some failed node if their identifiers are considered as the closest to each other).

* This work has been partially supported by the Italian Ministry for University and Research (MIUR) within the project PROFILES under the PRIN 2006 research program.

Another type of systems that use DHT principles are collaborative applications for data storage and editing by several geographically distant work groups. Such systems should provide a rapid and secure data exchange between different system units and possibility of team-work in real-time and transparent mode.

The systems described above represent active distributed collaborative environments, where every interaction between peers is important and, as L. Lamport said, “the failure of a computer you didn’t even know existed can render your own computer unusable” [20]. So, in these systems a number of successful downloads cannot be a sufficient instrument for reputation management. In this case parameters characterizing a community context (risk factor, number of lookup requests without response, number of join and leaves for a node, off-line status time) are also required.

In this work we consider the possibilities of incorporation of reputation mechanisms in DHT-based routing and lookup processes. We propose a solution that combines different instruments offered by some reputation management techniques analyzed in our previous work, applying them to lookup and data retrieval processes in a P2P network based on the popular Kademlia algorithm.

II. PREVIOUS WORK

Taking in consideration some particularities of deploying reputation mechanisms in a DHT setting, we introduced the following applicability criteria for reputation mechanisms:

1. technical realizability in overlay networks;
2. availability of individual reputation evaluation instruments.

The first criterion is critical for reputation evaluation technique with credential and policy elements [9]. In credential and policy based trust management systems peers use a set of credentials and policies to determine whether some unknown peer can be trusted or not. Obviously, in this case the presence of some sort of certification authority is required. Such techniques often require a central server for storing and distributing reputation information. Therefore, credential and policy based mechanisms are to be applied in centralized systems with a hierarchical structure. In the case of completely distributed DHT networks we need self-policing techniques providing mechanisms that can be realized in decentralized environment using means proposed by an overlay infrastructure, like pure reputation mechanisms.

The second criterion is represented by some different parameters, such as:

- possibility to provide recommendations;
- possibility to “weigh” recommendations, i.e. recommendations from different peers have different levels of trustworthiness;

- responsibility for the behavior of recommended entities;
- evaluation of a community context (an average level of vulnerability of the network environment and a level of cooperation between peers);
- incentives for feedback compilation.

So, a single node should be provided with all necessary instruments to analyze and independently evaluate reputation and reliability of other peers.

Table 1 summarizes results of our comparative analysis taking into account data regarding all the basic characteristics of the analyzed reputation models.

Concerning the first applicability criterion, most of the analyzed techniques can be subsumed under the reputation based category. NICE [15] and DCRC/CORC [19] realize some credential and policy elements: digital signatures of cookies in NICE, peer identification by “reputation computation agents” (RCAs) using public key in DCPC/CORC. Poblano [14] and XREP [16] also involve some mechanisms with a centralized nature. This fact represents some difficulties for application of these techniques to completely distributed DHT-based networks.

As to the second criterion, all these techniques have different completeness degrees. PeerTrust [12] and Fuzzy Model [11] represent the most complete techniques, as they realize almost all possible mechanisms for evaluation of a peer’s trustworthiness.

We can conclude that none of these techniques in its pure form represents a suitable solution for DHT-based P2P networks. The solution we present is a combination of different instruments offered by the analyzed models. We propose to apply each of these instruments when it is considered as the most efficient one for a certain situation.

III. INTEGRATION OF REPUTATION MECHANISMS AND LOOKUP PROCESSES

A. Lookup and reputation mechanisms

In DHT-based systems (CAN, Chord, Pastry, Kademlia) a group of distributed hosts collectively manages a mapping from keys to data values, without any fixed hierarchy, and with a very little human assistance [1]. The base of such systems is a routing table-based lookup service.

If a lookup initiator doesn’t find in its routing table an ID of a node responsible for some desired key, it sends a lookup query to a node it considers the “closest” to this key among all its contacts. The last one should reply with an IP address of the next hop (iterative lookup) or forward the query to the next “closest” node (recursive lookup). Applying opportune

