Institut Eurécom
Corporate Communications Department
2229, route des Crêtes
B.P. 193
06904 Sophia Antipolis
FRANCE

Research Report RR-06-176

Cooperation Incentive Schemes

26 September 2006

Nouha Oualha, Yves Roudier[1]

Tel: (+33) 4 93 00 81 95
Fax: (+33) 4 93 00 82 00
Email: {oualha, roudier}@eurecom.fr

[1] Eurecom's research is partially supported by its industrial partners: BMW Group Research & Technology – BMW Group Company, Bouygues Telecom, Cisco Systems, France Telecom, Hitachi, SFR , Sharp, STMicroelectronics, Swisscom, Thales

Cooperation Incentive Schemes

Nouha OUALHA, Yves ROUDIER[2]

Abstract

Decentralized systems, like peer-to-peer and ad hoc systems, may address scalability through the handing of system operational mechanisms over to the entities forming the system, in an autonomous manner. This paradigm raises concerns, in particular with respect to the establishment of trust between such entities, to the stimulation of their cooperation, and to the fairness of their respective contributions. This report surveys the cooperation incentive schemes that respond to these concerns, and that provide, given the diversity of cooperation techniques, many ways to manage resources and to control collaborating entities. This report first reviews application domains where collaboration between entities is required and enforced. The report then goes on to detail the cooperation incentive mechanisms, which can be categorized basically as reputation and remuneration based. Means of validating cooperation incentives are finally discussed.

[2] This work was also supported by the GET (Groupement des Ecoles de Télécoms) Initiative research program on autonomic and spontaneous networks, the ACI SI MOSAIC project, and the PACALAB research project.

Cooperation Incentive Schemes

Nouha OUALHA, Yves ROUDIER[3]

1. Introduction

Decentralized system algorithms and protocols have recently received a lot of interest in mobile ad-hoc networks as well as in peer-to-peer (P2P) systems. The development of such techniques is a necessity to be able to implement cost-effective and reliable applications deployable on a large scale, yet it brings up far-reaching issues that have to be dealt with. In decentralized systems, decision-making may not be located at a specific and central group of devices (repeaters, bridges, routers, gateways, servers) but can be distributed to end-user devices. Decisions and actions may use the computing power, bandwidth, and disk storage space of all the participants in the network rather than being concentrated in a relatively low number of special devices. The decentralized structure makes it possible to achieve minimal administrative and operational costs. Participants in this type of system are "peers" in the sense that they normally have equivalent responsibilities and privileges. The intricate notions of self-organization and self-management require that each peer provide its own contribution towards the correct operation of the system.

The handing of basic mechanisms of the system over to autonomous peers raises new concerns, in particular with respect to the establishment of trust between peers, to the stimulation of their cooperation, and to the fairness of their respective contributions. Self-organization opens up new security breaches because a peer must be able to defend against others perpetrating new forms of denial of service. Selfishness, as illustrated by the so-called free-riding attack, is a first type of such threats in which the attacker (called free-rider) benefits from the system without contributing its fair share. Systems vulnerable to free-riding either run at reduced capacity or collapse entirely because the costs of the system weigh more and more heavily on the remaining honest peers encouraging them to either quit or free ride themselves. Flooding is a second type of denial of service: the attack can be launched by sending a large number of query messages asking for resources to a victim peer in order to slow it until it is unusable or crashes. For example, an attacker can attempt to make a lot of read and write operations in a distributed storage application. Cheating (or retention) is a third form of denial of service in which the attacker retains data required for the system to work or does not comply with the normal course of action in order to obtain an unfair advantage over other peers. So-called "cooperation enforcement" mechanisms (which should more properly be called cooperation incentive schemes) provide ways of managing and organizing resources, and aim at dealing with the security challenges that traditional security approaches (e.g., authentication, access control) can not cope with.

The following sections introduce motivating applications for cooperation incentives, then detail how incentive schemes work, and finally discuss how these schemes may be validated.

[3] This work was also supported by the GET (Groupement des Ecoles de Télécoms) Initiative research program on autonomic and spontaneous networks, the ACI SI MOSAIC project, and the PACALAB research project.

2. Applications

Cooperation incentive mechanisms are present in various application domains. It is generally suggested that cooperation will help entities to succeed better than via competition. [Buttyán and Hubaux 2003] demonstrated that the best performance in mobile ad-hoc routing is obtained when nodes are very cooperative. In a cooperation incentive mechanism, cooperative behavior should be more beneficial than an uncooperative behavior. The two main categories of incentives are reputation and remuneration. This section describes several applications that benefit from cooperation enforcement.

2.1. Infrastructure based P2P applications

Infrastructure based P2P applications (sometimes termed P2P networks or even simply P2P) have become famous in several domains: file sharing is the flagship of such applications, yet other applications exist like the enabling of P2P file systems, or file backup systems.

2.1.1. File sharing

Peer-to-peer file sharing has become so widespread over the Internet that it now accounts for almost 80% of total traffic [Bolton and Ockenfels 2000] .

The Napster[4] protocol was historically the first to provide this service. Napster is based on a hybrid peer-to-peer infrastructure in which the index service is provided centrally by a coordinating entity, the Napster server. The functionality of the server is to deliver to a requesting peer a peers' list having the desired requested MP3 files. Then, the peer can obtain the respective files directly from the peer offering them.

In contrast, Gnutella[5] functions without any central coordination authority. Search requests are flooded into the network until the TTL (Time-To-Live hop counter) of the message has expired or the requested file has been located. Positive search results are sent to the requesting peer who can then download the file directly from the peer offering it. Both Napster and Gnutella focus more on information retrieval than on publishing.

Freenet [Cox and Noble 2002] provides anonymous publication and retrieval of data. Anonymity is provided through several means encrypted search keys and source-node spoofing. In Freenet, when peer storage is exhausted, files are deleted according to the least-recently-used principal so the system keeps only the most requested documents. Another drawback is the complexity of file search process. In fact there is a significant difference between Freenet and the systems presented so far which is that files are not stored on the hard disk of the peers providing them, but they are intentionally stored at other locations in the network. They are stored at peers having the numerically closest identification number to their IDs. The document lookup is a routing model based on keys to locate data similarly to Distributed Hash Tables (DHT). Free riding has been

[4] http://www.napster.com/

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

notably observed in such applications, and first attempts at using reputation incentives to counter it were made in systems like NICE [Lee et al. 2003].

NICE is a platform for implementing cooperative distributed applications, in particular P2P applications. The NICE system aims at identifying the existence of cooperative peers; it claims to efficiently locate the minority of cooperating users, and to form a clique of users all of whom offer local services to the community. The system is based on peer reputation which is stored in the form of cookies. These cookies provide a signed acknowledgment that a peer did (or did not) correctly provide the resources it promised, and are stored on that very peer. Whenever a server peer interacts with a client peer, the client will retrieve the cookie he previously sent to the server; if no interaction happened before, he might obtain some information out of the cookies stored by other client peers it may know.

2.1.2. Distributed file system

A generation of P2P applications uses the promising DHT-based overlay networks. DHTs such as CAN, Chord, Pastry, and Tapestry allow a decentralized, scalable, and failure-tolerant storage. Well-known approaches are PAST [Druschel and Rowstron 2000] based on Pastry and OceanStore [Kubiatowicz et al. 2000] based on Tapestry. Each PAST node can act as a storage node and a client access point. These schemes have basic similarities in the way they are constructed. Participants receive a public/private key pair. Keys are used to create an unambiguous identification number for each peer and for the stored files with the aid of a hash function. To use the storage, a peer has to pay a fee or to make available its own storage space. Key generation, distribution and monitoring are handled by “special” peers who have to be highly capable and highly available. Both PAST and OceanStore aim at ensuring a high data availability through means of file replication and random distribution of the identification numbers to peers. The procedure guarantees geographically-separated replicas which increases the availability of a given file.

Compared with PAST and OceanStore, Free Haven [Dingledine 2000] is designed for more anonymity and persistence of documents than for frequent querying. An author in Free Haven generates a public/private key pair, signs his document fragments, and uploads them into the server. Each server hosts data from the other servers in exchange for the opportunity to store data of its own into the community of servers, servnet. Trading of document fragments adds to author anonymity. When a reader wishes to retrieve a document from the servnet, he requests it from any server, including a location and key which can be used to deliver the document in a private manner. This server broadcasts the request to all other servers, and those which are holding shares for that document encrypt them and deliver them to the reader's location.

2.1.3. Data backup

The latest generation of peer-to-peer systems is a generation of storage systems having data backup as its primary function. Pastiche [Cox and Noble 2002] is based on Pastry for locating nodes and exploits excess disk capacity to perform peer-to-peer backup with no administrative costs. Each Pastiche node minimizes storage overhead by selecting peers that share a significant amount of data. It replicates its archival data on more than one peer. Most of these replicas are placed nearby to ease network overhead and minimize restoration time. To address the problem of storing data on malicious nodes, Pastiche uses a probabilistic mechanism to detect missing backup state by periodically querying peers for stored data. However it sacrifices a fair amount of privacy because nodes can grab some information about the backup data. This issue is less critical for the CIBS (Cooperative Internet Backup Scheme) [Lillibridge et al. 2003] scheme where fragments of a file are stored at different geographical locations, and partners are tracked by a central server. To ensure a high reliability, the scheme adds redundancy through Reed-Solomon erasure correcting code.

