

**Università degli Studi di Genova**  
**Dottorato in Informatica**  
**(XX Ciclo)**

Thesis Proposal

**Design of a Reputation-based Incentive  
System for P2P Applications**

**Candidate:** Matteo Dell'Amico  
dellamico@disi.unige.it  
Dipartimento di Informatica e Scienze dell'Informazione  
Università di Genova

**Advisor:** Giovanni Chiola  
chiola@disi.unige.it  
Dipartimento di Informatica e Scienze dell'Informazione  
Università di Genova

December 2005

## Abstract

*Free riding*, the behaviour of exploiting resources contributed by other nodes while not sharing anything valuable, is a plague in many P2P networks. Its main cause is that sharing resources has a cost which is not, or not sufficiently, counterbalanced by benefits.

Incentive systems have been developed in order to reward fair users and punish free riders; the most successful existent implementations obey the principle of *direct reciprocation*: nodes give to other peers a service having a quality which is proportional to the service quality received from them.

The *indirect reciprocation* approach, which is based on evaluating peers based on their reputation (a judgement of past interactions with *any* node), is more powerful. In fact, in the common case of large networks, peer pairs have a low probability of having interacted before, and nodes can be judged more accurately when having access to more data.

The goal of this thesis is to develop an indirect reciprocation system designed to be secure and efficient enough to be implemented in large-scale networks, where nodes are free to create new identities at any time.

The main concerns will be finding good metrics for reputation and algorithms to calculate them efficiently, rewarding peers that give accurate information about their history, providing significant experiments to evaluate the obtained benefits, implementing the designed system in an existing or new P2P network.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Trust and Resource Allocation . . . . .	2
1.2	Distributed Reputation Evaluation . . . . .	3
<b>2</b>	<b>State of the Art</b>	<b>4</b>
2.1	A Game Theoretical Point of View . . . . .	4
2.2	Direct Reciprocative Approaches . . . . .	5
2.3	Indirect Reciprocative Approaches . . . . .	6
<b>3</b>	<b>Preliminary Results</b>	<b>8</b>
3.1	Neighbourhood Maps . . . . .	8
3.2	Estimation of Shortest Path . . . . .	9
3.3	Approximating PageRank . . . . .	12
<b>4</b>	<b>Planned Work</b>	<b>15</b>
4.1	Reputation Evaluation . . . . .	15
4.2	Security Issues . . . . .	16
4.3	Experimental Setup . . . . .	16
4.4	Timeline . . . . .	17
4.5	Planned Thesis Structure . . . . .	18
	<b>Bibliography</b>	<b>19</b>

# Chapter 1

## Introduction

### 1.1 Trust and Resource Allocation

A crucial problem in many network applications is determining how to assign *trust*. Users, in fact, are given access to resources (e.g., information, bandwidth, storage space, and/or CPU processing time) when they can be *trusted* to fairly use the application. This problem is even more important in P2P applications, where nodes are both consumers and service providers, and the overall performance of the system is thus dependent on the quality of service given by peers.

In traditional client-server applications, the server administrator usually manages a database describing who the trusted users are, and trust is granted to a user only after the server has authenticated her/him. In large-scale P2P applications, the scenario is completely different: a centralised administration of such a “trust database” would unfortunately deny many of the well-known qualities of decentralised applications, creating a single point of failure and a possible performance bottleneck.

In P2P applications, nodes often have a high degree of freedom regarding the amount of resources they can make available to other peers in the network. Usually, the node pays a small cost in sharing those resources, while peers using those resources will receive the benefit associated with the use of that resource.

In applications developed without regard to trust evaluation, giving each peer the same level of access to resources, there is indeed an incentive to *free riding*: a “selfish” behaviour in which nodes consume resources given by other peers, while not giving their available resources back to the peer community in return. Indeed, measurements done on the Gnutella [2] file-sharing network in 2000 [4] showed that nearly 70% of the nodes did not share any files, and that 50% of all responses were returned by the best 1% of the hosts. This phenomenon is well-known in economy and game theory as the “tragedy of the commons” [16], modelling overconsumption of commonly-owned resources.

An idea for solving this problem is introducing *reciprocity*. This principle is applicable whenever interaction between actors is repeated. According to this

principle, an entity has a more cooperative attitude towards peers that had a cooperative behaviour with itself in the past. In many cases, cooperating entities receive a higher benefit [5], and thus incentives cooperation for rational actors. Indeed, the reciprocity approach has been implemented in the field of P2P file-sharing networks, both in BitTorrent [11] and in EDonkey2000 [1]. The great success of this approach is witnessed by the fact that these two networks are by far the two most popular file-sharing applications; according to a study by the network monitoring appliance vendor CacheLogic, at the end of 2004 they accounted for almost 50% of the total Internet traffic [9, 10].

## 1.2 Distributed Reputation Evaluation

The reciprocity approach is useful under the assumption that two interacting nodes will interact again with a high probability. In a P2P network, this is true whenever networks are small, nodes usually interact with the same peers, and/or mutual interactions have a sufficiently large duration (e.g., two peers sharing complementary pieces of a big file). Indeed, there are many cases in which these assumptions do not hold, and the incentives to cooperation are thus lost.

The notion of *reputation* is introduced to solve this problem. It is an evaluation made about the global history of interactions of a peer in the whole network. When nodes evaluate interactions of a node with other peers in addition to the ones they had with themselves, incentives to cooperation are then restored [24].