TABLE I. APPLICABILITY ANALYSIS SUMMARY TABLE

Parameter Model	Reputation Value Scale	Possibility to Provide Recommendations	Possibility to “Weigh” Recommendations	Responsibility for the Behavior of Recommended Entities	Transaction consideration (type, quantity, data, dimensions)	Evaluation of a community context	Incentive to feedback compilations	Credential and policy elements
Supporting Trust in Virtual Communities	4 possible levels	Yes	Yes	No	No	No	No	No
Peer Trust	Normalized from 0 to 1	Yes ¹	Yes ²	No	Yes	Yes	Yes	No
Personalized Trust Model	Normalized from 0 to 1	Yes	No	No	No	No	No	No
Fuzzy Model for context-dependent Reputation	Normalized from 0 to 1	Yes	Yes	Yes	Yes ³	Yes	No	No
Poblano	6 possible levels [-1; 4]	No	No	No	No	No	No	No
NICE	Normalized from 0 to 1	Yes ⁴	No	No	No	No	No	No
XREP	Binary ⁵	Yes ⁶	No	No	No	No	No	Yes
Sporas and Histos	From 0 to 3000 exclusive	Yes ¹	Yes ²	No	No	No	No	No
Beta	Normalized from 0 to 1	Yes ¹	Yes ²	No	No	No	No	No
DCRC/CORC	Non-negative	No	No	No	Yes	No	No	Yes

1 – feedbacks compiled by other users are considered

3 – only the events’ number is considered

5 - a binary value scale is not obligatory, it should just be indicated an interval of values used

2 – regards an entity compiling a feedback

4 – recommendations regarding some determined peer

6 – in the form of a vote

mechanisms for verifying lookup progress, the querying node can also make a conclusion about “honesty” of the nodes participating in the lookup process, assigning to them the corresponding reputation values. Analogically, a node that honestly shares its resources with other nodes gets reputation “points”, and a node denying the existence of data it is responsible for (storage and retrieval attacks), loses them.

So, each node of the network after every contact with another node assigns a new reputation value to the contacted peer depending on the interaction results. All the assigned reputation values are to be stored by the querying node and should be consulted before contacting corresponding nodes again. Moreover, these reputation values are used as recommendations that each node exchanges with others, when responding to an iterative lookup query.

It is important to note that in the case of recursive lookup it is problematic to apply verifying mechanisms at each hop. So, to avoid forwarding queries to malicious peers, nodes should also check reputation values of potential “delegates” when choosing next step of lookup routing instead of simply considering their “closeness” to a desired resource.

If some node is declared untrustworthy on the base of gained reputation points, it should be ignored by “good” peers during the future lookup and routing processes, i.e. it should be simply considered and treated as a node that has left the network (a node with off-line status). This mechanism doesn’t affect the stability of a DHT-based overlay due to the multiple data replication and keys reassignment scheme used in such environment. All data replicas are uniformly distributed on the network between several predefined responsible nodes. So, it is always possible to find another node enabled to provide the same resource.

B. Proposed scenario

In this section we present a possible scenario of application of some reputation mechanisms provided by the analyzed techniques.

As an environment for our scenario we chose a network based on Kademia DHT protocol, widely used by a number of P2P platforms (eDonkey, BitTorrent, etc). The proposed combination of the reputation instruments includes:

- risk evaluation method provided by PET model [13];

- resources and server repositories from XREP model [16];
- debit-credit based reputation computation model (DCRC) [19].

This scenario represents a situation in which a peer joins a network the first time and initiates a lookup process for a data file with a certain ID. Since it is a new node for this network, it has no idea about trustworthiness of other nodes.

As reputation is an accumulative value, it is not possible to evaluate someone's reputation just after the first contact. However, it is possible to define a level of vulnerability of the network environment on the base of results of the first experiences using the appropriate instrument offered by PET.

PET model derives the trustworthiness value T from two components: reputation factor R_e and risk factor R_i with different weights (incidence) W_{R_e} and W_{R_i} respectively. The trustworthiness value is defined as follows:

$$T = W_{R_e} R_e + W_{R_i} R_i = (\alpha, 1 - \alpha) \times (R_e, (1 - R_i))^T, 0 \leq \alpha \leq 1 \quad (1),$$

where $W_{R_e} = \alpha$ and $W_{R_i} = 1 - \alpha$, and the values of T , R_e and R_i are all from 0 to 1.