2.2. Wireless networks

Collaboration does not only benefit infrastructure based applications, but also proves essential in several areas of wireless networks. This section gives three examples of applications that critically depend on cooperation incentives to be effectively enabled: mobile ad-hoc routing wireless ad-hoc backup, and wireless data dissemination.

2.2.1. Mobile ad-hoc routing

Multi-hop ad-hoc networks, frequently referred to under the term MANETs (Mobile Ad hoc NETworks), can be set up rapidly and spontaneously. Connections are possible over multiple nodes. These nodes operate in a decentralized and self-organizing manner and do not rely on a fixed network topology. Intermediate nodes in a route have to act as routers to forward traffic towards its destination. To achieve this operation, incentives for cooperation between nodes become a requirement, because rational users would rather preserve the energy of their personal devices rather than spend it on cooperative routing. There has been a wealth of work on cooperative network forwarding.

In the Watchdog/Pathrater [Marti et al. 2000] scheme, the watchdog detects non-forwarding nodes by overhearing the transmission, and the pathrater keeps a rating of every node and updates it regularly. The two components enable nodes to route messages avoiding misbehaving nodes in their route. Misbehaving nodes are detected and avoided in the routing path but not punished.

In CONFIDANT (Cooperation Of Nodes, Fairness in Dynamic Ad-hoc NeTworks) [Buchegger and Le Boudec 2002], the response to misbehaving nodes is more severe than just avoiding them for routing; it also denies them cooperation. Similarly to Watchdog/Pathrater, in CONFIDANT reputation is self-carried by nodes. Nodes monitor their immediate neighborhood and also gather second-hand information from others. By Bayesian estimation, they classify other nodes as normal or misbehaving.

In CORE (COllaborative REputation) [Michiardi and Molva 2002], the information collected is classified into subjective reputation (direct information), indirect reputation (positive reports from other nodes), and functional reputation (task-specific information). The combined reputation value is used to make decisions regarding a given node, that is, to either cooperate with it or gradually isolate it.

TermiNodes ([Buttyán and Hubaux 2001]) uses a different approach based on a tamper-proof security module for each node maintaining a nuglet counter. When the node wants to send a packet, it decreases its nuglet value by a number of credits proportional to the estimated number of intermediate nodes in the route. When the node forwards a packet, its nuglet purse becomes bigger.

While TermiNodes uses a tamper-proof hardware placed at each node, Sprite [Zhong et al. 2003] does not require any tamper-proof hardware at any node. Sprite is based on a central Credit Clearance Service (CCS). Every node is supposed to have a digital ID obtained from a Certification Authority (CA). When a node receives a message, the node keeps a receipt of the message. Sprite assumes that every source node knows the entire path to the destination node through a secure ad hoc routing protocol based upon DSR. The underlying ad hoc routing protocol only exists for packet delivery, not for routing decision making. When the node has a fast connection to the CCS, which is reachable via an overlay network, it reports to the CCS the messages that it has received/forwarded by uploading its receipts. Depending on the reported receipts of a message, the CCS then determines the charge and credit to each node involved in the transmission of a message. Zhong et al. introduce a formal model in order to prove the effectiveness of Sprite in restraining selfish behavior at the network layer, however Sprite presents a weakness for it relies on the accessibility of the CCS.

2.2.2. Wireless ad-hoc backup

The Flashback [Loo et al. 2002] application is a first example of wireless ad-hoc backup system, which has been proposed for Personal Area Networks (PAN). Flashback is a solution designed to provide peer-to-peer power-aware backup for self-managing, mobile, impoverished devices. In Flashback, each device has an identifier assigned out-of-band during the installation of the Flashback. To ensure that devices only participate in the PAN to which they are assigned, a lightweight certificate-based authentication scheme is used. In order to keep track of replicas, Flashback maintains a replica table in persistent storage. It uses an opportunistic model to replicate local data according to the constraints imposed by the power and storage resources.

The MoSAIC project[6] [Killijian et al. 2004] is another example of data backup system and recovery service based on mutual cooperation between mobile devices. Such a service aims to ensure availability of critical data managed by wireless mobile devices. Data availability is ensured by replicating data. To finely control the level of data redundancy, MoSAIC uses an erasure coding technique for the production of redundant fragments [Killijian et al. 2006]. In contrast with Flashback, which assumes a single authority – the user – preinstalling identity certificates on every device to be backed up, MoSAIC targets ephemeral and self-organizing networks that come into existence spontaneously by virtue of physical proximity of mobile devices and where peers are mutually suspicious. A cooperation incentive scheme is critical in the MoSAIC system.

[6] The MoSAIC Project, project partners: Institut Eurécom, IRISA, LAAS. http://www.laas.fr/mosaic/

2.2.3. Nomadic computing

Nomadic computing aims at offering end users access to data or information from any device and network while they are mobile. There are two fundamental approaches to these applications: the first one, as proposed by the DataMan project[7] or by the Ubibus prototype [Banâtre et al. 2004], assumes the availability of information servers (called info-stations), placed at fixed positions (e.g., traffic lights, building entrances, and airport lounges), and that supply mobile users with contextual information; the second one assumes that data are distributed among neighboring mobile users in a cooperative fashion. 7DS [Papadopouli and Schulzrinne 2001] is an illustration of the latter approach: this system allows a peer to browse the content of the cache of another peer in order to search for URLs or keywords. This operation can be performed either on-demand or in a prefetching mode. When prefetching, 7DS anticipates the information needs of the peer, while on-demand retrieval only searches for information when the peer requests it. Mobile devices therefore do not need any base stations to gain access to the service in 7DS. However, while this system relies on the cooperation of nodes, users are not encouraged to provide storage space for information caching, nor to answer neighbor requests, nor even to provide accurate answers. History-based credentials used in context-aware applications like [Bussard et al. 2004] for instance provide means to implement a self-carried reputation scheme, thus aiming at ensuring such a cooperation of users or devices in nomadic computing applications. One-time capabilities [Bussard and Molva 2004] also aim at discouraging non-cooperation but using a remuneration based punishment scheme.

2.3. Web commerce

Cooperation incentives are often used in web commerce sites. This section presents some examples of such incentives: auction sites, and review or recommendation sites. All schemes are reputation systems where the reputation is computed either using a voting scheme or using the average of ratings.

2.3.1. Auction sites

Auction sites allow sellers to list items for sale, buyers to bid for these items, then the items to be sold to the highest bidder. In general, the person who puts the item up for auction pays a fee to the auctioneer. In some cases, there is a minimum or reserve price; if the bidding does not reach the minimum, the item is not sold. Reputation systems are used in auctioning in order to help users making good choices when selecting transacting partners. eBay[8] is one popular online auction site. The feedback forum on eBay allows sellers and buyers to rate each other as positive, negative, or neutral. Ratings of buyers and sellers are conducted after the completion of a transaction, which is monitored by eBay. The reputation system relies on a centralized repository that stores and manages ratings. The overall reputation of a participant is the sum of ratings about him over the last 6 months. eBay also provides one month old and seven day old ratings to let users know about recent behavior of the participant. The eBay system makes it possible to

[7] DATAMAN, http://planchet.rutgers.edu/~badri/dataman/research-projects.html

[8] http://ebay.com/

perform fake transactions. Even though, this incurs a cost, since eBay charges a fee for listing items: this still opens up opportunities to acquire undue ratings.

2.3.2. Review and recommendation sites

In review sites, individual reviewers, who are generally individuals, provide information to fellow consumers. In these systems, a reputation rating is applied to both products and reviewers themselves, in particular to discourage product bashing. One example of such a system is Amazon[9], an online bookstore that allows members to write book reviews. A user can become an Amazon member by simply signing up. Reviewers' reviews of a book are made of some text and a rating in the range of 1 to 5 stars. Members and users rate reviews as being helpful or not. Amazon ranks reviewers based on their rating and other parameters (which are not publicly revealed). Reviewers with a high ranking are given the status of top reviewers. To reduce repetitive ratings from the same users, Amazon only allows one vote per registered cookie for any given review. Epinions[10], a similar review site charges product manufacturers and online shops by the number of clicks that consumers generate as a result of reading about their products. This makes it possible for top reviewers to get paid, while in contrast, Amazon does not give any financial incentive for well-reputed reviewers.

3. Incentive Schemes

Cooperation is a central feature of decentralized systems, and even more so ad-hoc ones, to compensate for the lack of a central and dedicated entity and still achieve some general function. However, cooperation to achieve some functionality may be hampered by the fact that users have full authority on their devices and, as proven by experience, will on average try to maximize the benefits they get from the network. In general, the cooperative behavior of a device will indeed result in an increase in its resource consumption or missed opportunities to take more than its fair share of a resource (e.g. network, CPU, storage space). In case of mobile ad hoc forwarding for instance, the node forwarder is confronted with additional energy and bandwidth usage for reception and transmission of packets, as well as with the increase of computational resource consumption. Knowing that mobile devices have inherently scarce resources, each of these devices should better not cooperate from its point of view. In case of file sharing applications, a node can take advantage of the system by downloading files without contributing to it. To counterbalance this, and achieve an overall better result, it is primordial to design incentive mechanisms for cooperation that discourage uncooperative behaviour, be it passive or malicious. At the same time, these mechanisms can not prevent the non cooperative behavior of devices due to valid and reasonable reasons (e.g., crashing, energy shortage, route breaks), which should normally not be punished as if they were malicious non-cooperation.