The goal of the thesis is to design a system that uses reputation evaluation effectively in a P2P network, in order to significantly encourage cooperation, so that it can be used in a wide application domain. Simulation experiments will be conducted in order to evaluate the effectiveness of the system. This system is planned to be integrated in an existing P2P infrastructure.

The resulting system will be completely decentralised, thus avoiding single points of failure and performance bottlenecks. It will be applicable in situations where nodes have the possibility of arbitrarily creating new identifiers for them, thus making their past, possibly malicious, history unaccessible. Link analysis techniques [8] will be used in order to calculate metrics that are both significant and resilient to attacks.

## Chapter 2

# State of the Art

### 2.1 A Game Theoretical Point of View

The antinomy between egoistic behaviour and common good is one of the classic topics in Game Theory. As a simple example, let us consider an instance of the Prisoner's Dilemma (PD), in fig. 2.1.

The game is described using strategic notation<sup>1</sup>, and can be sketched in this way: player I chooses a row (identified by strategies  $C$  and  $D$ , *cooperate* and *defect*), while player II chooses a column (again, in this example, identified by  $C$  and  $D$ ). As a result of the game, both players receive a prize which is quantified by a numeric *payoff*. In this case, if both players cooperate (strategy  $C$ ), they both receive a payoff which is equal to 3, while if both defect ( $D$ ) they obtain a payoff of 1. If one player defects and the other one cooperates, then the defector obtains 5 and the cooperator obtains 0.

In this example, the “common good” can be identified as the  $C/C$  pair of strategies, where both players obtain a payoff of 3. It can be easily shown that rational players would instead choose the  $D$  strategy, since it would yield better payoffs no matter what the other player would choose. A formalisation of the “tragedy of the commons” situation mentioned in the Introduction leads to analogous results ([26], chapter 7).

In P2P environments, there is a similar situation: nodes can choose either to make their resources available to other nodes (cooperation) or to free ride (defection). As in the game theoretic situations seen so far, while the “social

---

<sup>1</sup>For an introduction to game theory, see e.g. [26], parts 1 and 2.

I\II	$C$	$D$
$C$	3,3	0,5
$D$	5,0	1,1

Figure 2.1: An instance of the Prisoner's Dilemma

optimum” occurs when all nodes cooperate, the strategy that will be chosen by selfish and rational players is defecting, as long as defectors enjoy the same service as the other nodes and avoid the cost of sharing resources.

Cooperation can be instead obtained in the Iterated Prisoner’s Dilemma. Rounds of this game are instances of the basic version of the Prisoner’s Dilemma; after each round the two players have a probability  $\delta$  of ending the game; otherwise, the game continues with a new round. In this framework, rational players have viable *reciprocative* strategies: they cooperate with players who had a sufficiently cooperative strategy with them in the past, while they defect with others. Since these strategies are based on repeated encounters between the same pairs of individuals, they are known as *direct reciprocative* strategies.

Reciprocative strategies are also successful in practical experimentation: in [5], an experiment was conducted with a tournament between strategies submitted by researchers, playing versus each other in the Iterated Prisoner’s Dilemma. Many sophisticated strategies were submitted, but the most successful one was the surprisingly simple Tit-for-Tat strategy: cooperate with players which cooperated in the last round, and defect with defectors.

In systems where transactions are asymmetric (i.e., one peer is behaving as a client and the other one as a server, and only the client has interest to a successful interaction), or they have a sufficient duration, they have a low probability of repeating their encounters. Direct reciprocative strategies thus become less useful. In this case, players have a *reputation* reflecting their behaviour history in interactions with other peers. In *indirect reciprocative* strategies, players are more willing to cooperate with other players that have a past history of cooperation. It has been experimentally shown that in many cases this kind of reciprocative strategies achieves good results [24, 22].

## 2.2 Direct Reciprocative Approaches

BitTorrent is a protocol for distributed transfer of large size files. A centralised *tracker* records which nodes are downloading the same file, and gives the information to all downloading clients. BitTorrent’s approach [11] is inspired, in principle, to the tit-for-tat strategy used in the Prisoner’s Dilemma. For each file in download, only four upload connections are kept receiving data (“unchoked”). Three of them are the connections that guarantee the best upload bandwidth, while the fourth one (“optimistic unchoke”) rotates through other peers with the hope of finding a better incoming bandwidth.

EDonkey2000 is a file-sharing network in which download requests from other peers are stored in a queue. The basic behaviour implies nodes having waited the longest time in the queue get the files. The eMule client uses a *credit system* as a direct reciprocative approach: the waiting time in the queue is multiplied by a function of the uploaded and downloaded total size between the two peers. Of course, this criterion is mostly helpful when more peers are downloading the same file, creating a dynamics which is similar to what happens in BitTorrent.

## 2.3 Indirect Reciprocal Approaches

It is apparent that indirect reciprocal approaches can yield better results in more general cases. Anyway, there are issues that are worth taking into consideration.

**No Global Knowledge** In networks having large size and/or a great number of interactions, it is unfeasible for peers to keep updated data about interaction history. Thus, either methods based on locally available knowledge or the use of decentralised data structures are preferable; standard solutions for the latter case are DHTs (distributed hash tables) such as Chord [28].