In our scenario, after the first contact $T = R_i$, as the node doesn't have sufficient data to evaluate the reputation factor, but it is necessary for it to define a level of vulnerability of the network it has joined. The risk value in PET model is calculated by the formula:

$$R_i = \frac{\sum_{i=B,N,L} (N_i \times h(i))}{h(B) \times \sum_{j=G,B,N,L} N_j} \quad (2),$$

where G , B , N , L are four levels of quality (Q) of services provided by a peer (it is applicable to any type of interaction between peers: elaboration and forwarding of queries during a lookup process, providing resources, etc.):

- G – Good,
- L – Low Grade,
- N – No Response,
- B – Byzantine Behaviour;

N_i is the number of services (interactions) provided with quality i ; h is a map function from Q to a score for one interaction between nodes, i.e. it shows how many reputation points a node has gained or lost at the end of one interaction:

$$h(Q) = \begin{cases} S_1, Q = G, S_1 > 0 \\ S_2, Q = L, S_2 < 0, \text{ and } |S_2| > S_1 \\ S_3, Q = N, S_3 < S_2 \\ S_4, Q = B, S_4 < S_3 \end{cases} \quad (3).$$

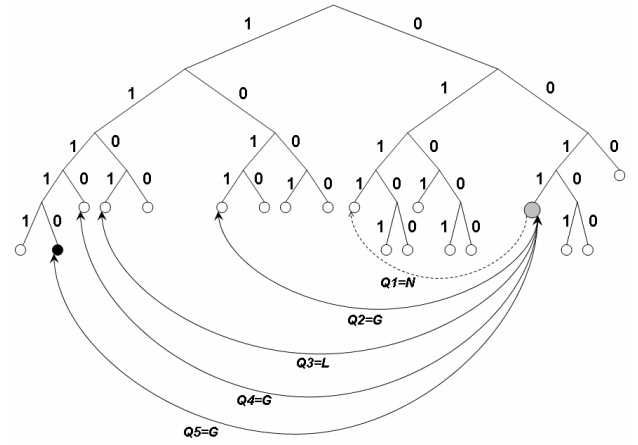


Fig. 1. Look-up process based on Kademia algorithm in terms of PET model

So, we can see that misbehaving reduces a reputation value faster than a good behaviour increases it. *No Response* is considered as a bad behaviour here. It helps to avoid too frequent and long leaving in P2P networks: it is not convenient for a peer to be off-line for a long period, because all requests sent to the node during its absence will rest without response, significantly reducing its reputation. It is explained by a high importance of the cooperation component for collaborative environments: a node that doesn't participate in routing and lookup mechanisms by processing and forwarding requests from other nodes cannot be positively evaluated by the community.

The risk value R_i is normalized to the worst case, that represents a situation when all services received by a peer in a certain time interval are Byzantine services. In our example we assign to S_1 , S_2 , S_3 and S_4 the values 1, -2, -3 and -4 respectively.

In Kademia, identifiers of nodes consist of 160 bit. Here we present a simplified model of Kademia's binary tree with a little number of sub-trees and leaves.

As seen in figure 1, the node with prefix 0011 initiates the process of look-up for some resource, sending FIND_VALUE RPC to two nodes residing in two different sub-trees with prefixes 0111 and 1011. The first node doesn't respond (the one-directional dashed arrow). The second node returns a triple <IP address, UDP port, Node ID> for the node 1101. Applying the look-up progress verification mechanism, the requestor concludes that the Node ID sent by the second node is really closer to the key.

According to PET model our requestor assigns to this contact the quality value G , and to the first contact – the value N . Then, it sends FIND_VALUE RPC to node 1101, that returns a triple containing Node ID 1110, but with a little delay. In its turn, 1110 replies with Node ID 11110, that stores the desired data. 11110 returns the stored value. Having controlled at each step the look-up progress, the requestor assigns to nodes 1101, 1110 and 11110 the quality values L , G and G respectively.

The risk value in our case is:

$$R_i = \frac{N_L \times S_2 + N_N \times S_3}{S_4 \times (N_G + N_L + N_N)} = \frac{1 \times (-2) + 1 \times (-3)}{-4 \times (3 + 1 + 1)} = 0,25$$

Information, regarding the quality of resources and the reliability of the peers obtained during this look-up process, could be stored in the resource and servent repositories according to the XREP model [16]. The repositories represent two tables with the following data structures:

- $\langle resource_id, value \rangle$;
- $\langle servent_id, num_plus, num_minus \rangle$.

The first table associates with each resource identifier (resource_id) a value that defines the quality of the corresponding resource. XREP's authors don't precise a type of data representing quality values allowing the liberty of interpretation: it may be a numeric value, or it may be simply defined as good or bad.