As seen in section B, there are many cooperation incentive schemes which are diverse not only in terms of the applications for which they are useful or critical, but also in terms of the features they implement, the type of reward and punishment used, and their operation over time. [Obreiter and Nimis 2003] classifies cooperation enforcement mechanisms into trust-based patterns and

[9] http://www.amazon.com/

[10] http://www.epinions.com/

trade-based patterns. The authors make a distinction between static trust, thereby referring to pre-established trust between peers, and dynamic trust, by which they refer to reputation-based trust. Oreiter et al. analyze trade-based patterns as being based either on immediate remuneration, which they term barter trade, or on deferred remuneration, which they term bond-based. While this classification was the first to try to address so many different incentive schemes together, other authors describe cooperation only in self-organized systems, in which case they classify cooperation schemes into reputation based (*"dynamic trust-based"* for Obreiter et al.) and remuneration based (*"trade-based"*) approaches. Trust establishment, a further step in many protocols, easily maps to reputation systems but may use remuneration systems as well. The following sections use the reputation versus remuneration classification.

3.1. Reputation based mechanisms

In reputation-based mechanisms, the decision to interact with a peer is based on its reputation. Reputation mechanisms need reputation management systems for which the architecture is either centralized, or decentralized, or both.

3.1.1. Reputation based system architecture

The estimation of reputation can be performed either centrally or in a distributed fashion. In a centralized reputation system, the central authority that collects information about peers typically derives a reputation score for every participant and makes all scores available online. In a distributed reputation system, there is no a central authority for submitting ratings or obtaining reputation scores of others. However, it might be some kind of distributed storage where ratings can be submitted. One example of such architecture is FastTrack [Liang et al. 2006] architecture which is used in P2P networks like KaZaA[11], Grokster[12], and iMesh[13]. These networks have two-tier hierarchy consisting of ordinary nodes (ONs) in the lower tier and supernodes (SNs) in the upper tier. SNs are generally more powerful in terms of connectivity, bandwidth, processing and non-NATed (Network Address Translation) accessibility. SNs keep tracks of ONs and other SNs and act as directory servers during the search phase. Such an architecture can be convenient to manage peer reputations using supernodes as distributed storage; unfortunately this is not the case in the existing FastTrack-based P2P networks. In KaZaA for example, each node has a participation level based some QoS (Quality of Service) parameters that is stored locally. The participation level score is used in prioritizing peers during periods of high demand. Most of the time in a distributed architecture, ratings are estimated autonomously by each peer. Each peer records ratings about its experiences with other peers and/or tries to obtain ratings from other parties who have had experiences with a given target peer. A good example of a decentralized reputation-based approach to trust management is NICE [Lee et al. 2003]. This system searches the network at runtime and builds a trust graph where each edge represents how much the source trusts the destination. A reputation value is calculated based on this trust graph. Then, the NICE algorithm selects a trust path based on whether it is the strongest path or using a weighted sum of strongest disjoint paths.

[11] http://www.kazaa.com/
[12] http://www.grokster.com/
[13] http://imesh.com

The centralized approach to reputation management is not fault-tolerant. In the decentralized approach, it is often impossible or too costly to obtain cooperation evaluations resulting from all interactions with a given peer. Instead reputation is based on a subset of such evaluations, usually obtained from the neighborhood. The reputation mechanism should therefore be designed such as to avoid inconsistencies. Distributed reputation management systems are most probably a more appropriate design than centralized ones to achieve a scalable solution to cooperation incentives. A distributed algorithm will allow applications to scale up to a large community of users by making it possible to have local standard of reputation. Some applications or networks being typically decentralized, like wireless ad hoc networks, and especially MANETs, they require the use of a reputation management system itself decentralized.

3.1.2. Reputation system operations

A reputation-based mechanism is composed of three phases (Figure 1):

1. Collection of evidence: Peer reputation is constructed based on the observation of the peer, experience with it, and/or recommendations from third parties. The semantics of the information collected can be described in terms of a specificity-generality dimension and a subjectivity-objectivity dimension:

- Specific vs. general information: specific information about a given peer relates to the evaluation of a specific functionality or aspect of this peer such as its ability to deliver a service on time. Whereas, general information refers to all functionalities (e.g., measured as a weighted average).
- Objective vs. subjective information: a peer obtains objective information (also known as direct or private information) about a given peer through his personal but concrete interactions with the considered peer, and subjective information (also known as indirect or public information) by listening to messages or negotiations that are intended to other peers, or by asking neighboring peers their opinions about the actual peer.

2. Cooperation decision: Based on the collected information, a peer can make a decision whether he should cooperate with another peer, based on the reputation of that other peer. There exist a variety of methods for computing the reputation of an entity, some of which being described below:

- Voting scheme: the simplest method to compute reputation is to compute the sum of positive ratings minus the sum of negative ratings. This is the principle used in eBay[14]'s reputation forum.
- Average of ratings: another simple scheme is to compute the reputation score as the average of all ratings given by peers or users. The Amazon[15] recommendation system uses such a scheme (although it does not provide standard deviation and may therefore be less precise than eBay's reputation system).

[14] http://ebay.com

[15] http://www.amazon.com/

- Bayesian based computation: reputation is computed based on the previous estimated reputation with the new evaluation score. Reputation systems (like [Jøsang and Ismail 2002] and [Mui et al. 2001]) use the beta PDF (Probability Density Function) denoted by *beta(p |α , β)* using the gamma function *Γ*.

  $$beta(p \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)+\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}; 0 \leq p \leq 1, \alpha > 0, \beta > 0$$

  Where *α* and *β* represent the amount of positive and negative ratings respectively, and p represents the probability variable. The PDF expresses the uncertain probability that future interactions will be positive (cooperation).

- Flow model: “systems that compute trust or reputation by transitive iteration through looped or arbitrarily long chains can be called flow models” [Jøsang et al. 2005]. One example of such a reputation computing function (although not a cooperation measurement) is PageRank [Page et al. 1998], Google[16]’s algorithm to rank web pages[17] based on their "reputation" (number of its referrals). The page rank of a web page *u* is defined as:

  $$R(u) = cE(u) + c \sum_{v \in N^-(u)} \frac{R(v)}{| N^+(v) |}$$

  where $N^-(u)$ denotes the set of web pages pointing to *u*, $N^+(v)$ denotes the set of web pages that *v* points to, and *E* corresponds to a source of rank. A hyperlink is a positive referral of the page it points to, and negative referrals do not exist because it is impossible to blacklist a web page using the above equation. EigenTrust [Kamvar et al. 2003] is another flow model that computes a global trust value for a peer by multiplying iteratively normalized local matrices of trust scores of each peer in the system. With a large number of matrices, the system will converge to stable trust values. In this model, trust and reputation are evaluated similarly.

- Other computation models are described in [Jøsang et al. 2005]: *(a) the discrete trust model*, in which the trustworthiness of the neighbor is taken into account before considering subjective information; *(b) the belief model*, which relates to the probability theory, and in which the sum of probabilities over all possible outcomes does not necessarily add up to 1, the remaining probability being interpreted as uncertainty; *(c) fuzzy models* finally, in which reputation and trust are considered as fuzzy logic concepts.

3. Cooperation evaluation: The occurrence of interaction with a peer is conditional on the precedent phase. After interaction, a node must provide an evaluation of the degree of

[16] http://www.google.com/

[17] The public PageRank measure does not fully describe Google's page ranking algorithm, which takes into account other parameters for the purpose of making it difficult or expensive to deliberately influence ranking results in what can be seen as a form of "spamming".

cooperation of the peer involved in the interaction. Peers performing correct operations, that is, behaving cooperatively, are rewarded by increasing their local reputation accordingly. A peer with a bad reputation will be isolated from the functionality offered by the group of peers as a whole. The evaluation of the current interaction can convey extra information about other past interactions (piggybacking) that can be collected by the neighboring peers.

3.1.3. Attacks and counter-measures

Cooperative mechanisms have to cope with several problems due to node misbehavior. Misbehavior ranges from simple selfishness or lack of cooperation to active attacks aiming at denial of service (DoS), attacks to functionality (e.g., subversion of traffic), and attacks to the reputation system (liars).

To guard against the impact of liars, the CORE mechanism [Michiardi and Molva 2002] for instance takes into account only positive reputation from indirect information, together with reputation from direct information (does the node hear the packet forwarded by its peer?): defamation is thus avoided, yet unjustified praising is still possible. In a more restrictive manner, RPG (Reputation Participation Guarantee) [Barreto et al. 2002] forbids the diffusion of reputation between peers. Only direct information is taken into account; selfishness is detected by sending probe packets.

A different approach, relying on indirect information, is taken in Watchdog/Pathrater [Marti et al. 2000]. A watchdog is in charge of identifying the misbehaving nodes, and a pathrater is in charge of defining the best route avoiding these nodes. It is pretty much the same approach that is taken in CONFIDANT [Buchegger and Le Boudec 2002]: a neighborhood monitor has the role of identifying misbehavior, which is rated.

A trust manager sends and receives alarm messages to and from other trust managers, while a path manager maintains path ranking. As a result, nodes in the network will exclude misbehaving nodes by both avoiding them for routing and by denying them cooperation. So, misbehaving nodes will be penalized by being isolated. Contrary to many proposals, Watchdog/Pathrater evaluates cooperation but does not enforce it: non cooperative nodes in Watchdog/Pathrater will not be punished like in CORE or CONFIDANT, their messages are still forwarded while they are not forced to forward the messages of the other nodes.