**Cheap Identities** In many cases, it is possible for new nodes to easily obtain new identities, effectively erasing their past history. Obviously, this can be used by nodes that had a malicious past behaviour. As [14] points out, this situation “. . . introduces opportunities to misbehave without paying reputational consequences. A large degree of cooperation can still emerge, through a convention in which newcomers ‘pay their dues’ by accepting poor treatment from players who have established positive reputations” .

**Collusive Attacks** A (possibly great, thanks to cheap identities) number of malicious nodes could introduce erroneous information in their history, in order to create attacks that aim to maliciously boost or decrease reputation of some nodes. Systems have to be designed in order to be resilient to this kind of attacks, giving more weight to data introduced by more reputable nodes.

In most cases, indirect reciprocation methods work on the graph having peers as nodes and reputation values calculated evaluating direct interaction history as edges. In the following, these local recommendations will be called *LRVs*, and the resulting graph will be called *LRV graph*. The goal is to evaluate distributed reputation values that are a function of the LRV graph  $G$ , the evaluating node  $x$  and the evaluated node  $y$ ; we will denote this as  $DRV(G, x, y)$ .

Some valuable ideas for evaluating DRVs can be borrowed from link analysis ranking [8]. Indeed, link analysis algorithms work on the graph representing links between web pages, and web links are recommendations between nodes, analogously to LRVs. Since it is obviously cheap to create a new web page, the problem of “cheap identities” is present in the WWW<sup>2</sup>. Moreover, collusive attacks on link analysis algorithms are known and studied as “web spam” [15, 13, 29]. The most significant difference, which implies there is no direct applicability of the same algorithms to P2P networks, is that web ranking algorithms are meant to be evaluated in a centralised structure storing all the collected data, doing an expensive centralised evaluation.

---

<sup>2</sup>Link analysis algorithms, in fact, implicitly punish new web sites that did not have enough time to gather links, confirming the cost for newcomers issue discussed above.

EigenTrust [19] is a decentralised implementation of the PageRank [25] algorithm used by the Google web search engine. It models a random walk over the LRV graph, and nodes are ranked according to the probability such a random walk has of ending on a given node. In this case, the DRV that gets calculated is independent of the evaluating node (i.e.,  $DRV(G, x, y) = DRV(G, x', y)$  for all  $x, x'$ ).

This algorithm requires that a predetermined set of nodes is “pre-trusted”, that is, they are automatically given an initial reputation value; those are the nodes from where the random walk is allowed to start. Unfortunately, the requirement of pre-trusted nodes creates weakness, since they introduce a degree of centralisation to the method. An attack to those nodes, and the LRVs given by them, would jeopardise the reliability of such method.

In [12], an incentive system is suggested where the proposed algorithm to evaluate  $DRV(G, x, y)$  is the maximal flow in  $G$  from  $x$  to  $y$ . This provides an effective incentive to cooperation, while maintaining resilience towards collusive attacks. Unfortunately, the MaxFlow calculation is expensive ( $O(V^3)$  in the worst case), and a global knowledge of the whole network is required in order to calculate it.

While approaches seen so far evaluate reliability of peers analysing links pointing to the evaluated node in the LRV graph, other approaches also take in consideration the similarity of links starting from the evaluating and the evaluated node. The idea is that nodes having similar “opinions” are expected to be more trustworthy [23, 21].

A different approach, suitable for unstructured P2P networks such as Gnutella [2] is a rewiring approach [3]: nodes constantly change their connections in the network whenever they are unsatisfied with their current neighbours, in order to get connected to better peers.

## Chapter 3

# Preliminary Results

Work developed so far has been centred on development of mechanisms for effective evaluation of DRVs, in the scenario defined by the issues introduced in section 2.3.

### 3.1 Neighbourhood Maps

This work is based on the idea that  $DRV(G, x, y)$  depends on the paths from  $x$  to  $y$  in  $G$ : given the assumption that a node which is trusted by a trusted node deserves some trust itself, it becomes apparent that paths on the network can be used to represent trust relationships, with shorter paths representing a more direct, thus more significant, relationship.

We make the assumption that the distance between two interacting peers in the LRV graph is usually low. In fact, we can expect LRV graphs for many P2P networks to be small-world graphs (i.e., low shortest path length between arbitrary node pairs):

- when nodes have the same probability of interacting with any peer, the resulting LRV graph can be modelled as a random graph, and random graphs have small-world properties;
- networks resulting from social relationships (e.g., in the P2P case, downloading the same file from a file-sharing network), have been found to be small-world networks [7] in most cases.

Basically, *neighbourhood maps* are a “local view” of reputation obtained by collecting information about the closest nodes. These maps are constructed by repeatedly contacting directly-connected nodes and combining data received from them. The size of the map is parametric, and the concept of closeness itself depends on the particular function used as a DRV.