The data of a servent repository contain IDs (servent_id) of contacted peers and corresponding numbers of successful (num_plus) and unsuccessful (num_minus) transactions effectuated with these peers. As in our case we use DCRC model, it makes sense to substitute the num_plus and num_minus with data regarding the total number of uploaded and downloaded megabytes of content for a certain peer, e.g. mb_up and mb_down.

To calculate the Query-Response Credit (QRC) parameter, for each contacted node the total number of queries addressed to a node and a number of queries processed and forwarded by a node should be stored. Adapting the data structure of the servent repository to our case, the latter table becomes:

$\langle servent_id, mb_up, mb_down, num_query, num_reply \rangle$.

According to the previous example, the node 0011 stores the following data in its servent repository:

servent_id	mb_up	mb_down	num_query	num_reply
0111	0	0	1	0
1011	0	0	1	1
1101	0	0	1	1
1110	0	0	1	1
11110	20	0	1	1

In terms of technical realization this mechanism represents a quite simple solution: a simple counter is essentially required.

According to DCRC model [19], a reputation value of a peer is defined by credits it gains or loses during a certain period interacting with other peers. The total number of reputation points can be calculated by the following formula:

$$R_e = n_q \times QR + \sum_l n_u \times UC_l - \sum_m n_d \times DD_m + t_f \times SC \quad (4),$$

where:

QR is the number of points gained by a node for each processed query;

n_q is the total number of queries processed by a peer;

n_u is the number of uploads facilitated by a peer;

n_d is the number of downloads performed by a peer;

t_f is the predefined time factor (in hours) that determines a time interval during which the described interactions have been performed;

UC_l and DD_m are the upload credit and download debit for files l and m respectively.

In its turn the UC is defined as follows:

$$UC = \frac{s}{f} \times \frac{bw}{b} \quad (5),$$

where:

s is the size of an uploaded file (in megabytes);

bw is the bandwidth available (in megabytes);

f is the file size factor that determines how many megabytes of data transfer increases the reputation score by a unit;

b is the bandwidth factor that classifies peers on the base of bandwidths they have at disposal.

The DD for a download of a resource of size s is given by:

$$DD = \frac{s}{f} \times \frac{bw_i}{b} \quad (6),$$

where bw_i is the bandwidth of a peer i from which a download is performed.

The Sharing Credit (SC) for a peer that shares n files during some predefined time interval is given by:

$$SC = \sum_j^n \frac{s_j}{f} \quad (7),$$

where, s_j is the size of j th file.

Let's calculate the reputation value of the node 11110 using the data stored by 0011 at the repository after the considered look-up process. Let the size s of the file downloaded by the node 0011 from 11110 be 20 MB, the file size factor $f=2$ MB, the available bandwidth $bw=6$ MB, and the bandwidth factor $b=2$ MB. Then, the UC of 11110 after this interaction is:

$$UC = \frac{20}{2} \times \frac{6}{2} = 30$$

For simplicity, let the number of points gained by a peer for each query processed $QR=1$. Let's suppose that the total size of the resources shared by 11110 node is 500 MB and the predefined time interval is 1 hour. Hence, $SC=250$.

Then, at $n_q = I$ and $DD = 0$, the total number of reputation points gained by the peer 11110 after the interaction in question is:

$$R_e = QRC_{tot} + UC_{tot} - DD_{tot} + SC_{tot} = 281.$$

IV. CONCLUSION

The results of the applicability analysis from our previous work show that none of the considered reputation techniques in its pure form represents a universal and suitable solution for DHT-based P2P environment. So, we have proposed a solution that consists in application of a combination of different reputation mechanisms provided by some analyzed techniques. It is motivated by the fact, that existing reputation evaluation techniques designed specifically for DHT-based networks mainly for file-sharing P2P applications are based only on a number of successful and unsuccessful downloads. However, although this mechanism can be considered as an indicative component of reputation evaluation, it cannot be a sufficient instrument for reputation management in collaborative environments (e.g. distributed computing and data editing).

In collaborative environments it is also very important to consider possible risks and various parameters regarding a community context, because the cost of a mistake, caused by malicious activity in such networks is incomparably higher than in file-sharing systems. Just a few unreliable peers that have not been discarded from routing tables in time can interrupt a long chain of calculations.

In this work we have presented a possible scenario of application of the proposed solution to a P2P network based on Kademlia DHTs, extracting some reputation evaluation instruments from the models analyzed before and adapting them to particularities of DHT-based environment. The proposed solution represents an individual mechanism of reputation management for a single peer based on its own experience.