The systems presented so far focus on network layer forwarding. In this type of application, cooperation evaluation is immediate, yet other mechanisms may require the evaluation to take place on a longer timescale. Reputation estimates then need to be preserved: this may mean that it is self-carried by the peer if reputation is based on direct information; otherwise, reputation should not depend on the proximity of the peer since nearby nodes are likely to move away over a long period of time. This is for instance the case for distributed backup applications.

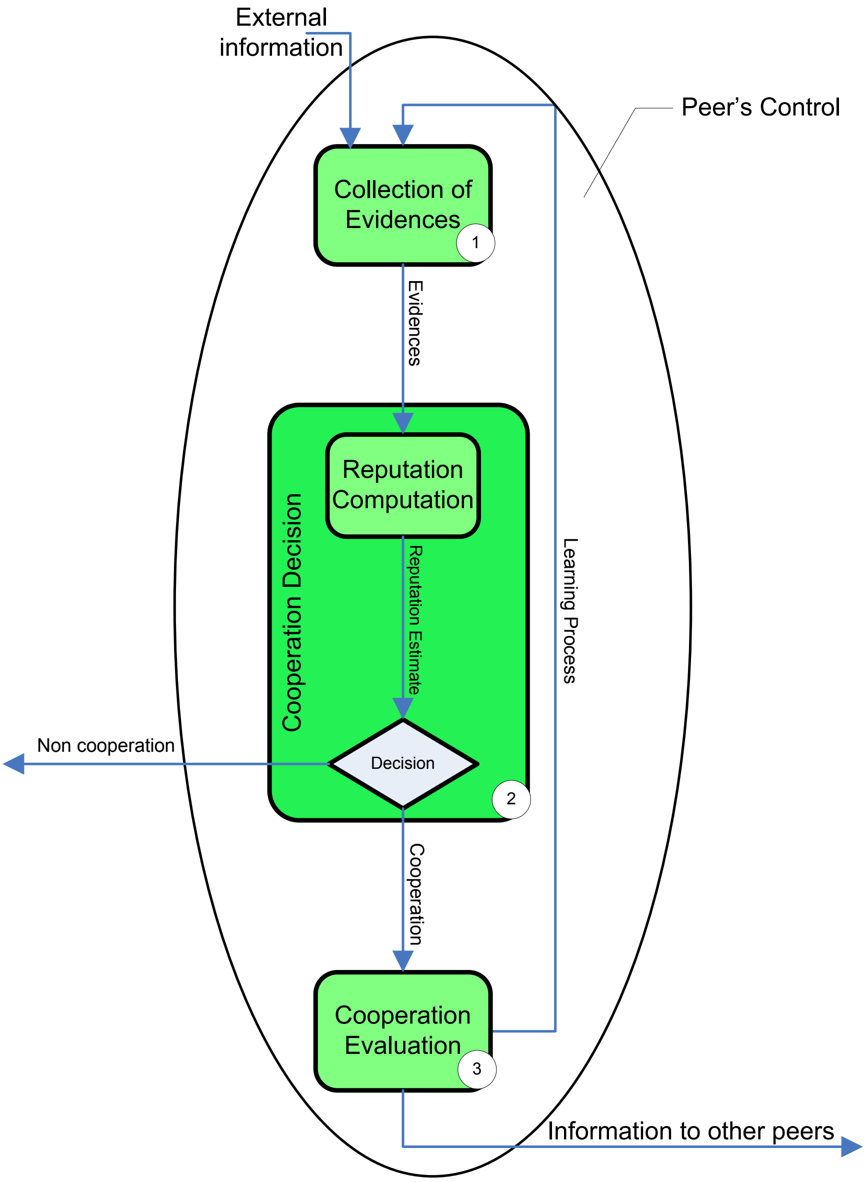

Figure 1: Reputation-based Mechanism

In the CIBS (Cooperative Internet Backup Scheme) scheme [Lillibridge et al. 2003], each computer has a set of geographically-separated partner computers that collectively hold its backed up data. In return, the computer backs up a part of its partner's data. To thwart free riding attacks, a computer can periodically challenge each of its partners by requesting him to send a block of the backed up data[18]. An attack can then be detected and the data blocks of the attacker that are stored in the attacked computer are consequently dropped. In this scheme, each peer takes note of its direct experience with a partner, and if this partner does not cooperate voluntarily or

[18] Some disruption attacks, i.e. attacks aiming at disrupting, impairing or destroying a system or a particular user, can be avoided by limiting reads to mutually chosen random blocks.

not beyond some threshold, the peer may decide to establish a backup contract with a different partner.

Another example of reputation-based mechanisms for distributed storage is the Free Haven project [Dingledine 2000]. The overall design of the project is based on a community of servers, called the servnet where each server hosts data from other servers in exchange of the opportunity to store data of its own in the servnet. The incentives for cooperation are based on a reputation mechanism. A trust module on each server maintains a database of each other server, logging past direct experience as well as what other servers have said.

3.2. Remuneration based mechanisms

In contrast to reputation-based mechanisms, remuneration based incentives are an explicit counterpart for cooperation and provide a more immediate penalty to misconduct. Remuneration brings up requirements regarding the fair exchange of the service for some form of payment [Asokan et al. 1997]. This requirement in general translates to a more complex and costly implementation than for reputation mechanisms. In particular, remuneration based mechanisms require trusted third parties (TTP) such as banks to administer remuneration of cooperative peers; these entities do not necessarily take part in the online service, but may be contacted in case of necessity to evaluate cooperation. Tamper proof hardware (TPH) like secure operating systems or smart cards have been suggested or used to enforce in a decentralized fashion the fair exchange of the remuneration against a proof that the cooperative service was undertaken by a peer node.

3.2.1. Remuneration based system architecture

A remuneration based mechanism comprises four main operations (see Figure 2):

- **Negotiation:** The two peers may often negotiate the terms of the interaction. Negotiating the remuneration in exchange for an enhanced service confers a substantial flexibility to the mechanism. The negotiation can be performed either between the participating peers or between peers and the authority.
- **Cooperation decision:** The peer in a self-organizing network is always the decision maker. During negotiation and based on its outcome, a peer can decide if it is better to cooperate or not.
- **Cooperation evaluation:** Cooperation has to be evaluated by the service requesting party, in terms of adequacy of the service to the request, as well as by the service providing party, in terms of adequate remuneration. Ensuring the fairness of both evaluations may ultimately require involving a trusted third party. Depending on the service, this TTP will ensure a fair exchange for every interaction, or may only be involved if arbitration is requested by one party (see below). The TTP, which may be centralized or distributed itself, may for instance give access to information unavailable to a peer, or more generally provide a neutral execution environment.
- **Remuneration:** The remuneration can consist in virtual currency units (a number of points stored in a purse or counter) or real money (banking and micropayment), or bartering units (for instance quotas defining how a certain amount of resources provided by the service may be exchanged between entities). The latter can even be envisioned in the form of micropayments [Jakobsson et al. 2003]. Regarding real money, this solution assumes that

every entity possesses a bank account, and that banks are enrolled in the cooperative system, directly or indirectly through some payment scheme. The collaborating peer is remunerated by issuing a check or making a transfer of money. In the first case, remuneration implies a number of points added to a counter connected with the collaborating peer. The remuneration can be guaranteed at once or only after a certain number of steps (deposit, remuneration for data storage, remuneration for data retrieval…).

These operations can be used repeatedly to perform some cooperative service on a finer granularity basis, which may ease cooperation enforcement. In particular, micropayment is often envisioned rather than an actual (macro-)payment in remuneration based cooperation enforcement mechanisms.

3.2.2. Fair exchange

As mentioned in [Asokan et al. 1997], "*many commercial transactions can be modeled as a sequence of exchanges of electronic goods involving two or more parties. An exchange among several parties begins with an understanding about what item each party will contribute to the exchange and what it expects to receive at the end of it. A desirable requirement for exchange is fairness. A fair exchange should guarantee that at the end of the exchange, either each party has received what it expects to receive or no party has received anything*." Fair exchange protocols thus provide ways to ensure that items held by two or more parties are exchanged without one party gaining an advantage. In remuneration systems, obtaining an efficient cooperation incentive depends upon devising a protocol that enforces a fair exchange of the remuneration (virtual or not) against some task. This property can only be attained by intricately integrating the remuneration operation with the application functionality. Fair exchange protocols rely on the availability of a trusted (and neutral) third party (TTP) caring for the correctness of the exchange. Two types of protocols should be distinguished: online protocols, which mediate every interaction through the TTP, which can lead to performance and reliability problems with the TTP constituting a bottleneck as well as a single point of failure; offline ones, also called optimistic fair exchange protocols, which resort to the TTP intermediation only if one of the parties wants to prove that the exchange was not fairly conducted.

The TermiNodes project ([Buttyán and Hubaux 2001]) addresses the security of the networking function of packet forwarding through remuneration schemes. Each device possesses a security module that manages its account by maintaining a counter called nuglet, interpreted as virtual money. The project proposes two models for remuneration aiming at enforcing fair exchange for stimulating a cooperative behavior. In the first one, called Packet Purse Model, each packet carries a given number of nuglets and intermediate nodes get paid with some nuglets, which get removed from the packet purse when forwarded by the node. In the second model, called Packet Trade Model, each intermediate node buys packets from the previous node on the route then sells them to the next node for more nuglets, until the destination node, which finally pays the total cost of forwarding packets.