When peer  $x$  wants to rank peer  $y$ ,  $y$ 's reputation is calculated by evaluating data based on the nodes known in both  $x$ 's and  $y$ 's neighbourhood maps. Since

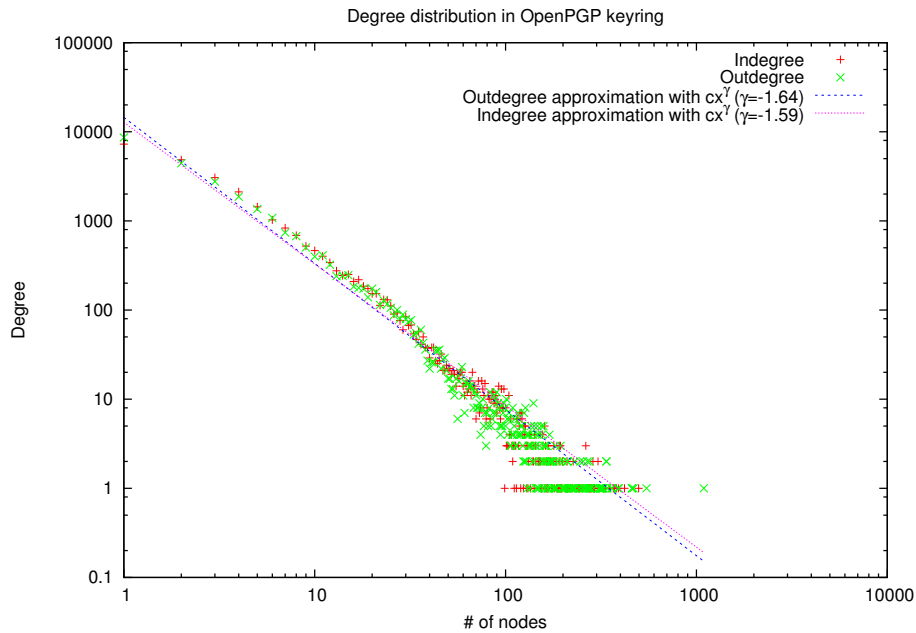


Figure 3.1: OpenPGP degree distribution

in small-world networks the average distance between couples of nodes is low, if neighbourhood maps are big enough, with high probability there will be an intersection between the two neighbourhood maps. Intersections represent middle points in paths from  $x$  to  $y$ , and data related to them is used to approximate the desired metrics.

We tested the correctness of our approximations on the OpenPGP web of trust. OpenPGP is a decentralised application for privacy and authentication. Since nodes are users and arcs are certifications between users of correspondence with a given identity, it is, in our opinion, relevant for our means, since it is a network of real-life trust relationships between nodes.

The OpenPGP data were gathered from [18] on May 9, 2005. It is a graph having 27398 nodes and an average of 8.99 outgoing edges per node; it is a small-world graph having an average shortest distance of 5.96 links. It is a scale-free [7] network, since its degrees can be approximated with a power-law relationship, as shown by the graph in figure 3.1.

## 3.2 Estimation of Shortest Path

Graph distance is a simple way of evaluating trust for another node. It is quite natural to think that nodes recommended by a “friend” (i.e., at distance 2) can be trusted up to a certain degree, those at distance 3 to a lesser degree, and so on. The concept of shortest path is indeed used as a reputation evaluation

means in the PGP web of trust [17, 18].

The construction of a neighbourhood map for distance can be done just by contacting neighbours in the network. Given a fixed size delimiter  $k$ , each peer stores the closest  $k$  nodes and their distances with regard both to incoming and outgoing paths. Finding an intersection between node  $x$ 's map for outgoing paths and node  $y$ 's map for incoming ones means that a path from  $x$  to  $y$  has been found.

Once the neighbourhood maps have been built, estimating distance means finding the minimal sum of distances for nodes that appear in both maps, i.e. returning the shortest path found. If such a path is not found, a “guess” has to be taken. Since we assumed to work with small-world graphs, the returned value is the sum of the maximum distances indexed by the two neighbourhood maps.

**Non-empty Intersections** Having a non-empty intersection between the two neighbourhood maps means a path has been found, and thus there is an upper bound on the distance between the nodes.

The graph in figure 3.2 on the following page shows the evolution of non-empty intersection probability versus the neighbourhood maps sizes. We compared the behaviour of the algorithm for the OpenPGP web of trust, a theoretical prediction for intersection probability for a random node sample, and the behaviour in a random network having the same number of nodes and edges as the web of trust graph. The neighbourhood map sizes vary from 0 to  $2\sqrt{n}$ , and the experimental data have been produced by evaluating intersection on 10000 random node pairs.

The curve for the PGP web of trust significantly deviates from the other two; we infer this could be motivated by two reasons:

- for low-size maps, having hubs<sup>1</sup> helps us, because it is more likely that a “famous” node is at a short distance between source and target, and thus we are more likely to find a path;
- for larger maps, we are paying for the high clustering of our network: information in neighbourhood maps is more redundant than in other cases, since maps of close nodes will be very similar and thus will convey less information.

**Approximation** In order to discuss the accuracy of our method in evaluating distances, we compared the results of our approximation with the exact distance.

The graph in figure 3.3 on the next page shows (with logarithmic scale on the  $y$  axis) how the relative average error evolves, compared to the map size. As usual, the test graph is the OpenPGP web of trust. The graph appears to suggest that the approximation given by our method increases exponentially with the map size.

---

<sup>1</sup>Hubs are “famous” nodes having a great number of links; it is a distinguishing feature of scale-free networks.