We can conclude that, the risk calculation method used in PET model helps to define a level of vulnerability of an unfamiliar environment, while DCRC technique represents a an objective method of reputation evaluation based on points gained by a peer due to its collaboration with the community. In its turn, the repository mechanism is a simple and efficient solution for systematization of the data necessary for reputation points calculation.

All of these instruments can be easily adapted to DHT-based environment. This example demonstrates that the mechanisms used in our scenario successfully complement each other, even if they were "extracted" from three different reputation evaluation models.

We are currently working toward a software implementation of the described scenario and other reputation mechanisms in Kademlia network.

REFERENCES

- [1] D. Doval, D. O'Mahony, "Overlay Networks: A Scalable Alternative for P2P", IEEE Journal on Internet Computing, Vol.7, No.4, pp. 79-82, August 2003
- [2] H. Balakrishnan, M. F. Kaashoek et al., "Looking Up Data in P2P Systems", Communications of the ACM, Vol. 46, No. 2, pp.43-48, Feb. 2003.
- [3] E. Sit, R. Morris, "Security considerations for Peer-to-Peer Distributed Hash Tables", in Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS'02), Cambridge, Massachusetts, March 2002
- [4] D. Stutzbach, R. Rejaie, "Improving Lookup Performance over a Widely-Deployed DHT", in Proceedings of 25th IEEE International Conference on Computer Communications, INFOCOM'06, April 2006
- [5] N. Fedotova, M. Bertucci, L. Veltri, "Reputation Management Techniques in DHT-based Peer-to-Peer Networks", in Proceedings of ICIW'07, May 2007
- [6] S. D. Kamvar, M.T. Schlosser, H. Garsia-Molina, "The Eigen Trust Algorithm for Reputation Management in P2P Networks", in Proceedings of the 12th International World Wide Web Conference, May 2003
- [7] S. Y. Lee, O-H. Kwon, J. Kim, S. J. Hong, "A Reputation Management System in Structured Peer-to-Peer Networks", in Proceedings of 14th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprise (WETICE'05)
- [8] V. Martins, E. Pacitti, P. Valduriez, "Survey of data replication in P2P systems", research report, INRIA, December 2006
- [9] G. Suryanarayana, R.N. Taylor, "TREF: A Threat-centric Comparison Framework for Decentralized Reputation Models", ISR Technical Report UCI-ISR-06-2, January 2006
- [10] A. Abdul-Rahman, S. Hailles, "Supporting Trust in Virtual Communities", In Proceedings of the 33rd Hawaii International Conference on System Sciences, Maui, Hawaii, January 2000.
- [11] V. Grishchenko, "A fuzzy model for context-dependent reputation", Trust, Security and Reputation Workshop at ISWC 2004, Hiroshima, Japan.
- [12] L. Xiong, L. Liu, "A Reputation-Based Trust Model for Peer-to-Peer Ecommerce Communities", Proc. IEEE Conf. E-Commerce (CEC '03), June 2003.
- [13] Z. Liang, W. Shi, "PET: A Personalized Trust Model with Reputation and Risk Evaluation for P2P Resource Sharing", In Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.
- [14] R.Chen, W. Yeager, Poblano: A Distributed Trust Model for Peer-to-Peer Networks. Technical report, Sun Microsystems. <http://www.jxta.org/docs/trust.pdf>.
- [15] Lee, S., Sherwood, R., et al., "Cooperative peer groups in NICE", IEEE Infocom, San Francisco, USA, 2003.
- [16] Damiani, E., di Vimercati S., et al., "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks", 9th ACM Conference on Computer and Communications Security, Washington DC, 2002.
- [17] Zacharia, G. and Maes, P., "Collaborative Reputation Mechanisms in Electronic Marketplaces", 32nd Hawaii International Conference on System Sciences, Hawaii, 1999.
- [18] A. Josang, R. Ismail, "The Beta Reputation System", 15th Bled Electronic Commerce Conference, Bled, Slovenia, 2002.
- [19] M. Gupta et al., "A Reputation System for Peer-to-Peer Networks", Thirteenth ACM International Workshop on Network and Operating Systems Support for Digital audio and Video. Monterey, California, 2003.
- [20] <http://research.microsoft.com/users/lamport/pubs/pubs.html>