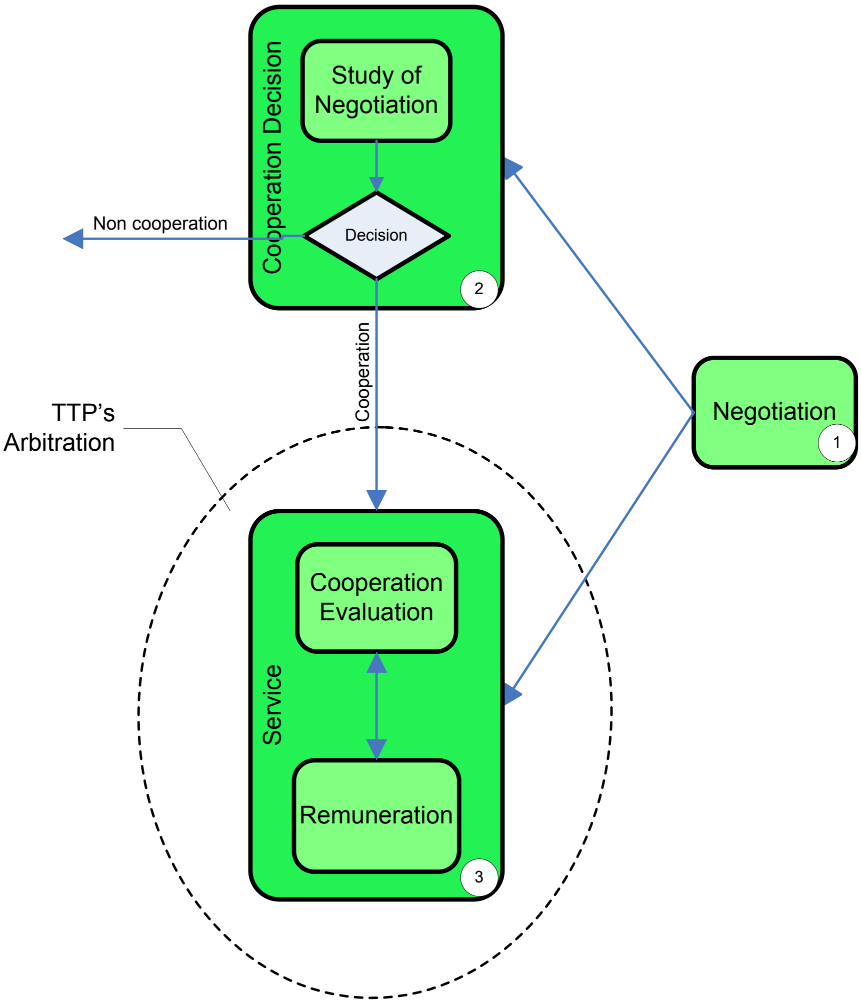

Figure 2: Remuneration-based Mechanism

The architecture of Sprite [Zhong et al. 2003], a credit-based system for stimulating cooperation among selfish nodes in mobile ad hoc networks, is not very different from TermiNodes except for the fact that it does not use security modules. It consists of a Credit Clearance Service (CCS) and of mobile nodes. In Sprite, a node transmitting its own messages loses some credits (i.e., virtual money paid by the node to the CCS), which will be used to cover the costs for packet forwarding by intermediate nodes. In order to earn credits, a node must transmit the CCS receipts of forwarded messages. The system does not guarantee balanced payments, i.e., it does not require that the sender's total debt equal the total credit received by intermediate nodes for their forwarding activity. In fact, to prevent cheating behavior, the CCS debits the sender with a higher amount than that due to intermediate nodes; only later does the CCS uniformly share the exceeding credit among nodes, or give a fixed credit amount to each node. Sprite focuses on combating cheating behavior and on promoting cooperation among network nodes; it does not prevent active attacks on the system (e.g., Denial of Service attacks).

Akin to Sprite, Mojo Nation [McCoy 2001], a content distribution technology, is built on a micropayment system. A common digital currency, the Mojo, is used to buy disk space, bandwidth, and processing cycles contributed to the system. Peers who have contributed resources to the system are credit for their exact participation. Interaction between peers across the network involves the exchange of Mojo currencies. A TTP ensures honest transfers between agents within the network.

In contrast, in OceanStore [Kubiatowicz et al. 2000], the remuneration of cooperative peers is monetary as the service is envisioned to be provided by a confederation of companies. In exchange for economic compensation, computers joining the system contribute with storage or provide access to local users. Each user is supposed to pay a fee to one particular provider who buys storage space from and sells it to other providers. Legal contracts and enforcement can be used to punish peers that do not keep their end of the bargain, based on planned billing and auditing systems.

Networking context		Application	Incentives for cooperation		No Incentive for cooperation (or closed system)
			Reputation-based incentives	Remuneration-based incentives	
Infrastructure-based	P2P network	File-sharing	NICE	Mojo Nation	Napster[19], Gnutella[20], Freenet
		File system	Free Haven	PAST, OceanStore	
		Backup	CIBS		Pastiche
Wireless	Ad hoc network	Packet forwarding	CORE, RPG, Watchdog/pathrater, CONFIDANT	TermiNodes	Ad hoc IEEE 802.11 standards[21]
		Backup	MoSAIC[22]	FlashBack	
	Nomadic computing	Nomadic computing	History-based certificates	One-time capabilities	Ubibus, 7DS
Centralized network		Web commerce	Ebay[23], Amazon[24], Epinions[25]		Most review sites

Table 1: Cooperation enforcement schemes in various applications

[19] http://www.napster.com/
[20] http://www.gnutella.com/
[21] http://grouper.ieee.org/groups/802/11/
[22] The MoSAIC (http://www.laas.fr/mosaic/) project partners: Institut Eurécom, IRISA, LAAS are members of ReSIST
[23] http://ebay.com/
[24] http://www.amazon.com/
[25] http://www.epinions.com/

Smart cards (or similar forms of tamper-proof hardware like secure operating systems) have been proposed for some time now as a means to implement an optimistic fair exchange protocol. The use of smart cards is especially interesting since it provides both the convenient form factor of a personal token and ideally a perfect implementation of a secure purse. Smart cards also offer a tamper-resistant area for a trusted third party to store secrets, or implement critical security functions, in particular for revoking the user from the system. [Vogt et al. 2001] for instance proposes the use of one card for all parties involved in a transaction and focuses on providing a neutral platform for enforcing fair exchange. Another solution is to use one card for each party [Terada et al. 2004], which aims at reducing the number of messages exchanged; this solution may also be interesting, although this benefit is not mentioned by the authors, for gaining a better understanding of the liability of each party involved and thus ease the reconciliation of the data by the TTP during the online phase of the protocol, in case of a problem in the offline phase. In addition, the latter technique makes it possible to attach data like some credit to every user and let the smartcard manage this "currency" as part of the fair exchange protocol. As discussed above, however, remuneration has to be integrated with the application: this means that interactions with such hardware must be carefully thought out in the application protocol in order to prevent its bypassing or abuse: with smart cards for instance, this involves the mediation of terminals, which are distrusted with respect to remuneration handling.

A smart card based remuneration mechanism is for instance used in the peer-to-peer storage system PAST [Druschel and Rowstron 2000]. PAST is based on the Pastry routing scheme that guarantees that peers contributing to cooperative storage are geographically separated. The storage scheme relies on the use of smart cards to ensure that clients cannot use more remote storage than they are providing locally, which is optional in PAST. Smart cards are held by each PAST user and issued by a third party, and support a quota system that balances supply and demand of storage space in the system. With fixed quotas and expiration dates, users are only allowed to use as much storage as they contribute.

Table 1 summarizes the various approaches to cooperation and their respective features as discussed in the last two sections.

4. Validation techniques

In the context of self-organizing networks like for instance wireless mobile ad hoc networks, cooperative mechanisms have to be investigated in terms of performance, fairness, and resilience to attacks, as well as cooperation enforcement. Experimentation is an obvious validation approach, yet it suffers from scalability issues. Otherwise, cooperation incentives may be validated using either simulation or game theory.

4.1. Prototype-based evaluation

A cooperation mechanism can be validated by building a prototype, that is, a physical model of a proposed product concept that allows demonstration, evaluation, or testing of the most representative attributes and idiosyncrasies of a mechanism. Prototypes are especially important to fine tune parameterized schemes. The literature offers a lot of examples of P2P or ad hoc cooperation mechanisms whose evaluation process was based on prototypes. To validate their incentive system for ad hoc networking, a prototype was used for Sprite [Zhong et al. 2003] to determine how much overhead was necessary for the incentive scheme and to evaluate the packet routing performance of the system (percentage of packets successfully relayed from the sender to the destination). Results show that the overhead of the Sprite system is insignificant, and that nodes in the system cooperate and forward each other's messages unless their resources are extremely low. The Pastiche [Cox and Noble 2002] scheme was also evaluated using the prototyping approach. Pastiche's prototype consists of two main components: the chunkstore file system, implemented in user space and written in C, and a backup daemon. With the evaluation of the prototype, it was demonstrated that the backup service does not penalize the file system performance unduly and also that node discovery was effective. CIBS [Lillibridge et al. 2003] was also prototyped for the validation of the backup scheme. To measure the performance of their Internet backup scheme, its authors used a number of personal computers running instances of the prototype software. Each instance was partnered with the other instances located in different PCs so that all communication between partners went through the network. Experiments on the prototype have shown that the backup scheme performance is acceptable in practice and that the technique is feasible and cheap. Validating cooperation incentive schemes using pure experimentation however proves difficult because of scale for many applications that were considered in the previous sections. This experimental approach has however proven quite successful with real-world evaluations in P2P file sharing systems, especially so because of their widespread usage.