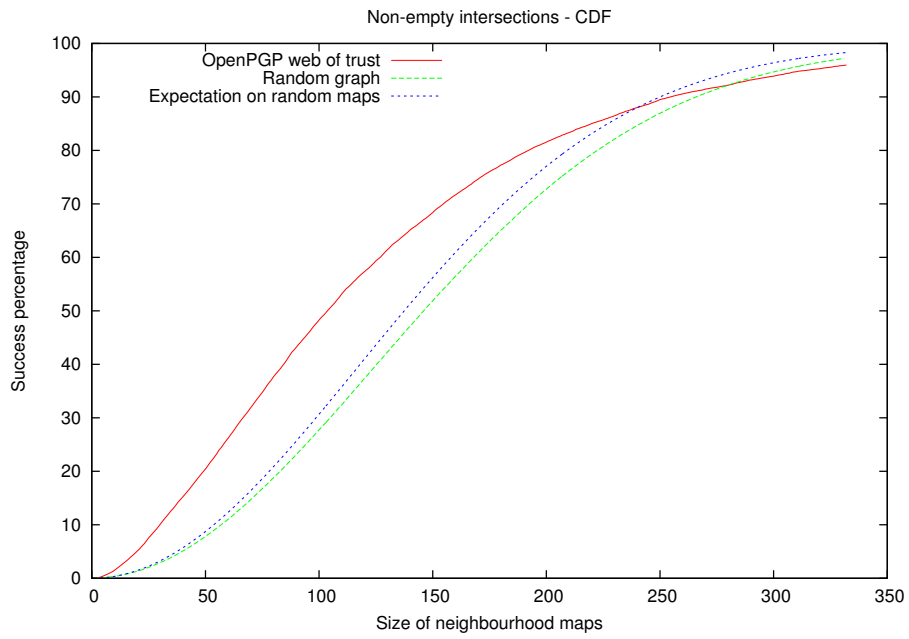


Figure 3.2: Non-empty intersection probability

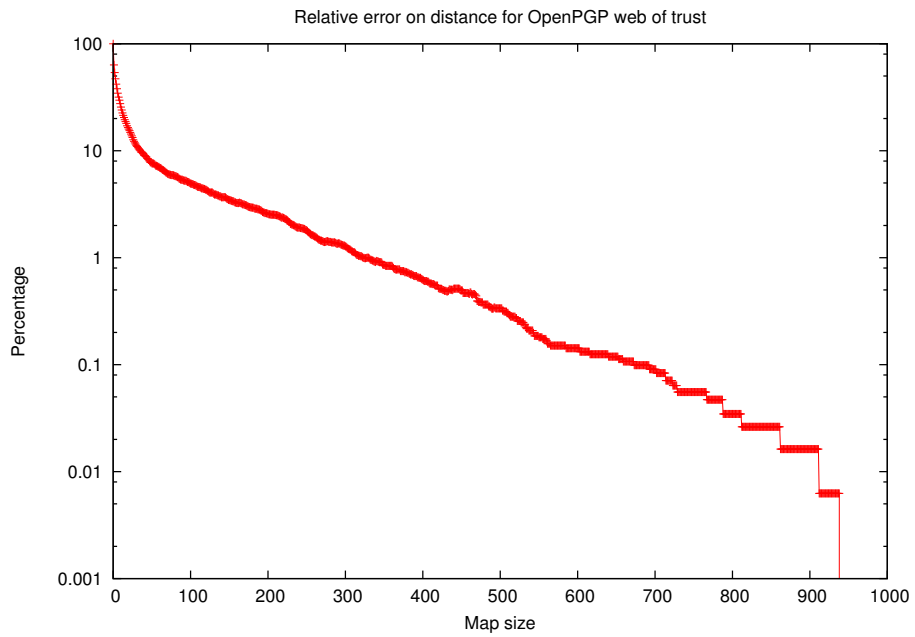


Figure 3.3: Relative error plot

The irregular behaviour on the right hand of the graph is motivated by the fact that when the map size is large the number of node pairs with an error is very low. In our 10000 sample pairs we could not find errors for a map size greater than 950.

### 3.3 Approximating PageRank

The most significant weakness of using graph distance as LRV function is that only one path between each pair of nodes on the LRV network is taken into consideration; thus, the resulting evaluation conveys little information. For having more significant results, we refer to link analysis algorithms. PageRank [25] is probably the most widely known of them, being used to rank results in the Google web search engine.

In PageRank, a random walk on the LRV graph is performed, with a probability  $\alpha$  of stopping at each step. The result of PageRank is the probability that each node has of being the end point of such a walk. Nodes having a higher probability in such distribution get a higher ranking. In the final asymptotic result, all paths of all lengths from the starting nodes are taken into account, with a decreasing weight for longer paths.

In usual PageRank implementations, the starting point is a random distribution over many or all the nodes in the graph; in our approach, the starting point is the evaluating node. This results in a *subjective* (i.e., depending on the node evaluating it) reputation evaluation. This provides security against collusive attacks.