4.2. Simulation

An alternative validation technique for cooperative systems consists in taking advantage of existing network simulators in order to obtain results for a virtual deployment on a large scale of distribution. These simulators are tailored to fit the simulation context and match the objectives of a given application by the use or development of patches and/or individual adjustments. According to [Brown and Kolberg 2006], "simulation can be defined as the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behavior of the system and/or evaluating various strategies for the operation of the system". This means that simulating cooperation incentives with the purpose of testing their efficiency would also require simulating non-cooperative behaviors (and not only an ideal cooperative behavior).

Many applications considered beyond layer 3 require simulating overlay networks, which proves a bit difficult. Firstly, most overlay networks need to be scalable (thousands of simultaneous users) which is difficult to realize due to memory constraints even for most powerful machines. However, some tools allow a simulation to be distributed over a set of machines (distributed simulation). Additionally, it is generally desirable that a simulation behaves in accordance with real network parameters (packet delay, traffic and network congestion, bandwidth limitations etc). These considerations increase the overhead on the host machine. Packet-level network simulators such as ns-2, OMNET++[26], GloMoSim[27]/QualNet[28], and OPNET[29] must be distinguished from overlay network simulators like PeerSim[30].

[26] The OMNeT++ Community Site, http://www.omnetpp.org/

[27] Global Mobile Information Systems Simulation Library, GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim/

Ns-2. There is no doubt that the most popular network simulator is ns (version 2)[31]. Ns-2 is an object-oriented, discrete event[32] driven network simulator developed at UC Berkeley.. The simulator ns-2 is written in C++ and OTcl. Ns-2's code source is split between C++ for its core engine and OTcl, an object oriented version of Tcl for configuration and simulation scripts. The combination of the two languages offers an interesting compromise between performance and ease of use. An ns-2 simulation scenario is a Tcl file that defines the topology and the movement of each host that participates in an experiment. Implementing a new protocol in ns-2 typically requires adding C++ code for the protocol's functionality, as well as updating key ns-2 OTcl configuration files in order for ns-2 to recognize the new protocol and its default parameters. The C++ code also describes which parameters and methods are to be made available for OTcl scripting. Debugging is difficult in ns-2 due to the dual C++/OTcl nature of the simulator. For the moment, there is only one P2P simulation available for ns-2 which is Gnutella. More troublesome limitations of ns-2[33] are its large memory footprint and lack of scalability as soon as simulations of a few hundred to a few thousand of nodes are undertaken. Ns-2 is well documented with active mailing lists.

OMNET++. OMNET++[34] is another discrete event simulator. It is an open-source, component-based environment with a strong focus on supporting the user with a Graphical User Interface (GUI). The simulator is very well structured and modular, modules being programmed in C++ and assembled into larger components using a high level language (NED). It is possible to simulate peer-to-peer networks with OMNET++ which can also run distributed simulations over a number of machines. OMNET++ has a rapidly increasing user base now, with lots of useful modules, an active mailing list and even workshops. Both ns-2 and OMNET++ are packet-level simulators; so scalability is also a major issue; just like ns-2, OMENT++ is more suitable for small networks.

GloMoSim[35]/QualNet[36]. GloMoSim is built as a scalable simulation environment for wireless and wired network systems. It is designed using the parallel discrete-event simulation capability provided by PARSEC (C-based simulation language). PARSEC is designed to cleanly separate the description of a simulation model from the underlying sequential or parallel simulation protocol, used to execute it. GloMoSim is built using a layered approach. Standard APIs are used between the different layers. This allows the rapid integration of models developed at different layers. To specify the network characteristics, the user has to define specific scenarios in text configuration files: app.conf and Config.in. The first contains the description of the traffic to generate (application type, bit rate, etc.) and the second contains the description of the remaining parameters. The statistics collected can be either textual or graphical. With GloMoSim, it is

[28] Scalable Network Technologies (SNT), http://www.scalable-networks.com/products/

[29] OPNET, http://www.opnet.com/

[30] Biology-Inspired techniques for Self-Organization in dynamic Networks, BISON Project, http://www.cs.unibo.it/bison/

[31] The network Simulator ns-2, http://www.isi.edu/nsnam/ns/index.html

[32] A discrete-event simulator is a simulator where state variables change only at discrete points in time at which events occur caused by activities and delays

[33] The network Simulator ns-2, http://www.isi.edu/nsnam/ns/index.html

[34] The OMNeT++ Community Site, http://www.omnetpp.org/

[35] Global Mobile Information Systems Simulation Library, GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim/

[36] Scalable Network Technologies (SNT), http://www.scalable-networks.com/products/

difficult to describe a simple application that bypasses most OSI layers. Bypassing the protocol stack is not obvious to achieve as most applications usually lie on top of it which makes the architecture much less flexible. The free GloMoSim version is for academic use only. The commercial GloMoSim-based product is QualNet. The development framework is in C/C++ and mostly provided in source form. It includes a graphic development tool for adding/revising protocols. The two simulators provide substantial support for simulation of routing protocols over wired and wireless networks.

OPNET. OPNET Modeler[37] is a commercial tool part of the many tools from the OPNET Technologies suite (Optimized Network Engineering Tools). OPNET is an event-driven scheduled simulator integrating analysis tools for interpreting and synthesizing output data, graphical specification of models and hierarchical object-based modeling. It can simulate all kinds of wired networks, and an 802.11 compliant MAC layer implementation is also provided. OPNET is a well established product used by large companies to diagnose or reorganize their networks. It can simulate wired and wireless networks. Models built with OPNET are hierarchically structured. At the lowest level, the process domain is structured as a finite state machine (FSM). The FSM can be structured with the help of a graphical editor that allows the user to specify the relation between the single states and their transitions. The single states and the transition conditions can then be programmed with a C-like language called Proto-C. Basically, the deployment process goes through the following phases. First, one has to choose and configure the node models required in the simulations, (for example a wireless node, a workstation, a firewall, a router, a web server, etc.). Then the network is built and organized by connecting the different entities. The last step consists in selecting the statistics to collect during the simulations. Most of the deployment in OPNET is done through a hierarchical GUI. OPNET scales quite well but not many data in the literature demonstrate its capabilities.

PeerSim. In addition to these network simulators, there also exist simulators that only focus on overlay networks. A good example is PeerSim[38], which has been developed for large-scale overlay systems within the BISON project. It makes it possible to simulate scalable and dynamic overlays. PeerSim is written in the Java language. It is composed of two simulation engines: a cycle-based one and a more traditional event-based engine. The cycle-based engine does not model the overhead of the transport layer and subsequently is more scalable. The event-based engine is less efficient but more realistic. The simulation engines are supported by many simple, extendable, and pluggable components, with a flexible configuration mechanism.

Other simulators may be found in the literature about peer-to-peer or overlay systems, e.g., **3LS** [Ting and Deters 2003], **Query-Cycle Simulator** [Schlosser and Kamvar 2002], **Anthill** [Babaoglu et al. 2002] and **NeuroGrid[39]**. None of these simulators seems really satisfactory. 3LS, Anthill and NeuroGrid have scalability limitations. Query-Cycle is limited to file-sharing. All seem to lack enough support for dynamicity. In conclusion, ns-2[40], GloMoSim[41]/QualNet[42],

[37] OPNET, http://www.opnet.com/

[38] Biology-Inspired techniques for Self-Organization in dynamic Networks, BISON Project, http://www.cs.unibo.it/bison/

[39] NeuroGrid, http://www.neurogrid.net

[40] The network Simulator ns-2, http://www.isi.edu/nsnam/ns/index.html

[41] Global Mobile Information Systems Simulation Library, GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim/

[42] Scalable Network Technologies (SNT), http://www.scalable-networks.com/products/

OMNET++[43], OPNET[44] and PeerSim are potential candidates to build up a simulation environment for evaluating cooperation incentives in an ad hoc or P2P system. OPNET and ns-2 possess an extensive set of models, protocols and algorithms already produced, but less than OMNET++. The modular nature of OMNET++ makes it possible to carry out studies over a wide range of situations in detail. Also, regarding the ease of use and extensibility, OMNET++ appears to be the best simulator. OPNET and QualNet are also more than satisfactory with respect to this capability, however ns-2 scores poorly.

Some of the incentives schemes for cooperation listed above were investigated using a network simulator. In 7DS [Papadopouli and Schulzrinne 2001] simulation scenarios, hosts were modeled as ns-2 mobile nodes. Mobile nodes move according to the random waypoint mobility model, which is commonly used to model the movement of individual pedestrians. A waypoint model breaks the movement of a mobile node into alternating motion and rest periods. A mobile node moves at a speed uniformly chosen from an interval to a randomly chosen location where it stays for a fixed amount of time; then it chooses another random location and moves towards it, and so on. An ns-2 simulation study was also carried out for the ORION project [Klemm et al. 2003] where the performance of ORION was compared to off-the-shelf approaches based on a P2P file-sharing system for Internet, TCP, and a MANET routing protocol. In the simulated scenarios, an IEEE 802.11 standard MAC layer was used along with the standard physical layer, the two-ray ground propagation model. Ns-2 was widely employed for simulation principally in wireless mobile networks more than in P2P or ad-hoc networks where other simulators like QualNet[45] [46] were mostly adopted. CORE was evaluated with QualNet simulations [Michiardi 2004], and CONFIDANT [Buchegger and Le Boudec 2002] with GloMoSim[47] ones.

In addition to existent simulators, it is also possible to devise an ad-hoc simulation model to validate a cooperative incentive scheme. The mobile P2P file-sharing simulation model from [Oberender et al. 2005] contains a “network” component to model the network and devices’ particularities and restrictions. It also includes a “source traffic” component to model the data transmitted and the behavior of peers in the network. The mobile P2P architecture used in this model is based on the eDonkey P2P file-sharing protocol and is enhanced by additional caching entities and a crawler. In the simulation, a mobile peer is described by an ON/OFF-process to reflect the fluctuating connection status of a mobile peer. ON and OFF periods are determined by exponential distributions, while the arrival of file requests is modeled by a Poisson process. An abstract model using a subset of the parameters of the detailed simulation is proposed to reduce the computing time. The abstract model is used to identify which cache replacement strategy fits the best for the mobile P2P system. In order to do so, the request arrival process is simulated in detail while the used transport mechanism and the upload queue mechanism are neglected.

[43] The OMNeT++ Community Site, http://www.omnetpp.org/

[44] OPNET, http://www.opnet.com/

[45] Global Mobile Information Systems Simulation Library, GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim/

[46] Scalable Network Technologies (SNT), http://www.scalable-networks.com/products/

[47] Global Mobile Information Systems Simulation Library, GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim/

	Simulator	P2P protocols	Language	Distributed simulation	Conditions
Packet-level network simulator	ns-2[48]	Gnutella	C++/OTcl	Yes	Open source
	OMNET++ [49]	None	C++/NED	Yes	Academic public license
	GlomoSim [50]/QualNet[51]	---	C/C++	Yes	Free for universities / commercial
	OPNET[52]	---	Proto-C	Yes	Commercial
Overlay network simulator	PeerSim[53]	Collection of internally developed P2P models	Java	No	Free
	3LS	Gnutella	Java	No	---
	NeuroGrid[54]	Gnutella, NeuroGrid, Pastry, FreeNet	Java	No	Free

Table 2: Characteristics of discussed simulators

4.3. Game theory

Results obtained through simulation studies give a proof-of-concept of the proposed cooperative mechanism. The results do not demonstrate if the incentives for cooperation are crucial or work by chance for instance. Game theory provides an alternative tool to decide if a cooperative mechanism is a cooperation strategy. Game theory models strategic decision situations where self-interested users follow a strategy aiming at maximizing their benefits and minimizing their resource consumption. Game theory offers different methods for study, e.g., non-cooperative game, cooperative game, and evolutionary game. *Non-cooperative game* focuses on users' strategies. It describes the strategy of a user that has to make a decision about whether to cooperate or not with a randomly chosen user. *Cooperative game* focuses on mutually advantageous results for the different parties. In this game, users are able to enforce contracts and make binding agreements. Finally, *evolutionary games* address the evolution of various strategy profiles over time and space.

The decision making process that a peer will undertake when participating in a non-cooperative game is often illustrated through the example of a classical game, *the prisoner's dilemma*. In this well-known game, two players are both faced with a decision to either cooperate (C) or defect (D). The two players' decisions are made simultaneously with no knowledge of the each other's decision. If the two players cooperate they receive a benefit R. If both defect they receive a

[48] The network Simulator ns-2, http://www.isi.edu/nsnam/ns/index.html
[49] The OMNeT++ Community Site, http://www.omnetpp.org/
[50] Global Mobile Information Systems Simulation Library, GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim/
[51] Scalable Network Technologies (SNT), http://www.scalable-networks.com/products/
[52] OPNET, http://www.opnet.com/
[53] Biology-Inspired techniques for Self-Organization in dynamic Networks, BISON Project, http://www.cs.unibo.it/bison/
[54] NeuroGrid, http://www.neurogrid.net

punishment P. If one player defects and the other one cooperates, the defecting player receives a given benefit T and the cooperator a punishment S. The canonical form of the prisoner's dilemma pay-off is shown in the table below:

		Player 2	
		C	D
Player 1	C	(R, R)	(S, T)
	D	(T, S)	(P, P)

Table 3: Prisoner's dilemma pay-off matrix

In order to have a dilemma, the following expressions must hold:

$$T>R>P>S \text{ and } R>(S+T)/2$$

A player in this game has better to defect regardless of the decision of the other player because the strategy D strictly dominates the strategy C ($T>R$ and $P>S$). The solution to this game is called a *Nash equilibrium*, that is, the set of strategies for which no player could improve his payoff (by changing his strategy) while the other players keep their strategies unchanged. The analysis of the interaction between decision-makers involved in the prisoner's dilemma game can be extended to *repeated (or iterated) games*. There are several strategies that a player can adopt to determine whether to cooperate or not at each of its moves in the repeated game. The basic strategy known as tit-for-tat corresponds to a player that cooperates in the first place and then copies his opponent's last move for all subsequent periods. Another strategy called Spiteful is to cooperate in the first period and for later periods cooperate if both players have always cooperated; if either player defects then defect for the remainder of the game. A cooperation enforcement mechanism can be translated into a strategy for a player and compared to these straightforward strategies.

Another important concept is the idea of *evolutionary stable strategy*. A set of strategies is at evolutionary stable strategy equilibrium if (a) no individual playing one strategy could improve its payoff by switching to one of the other strategies in the population and (b) no individual playing a different strategy (called a mutant) could establish itself in (invade) the population, i.e., make other individuals in the population choose his strategy. With these different concepts, one can give a good analysis of the cooperation enforcement mechanism in terms of both promoting cooperation and the evolution of cooperation. Cooperation and coalition formation can be explained using a two-period structure. Players first decide whether or not to join a coalition and in the second step the coalition and non-cooperative peers choose their behavior non-cooperatively. A coalition is defined as stable if no peer in the coalition has an incentive to leave. In this sense, a preference structure was suggested by the *ERC-theory* [Bolton and Ockenfels 2000]. In this theory, the utility of a decision-maker is not solely based on the absolute payoff but also on the relative payoff compared to the overall payoff to all peers. The model explains observations from games where equity, reciprocity or competition plays a role.

The CORE mechanism [Michiardi 2004] was for instance validated following two methodologies. The first approach was to use a simulation tool, GloMoSim[55], while the second validation used a game-theoretic methodology. For the latter, two models, cooperative and non-cooperative game theory, were evaluated to demonstrate the need for cooperation incentives in the network. The CORE mechanism was then translated into a strategy model and its evolutionary stability was proven. Further, it was shown that in a more realistic scenario (communication errors, failures, etc.), CORE outperforms other basic cooperation strategies. Sprite [Zhong et al. 2003] also used a game-theoretic model to prove that the scheme proposed prevents cheating behaviors. The main results are intended for packet forwarding in unicast communication. The most important role of the game-theoretic validation of Sprite algorithms is in determining payments and charges of nodes in the system to motivate each node to cooperate honestly and to report its behavior to the CCS (Credit Clearance Service). Finally, remuneration fairness was studied from a game theoretic point of view in [Buttyán and Hubaux 2000], in which the authors discuss different equilibrium concepts as a model for the various types of fair exchange.

5. Conclusion

Different approaches can be taken for cooperation enforcement, yet cooperation evaluation is clearly the part most dependent on application-specific requirements and constraints, in particular concerning deployment. The choice of a particular technique for validating the effectiveness of cooperation, a critical step to ensuring that the application will reach its objectives, depends heavily on the application chosen. This may in particular hamper the use of one technique because of scalability issues or because of the properties that need to be proven.

Trust, as one would like to evaluate it in the applications mentioned above, can be static (based on identity for instance) or dynamic (self-organized). Static trust refers to a statement of trustworthiness that remains the same until it is revoked, whereas dynamic trust exhibits self-learning and self-amplifying characteristics. The latter arises from behaviors experienced in the system and continuously changes accordingly to them. An entity trusts a peer more when it has information about that peer that shows its trustfulness. [Carbone et al. 2003] for instance introduces a trust model that does not only concentrate on the content of evidence but also on the amount of such evidence.

Trust, although closely related with cooperation, may not be valuably accounted for by all cooperation evaluation metrics of the mechanisms listed above. In particular, whereas reputation seems well adapted to reason with the trustworthiness of a peer, remuneration may be much poorer semantically, especially if the payment may be used to enforce cooperation for different self-organized services. This does not mean that trust does not require cooperation as a prerequisite, but instead that trust establishment might not be as reliable as expected if the evaluation of its cooperative component relies on an unsuitable incentive mechanism.

[55] Global Mobile Information Systems Simulation Library, GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim/

6. References

[Asokan et al. 1997] N. Asokan, M. Schunter, and M. Waidner. Optimistic Protocols for Fair Exchange. In Proceedings of the 4th ACM Conference on Computer and Communications Security, Zurich, April 1997.

[Babaoglu et al. 2002] O. Babaoglu, H. Meling, and A. Montresor, “Anthill: A framework for the development of agent-based peer-to-peer systems”, In Proceedings of the 22th International Conference on Distributed Computing Systems (ICDCS'02), Vienna, Austria, July 2002.