**Map Construction** We choose  $DRV(G, a, b)$  to be the probability that a random walk in graph  $G$  starting from node  $a$  ends at node  $b$ . If  $N(G, n)$  is the set of neighbours (outgoing links) of  $n$ ,  $N^{-1}(G, n)$  the set of incoming links for  $n$ , and  $\#X$  the cardinality of the set  $X$ , the following equations hold:

$$DRV(G, a, b) = \begin{cases} \alpha + \frac{(1 - \alpha)}{\#N(G, a)} \sum_{n \in N(a)} DRV(G, n, b) & \text{if } a = b \\ \frac{(1 - \alpha)}{\#N(G, a)} \sum_{n \in N(a)} DRV(G, n, b) & \text{otherwise} \end{cases} \quad (3.1)$$

$$DRV(G, a, b) = \begin{cases} \alpha + (1 - \alpha) \sum_{n \in N^{-1}(b)} \frac{DRV(G, a, n)}{\#N(G, n)} & \text{if } a = b \\ (1 - \alpha) \sum_{n \in N^{-1}(b)} \frac{DRV(G, a, n)}{\#N(G, n)} & \text{otherwise} \end{cases} \quad (3.2)$$

Equations 3.1 and 3.2 can be used in order to evaluate neighbourhood maps for PageRank, using a fixed-point algorithm. In our case, the outgoing map for a node represents the nodes that are more likely to be reached after a random walk, while the incoming one the nodes with a higher probability of being reached.

**Using Neighbourhood Maps to Approximate PageRank** Let us say that peer  $a$  wants to evaluate peer  $b$ 's PageRank value (i.e.,  $DRV(G, a, b)$ ).

Let  $X_{a,b}$  be the set of nodes in both  $a$ 's outgoing map and  $b$ 's incoming one, that is  $x \in X_{a,b} \iff x \in O_a \wedge x \in I_b$ . For each  $x \in X_{a,b}$ , we have then calculated a probability  $O_a[x]$  that a random walk starting from  $a$  will stop at  $x$ , and a  $I_b[x]$  probability that a walk starting from  $n$  will arrive at  $b$ . Given the fact that the probability that a walk stops is  $\alpha$ , the probability of having a random walk getting from  $a$  to  $b$  passing through  $x$  is then

$$\frac{1 - \alpha}{\alpha} O_a[x] \cdot I_b[x].$$

As an estimation of ranking, we then use the following formula, ignoring the  $\frac{1-\alpha}{\alpha}$  constant which is not going to affect the ranking:

$$\sum_{x \in X_{a,b}} O_a[x] \cdot I_b[x].$$

**Avoiding “spam” attacks** An attacker could attack the “incoming” neighbourhood map for a node by creating a high number of fake nodes, and making them give a high ranking to the attacked node. In this way, that map would become full of references to insignificant nodes, and thus useless.

In order to avoid this kind of attack, nodes constructing the maps for incoming links can decide to give precedence to nodes that appear in their own “outgoing” map. For this reason we defined a priority value  $P(a, b)$  for each pair of nodes  $(a, b)$  such that  $b \in I_a$  as follows:

$$P(a, b) = \begin{cases} O_b[a] \cdot I_b[a] & \text{if } a \in O_b \\ \min_{x \in O_b} (O_b[x]) \cdot I_a[b] & \text{otherwise} \end{cases} .$$

Each node constructs its  $I_x$  map putting the first  $k$  nodes according to this priority value.

**Experimental Results** We evaluated the similarity of ranking between our method and the PageRank evaluation, using Kendall's tau distance as metrics. It consists in evaluating the probability that, given two rankings and a pair of random nodes, the two rankings agree on which one is ranked higher. The graph in figure 3.4 on the following page shows how the neighbourhood map size is related to this probability.

We can see that the “secure” method described in section 3.3 provides better results than the original one. This is probably due to the fact that using the secure method more relevant nodes are taken into account.

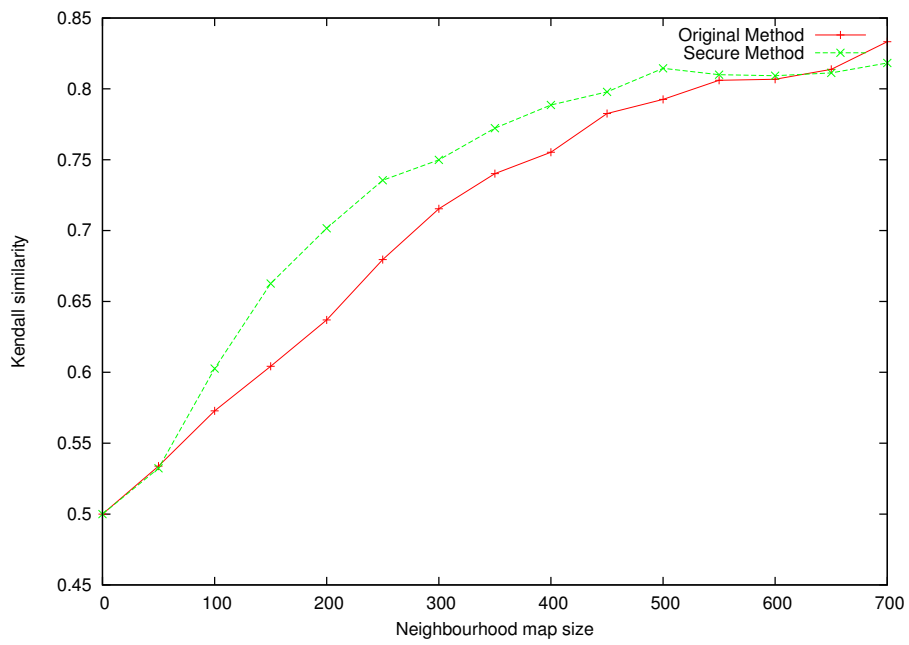


Figure 3.4: Kendall's tau distance for PageRank

## Chapter 4

# Planned Work

Methods intended be used to develop our reputation infrastructure will be presented in this chapter. Our scenario can be loosely defined by the following characteristics.

- The system will be designed in order to be independent with regard to the particular application being used (e.g., file-sharing, distributed computation, etc.).
- Our system is required to be viable in large-scale networks, having size of the order of millions of peers (such as today's P2P file-sharing networks).
- While a peer identifier corresponds to a unique user identity, the reverse is not true. In other words, nodes are free to create new identifiers whenever they want. This brings the issue of cheap identities discussed in section 2.3.
- Apart from itself, a peer does not need to unconditionally trust any other node. Trust is built along time with a correct behaviour, and should be lost when this kind of behaviour changes.

### 4.1 Reputation Evaluation

Work using the methodology of “neighbourhood maps” seen in chapter 3 will be carried on.

**Clustering** Since real-world networks usually have a high clustering, experimental studies will be made in order to quantify the impact of clustering in neighbourhood maps measurements. Real and synthetic networks will be used.

**Precision** Methods for increasing precision of the calculated metrics will be considered.

As seen in figure 3.4 on page 14, precedence values as discussed in section 3.3 have an effect on the calculation precision. In order to obtain a better precision, other precedence values may be taken into account.

**Other Metrics** Other metrics for reputation will be evaluated, and compared against the already implemented ones. As an example, the MaxFlow algorithm proposed in [12] and algorithms like HITS [20], that distinguish between hubs (in this sense, authoritative sources for recommendation) and authorities (entities that get recommended) will be taken into consideration.

Our approaches tend to take into consideration commonality between nodes recommended by the evaluating node and nodes that recommend the evaluated node. Another possibility is evaluation of similarity between nodes that are recommended by both parties. The latter approach implies that a degree of trust can descend from a commonality of views.

## 4.2 Security Issues

Besides the main goal of giving incentives for fair behaviour, there are two main security issues involved with the development of our infrastructure.

**Incentives for Correct Recommendations** Since maintaining correct information on other nodes has a (hopefully small) cost, we need to reward peers giving recommendations to good nodes. Thus, peers giving good recommendations need to receive a better quality of service than those which do not.

**Avoiding Collusive and DOS Attacks** Malicious users could collude, in order to artificially alter reputation values calculated for some nodes. Moreover, DOS attacks such as the “spam” attack described in section 3.3 could be created in order to make the reputation system return useless data.

The resulting system needs to be resilient to this kind of attacks, either by having a protocol in which such attacks are either unfeasible (for instance, because only trusted users are taken into consideration when evaluating another node) or detectable.

## 4.3 Experimental Setup

The effectiveness of the developed framework will be evaluated through extensive experimentation.

Data will be gathered from real-world networks, as with the PGP web of trust, in order to see how the different characteristics of the network influence the outcomes. Networks created using automated methods [6, 27] will be evaluated

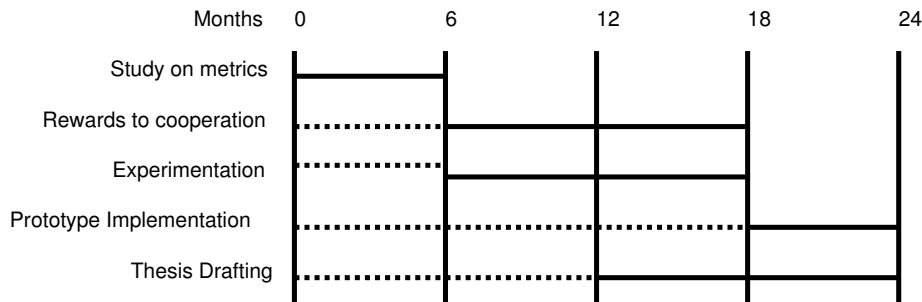


Figure 4.1: Work timeline

too. The study of differences between these synthetic networks and real ones is an interesting topic on its own.

A first step will be evaluation of similarity of approximated metrics by comparison with the exact ones, as done in chapter 3.

A game theoretical analysis will be carried out using an evolutionary approach [5, 12, 24]. A fixed pool of strategies are given, and the number of players using each strategy evolves relatedly to their *fitness* (i.e., the quality of service they obtain from the network) in the environment.

Effort will be spent on obtaining relevant data for usage patterns in P2P applications, such as file sharing, instant messaging or multimedia streaming. This will allow us to obtain a realistic simulation, resulting in more significant data.

The last goal is implementation of a prototype of the designed system, possibly in an existing and widely used P2P network. The choice of the particular platform that is going to be used is postponed, because of the high rate of innovation in the field.

## 4.4 Timeline

The thesis is planned to be developed in 24 months of work. In figure 4.1 a graphic presentation of the work plan is presented.

- In the first 6 months, work on metrics for reputation evaluation will be carried on as described in section 4.1.
- Months 6-18 will be devoted to the study of rewards to cooperation, proposing new methods and evaluating their security with regard to the security issues seen in section 4.2. This analysis will be continuously supported by experimentation, using the methods introduced in section 4.3.
- The last 6 months will be devoted to the implementation of a prototype of the developed system.
- The thesis will be written throughout the last 12 months of work.

## 4.5 Planned Thesis Structure

The thesis will be structured in 4 parts.

1. **Criteria for Reputation Evaluation:** state-of-the-art and novel methods for evaluating reputation in P2P systems will be presented.
2. **Mechanisms for Cooperation:** the metrics seen in the previous part will be used to build a system where nodes are effectively encouraged to cooperate; how to build and securely maintain a structure such as a LRV graph will be discussed.
3. **Experimental Results:** experiments using data both synthetic and taken from real-world use-cases will be conducted, in order to quantify the effectiveness of the developed methods. Analytical results, if relevant, will also be presented.
4. **Prototype Implementation:** this section will describe the implementation of the prototype that will be developed according to the results of our work.

# Bibliography

- [1] Edonkey2000. <http://www.edonkey2000.com>.
- [2] Gnutella - A Protocol for a Revolution. <http://rfc-gnutella.sourceforge.net>.
- [3] A. MARCOZZI, D. HALES, G. JESI, S. ARTECONI, AND O. BABAĞLU. Tag-Based Cooperation in Peer-to-Peer Networks with Newscast. Tech. Rep. UBLCS-2005-15, University of Bologna, Dept. of Computer Science, May 2005.
- [4] ADAR, E., AND HUBERMAN, B. A. Free riding on gnutella. *First Monday* 5, 10 (2 Oct. 2000).
- [5] AXELROD, R. *The Evolution of Cooperation*. Basic Books, New York, 1984.
- [6] BARABÁSI, A., ALBERT, R., AND JEONG, H. Mean-field theory for scale-free random networks. *Physica A* 272 (1999), 173–187.
- [7] BARABÁSI, A. L. *Linked*. Perseus, Cambridge, Massachusetts, 2002.
- [8] BORODIN, ROBERTS, ROSENTHAL, AND TSAPARAS. Link analysis ranking: Algorithms, theory, and experiments. *ACMTIT: ACM Transactions on Internet Technology* 5 (2005).
- [9] CACHELOGIC. The True Picture of Peer-to-Peer Filesharing. <http://www.cachelogic.com/research/p2p2004.php>, July 2004.
- [10] CACHELOGIC. Peer-to-Peer in 2005. <http://www.cachelogic.com/research/p2p2005.php>, 2005.
- [11] COHEN, B. Incentives build robustness in bittorrent. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems* (Berkeley, CA, USA, 2003).
- [12] FELDMAN, LAI, STOICA, AND CHUANG. Robust incentive techniques for peer-to-peer networks. In *CECOMM: ACM Conference on Electronic Commerce* (2004).

- [13] FETTERLY, D., MANASSE, M., AND NAJORK, M. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings of the 7th International Workshop on the Web and Databases* (Paris, France, June 2004), pp. 1–6.
- [14] FRIEDMAN, E. J., AND RESNICK, P. The Social Cost of Cheap Pseudonyms. *Journal of Economics & Management Strategy* (Aug. 17 2001).
- [15] GYÖNGYI, Z., GARCIA-MOLINA, H., AND PEDERSEN, J. Combating web spam with trustrank. In *30th International Conference on Very Large Data Bases* (2004), pp. 576–587.
- [16] HARDIN, G. The tragedy of the commons. *Science* 162 (1968), 1243–1248.
- [17] HENK P. PENNING. PGP pathfinder and key statistics. <http://www.cs.uu.nl/people/henk/henkpgp/pathfinder/>.
- [18] JÖRGEN CEDERLÖF. Wotsap. <http://www.lysator.liu.se/jc/wotsap/index.html>.
- [19] KAMVAR, S. D., SCHLOSSER, M. T., AND GARCIA-MOLINA, H. The eigentrust algorithm for reputation management in P2P networks. In *WWW* (2003), pp. 640–651.
- [20] KLEINBERG, J. Authoritative sources in a hyperlinked environment. *JACM: Journal of the ACM* 46 (1999).
- [21] LIK MUI. *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*. PhD thesis, Massachusetts Institute of Technology, 2003.
- [22] MANFRED MILINSKI, DIRK SEMMANN, AND HANS-JÜRGEN KRAMBECK. Reputation helps solve the ‘tragedy of the commons’. *Nature* 415 (Jan. 2002), 424–426.
- [23] NJÅL T. BORCH. Improving semantic routing efficiency. [http://the.socialized.net/papers/2005-Improving\\_semantic\\_routing\\_efficiency.pdf](http://the.socialized.net/papers/2005-Improving_semantic_routing_efficiency.pdf), 2005.
- [24] NOWAK, M. A., AND SIGMUND, K. Evolution of indirect reciprocity by image scoring. *Nature* 393, 6685 (1998), 573–577.
- [25] PAGE, AND LAWRENCE. PageRank: Bringing order to the web. Stanford Digital Libraries Working Paper 1997-0072, Stanford University, 1997.
- [26] PRAJIT K. DUTTA. *Strategies and Games*. MIT Press, 1999.
- [27] R. MILO, N. KASHTAN, S. ITKOVITZ, M. E. J. NEWMAN, AND U. ALON. Uniform generation of random graphs with arbitrary degree sequences. *Phys. Rev. E* 64, 2 (2001).

- [28] STOICA, I., MORRIS, R., KARGER, D. R., KAASHOEK, M. F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM* (2001), pp. 149–160.
- [29] Z. GYÖNGYI, AND H. GARCIA-MOLINA. Web spam taxonomy. In *First International Workshop on Ad-versarial Information Retrieval on the Web* (2005).