[Banâtre et al. 2004] M. Banâtre, P. Couderc, J. Pauty, and M. Becus. Ubibus: Ubiquitous Computing to Help Blind People in Public Transport. In proceedings of Mobile HCI 2004, pages 310-314, 2004.

[Barreto et al. 2002] D. Barreto, Y. Liu, J. Pan, and F. Wang, “Reputation-based participation enforcement for ad hoc networks”, 2002.

[Ben Azzouna and Guillemin 2004] N. Ben Azzouna and F. Guillemin, “Experimental analysis of the impact of peer-to-peer applications on traffic in commercial IP networks”, European transactions on Telecommunications: Special issue on P2P networking and P2P services, ETT 15(6), November-December 2004.

[Bolton and Ockenfels 2000] G. E Bolton and A. Ockenfels, “ERC: a theory of equity, reciprocity, and competition”, American Economic Review 90(1): 166-193, 2000.

[Brown and Kolberg 2006] A. Brown, M. Kolberg, “Tools for Peer-to-Peer Network Simulation”, March 3rd, 2006, http://www.ietf.org/internet-drafts/draft-irtf-p2prg-core-simulators-00.txt

[Buchegger and Le Boudec 2002] S. Buchegger, and J. Y. Le Boudec, “Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes — Fairness In Distributed Ad-hoc NeTworks”, In Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC), Lausanne, CH, IEEE (2002) 226–236, 2002.

[Bussard et al. 2004] L. Bussard, Y. Roudier, R. Molva. Untraceable Secret Credentials: Trust Establishment with Privacy. In Proceedings of the First IEEE International Workshop on Pervasive Computing and Communication Security (PerSec 2004), Orlando, Florida, March 14, 2004.

[Bussard and Molva 2004] L. Bussard, R. Molva. One-Time Capabilities for Authorizations without Trust. In Proceedings of the second IEEE International Conference on Pervasive Computing and Communications (PerCom 2004), Orlando, Florida, March 13-17, 2004, pages 351-355.

[Buttyán and Hubaux 2000] L. Buttyan and J.-P. Hubaux, Toward a formal model of fair exchange -- a game theoretic approach, 2000, http://citeseer.ist.psu.edu/article/buttyan00toward.html, updated version of the SSC Technical Report No. SSC/1999/039

[Buttyán and Hubaux 2001] L. Buttyán and J. Hubaux, “Nuglets: a virtual currency to stimulate cooperation in self-organized ad hoc networks”, Technical report, EPFL, 2001.

[Buttyán and Hubaux 2003] L. Buttyán and J.-P Hubaux, “Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks”, ACM/Kluwer Mobile Networks and Applications, 8(5), October 2003.

[Carbone et al. 2003] M. Carbone, M. Nielsen, and V. Sassone, “A Formal Model for Trust in Dynamic Networks”, BRICS tech. report RS-03-4, Univ. Aarhus, 2003.

[Clarke et al. 2001] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, “Freenet: A distributed anonymous information storage and retrieval system”, In Designing Privacy Enhancing Technologies: International Workshop on Design Issues inAnonymity and Unobservability, LNCS 2009, New York, 2001.

[Cox and Noble 2002] L. P. Cox and B. D. Noble, “Pastiche: making backup cheap and easy”, in Proceedings of the Fifth USENIX Symposium on Operating Systems Design and Implementation, Boston, MA, December 2002.

[Dingledine 2000] R. Dingledine, “The Free Haven project: Design and deployment of an anonymous secure data haven”, Master’s thesis, MIT, June 2000.

[Druschel and Rowstron 2000] P. Druschel and A. Rowstron, “PAST: A large-scale, persistent peer-to-peer storage utility”, in Proceedings of HotOS VIII, May 2001.

[Jakobsson et al. 2003] M. Jakobsson, J.-P. Hubaux, and L. Buttyan, “A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks”, In Proceedings of Financial Crypto, La Guadeloupe, Jan. 2003.

[Jøsang and Ismail 2002] A. Jøsang and R. Ismail. The Beta Reputation System. In Proceedings of the 15th, Bled Electronic Commerce Conference, Bled, Slovenia, June 2002.

[Jøsang et al. 2005] A. Jøsang, R. Ismail, and C. Boyd, “A Survey of Trust and Reputation Systems for Online Service Provision”, In Proceedings of Decision Support Systems, 2005.

[Kamvar et al. 2003] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The EigenTrust Algorithm for Reputation Management in P2P Networks”, In Proceedings of the Twelfth International World Wide Web Conference, Budapest, May 2003.

[Killijian et al. 2004] M.O. Killijian, D. Powell, M. Banâtre, P. Couderc, Y. Roudier, Collaborative Backup for Dependable Mobile Applications. In proceedings of the 2nd International Workshop on Middleware for Pervasive and Ad-Hoc Computing, Middleware 2004, Toronto, Ontario, Canada, October 18th - 22nd, 2004, ACM.

[Killijian et al. 2006] M.-O. Killijian, M. Banâtre, C. Bryce, L. Blain, P. Couderc, L. Courtès, Y. Deswarte, D. Martin-Guillerez, R. Molva, N. Oualha, D. Powell, Y. Roudier, I. Sylvain, “MoSAIC: Mobile System Availability Integrity and Confidentiality”, Progress Report, June 2006.

[Klemm et al. 2003] A. Klemm, C. Lindemann, and O. Waldhorst, “A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks”, Proc. IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall), Orlando, FL, October 2003.

[Kubiatowicz et al. 2000] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, " OceanStore: An architecture for globalscale persistent storage", in Proceedings of the Ninth international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), Nov. 2000.

[Lee et al. 2003] S. Lee, R. Sherwood, B. Bhattacharjee. "Cooperative peer groups in NICE". In INFOCOM'03, April 2003.

[Liang et al. 2006] J. Liang, R. Kumar, and K.W. Ross, "The FastTrack overlay: A measurement study", Computer Networks, 50, 842-858, 2006.

[Lillibridge et al. 2003] M. Lillibridge, S. Elnikety, A. Birrell, M.Burrows, and M. Isard, "A Cooperative Internet Backup Scheme", In Proceedings of the 2003 Usenix Annual Technical Conference (General Track), pp. 29-41, San Antonio, Texas, June 2003.

[Loo et al. 2002] B. T. Loo, A. LaMarca, and G. Borriello, "Peer-To-Peer Backup for Personal Area Networks", Intel Research Technical Report IRS-TR-02-15, October 2002.

[Marti et al. 2000] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks", Mobile Computing and Networking 255–265, 2000.

[McCoy 2001] J. McCoy. "Mojo Nation Responds". January 2001. http://www.openp2p.com/pub/a/p2p/2001/01/11/mojo.html.

[Michiardi 2004] P. Michiardi, "Cooperation enforcement and network security mechanisms for mobile ad hoc networks", PhD Thesis, December 14th, 2004.

[Michiardi and Molva 2002] P. Michiardi, and R. Molva, "CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks", CMS'2002, Communication and Multimedia Security 2002 Conference, Portorosz, Slovenia, September 26-27, 2002.

[Montresor et al. 2004] A. Montresor, G. Di Caro, P. E. Heegaard, "Architecture of the Simulation Environment", BISON IST-2001-38923, January 29, 2004.

[Mui et al. 2001] L. Mui, M. Mohtashemi, C. Ang, P. Szolovits, and A. Halberstadt. Ratings in Distributed Systems: A Bayesian Approach. In Proceedings of the Workshop on Information Technologies and Systems (WITS), 2001.

[Oberender et al. 2005] J. Oberender, F. –U. Andersen, H. de Meer, I. Dedinski, T. Hoßfeld, C. Kappler, A. Mäder, and K. Tutschku, "Enabling Mobile Peer-to-Peer Networking. Mobile and Wireless Systems", In Proceedings of Mobile and Wireless Systems, LNCS 3427, Dagstuhl, Germany, January 2005.

[Obreiter and Nimis 2003] P. Obreiter & J. Nimis, "A Taxonomy of Incentive Patterns - the Design Space of Incentives for Cooperation", Technical Report, Universität Karlsruhe, Faculty of Informatics, 2003.

[Page et al. 1998] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web”, Technical report, Stanford Digital Library Technologies Project, 1998.

[Papadopouli and Schulzrinne 2001] M. Papadopouli and H. Schulzrinne, “A Performance Analysis of 7DS, A Peer-to-Peer Data Dissemination and Prefetching Tool for Mobile Users”, In Advances in wired and wireless communications, IEEE Sarnoff Symposium Digest, March 2001.

[Schlosser and Kamvar 2002] M. Schlosser and S. Kamvar, “Simulating a file-sharing p2p network”, In Proceedings of the First Workshop on Semantics in P2P and Grid Computing, 2002.

[Terada et al. 2004] M. Terada, M. Iguchi, M. Hanadate, and K. Fujimura. An Optimistic Fair Exchange Protocol for Trading Electronic Rights. In 6th Smart Card Research and Advanced Application conference (CARDIS'2004), 2004.

[Ting and Deters 2003] N. S. Ting, R. Deters, “3LS - A Peer-to-Peer Network Simulator”, IEEE International Conference on Peer-to-Peer Computing, 2003.

[Vogt et al. 2001] H. Vogt, H. Pagnia, and F. C. Gärtner. Using Smart cards for Fair-Exchange. WELCOM 2001, LNCS 2232, Springer, pp. 101-113, 2001. http://citeseer.ist.psu.edu/vogt01using.html

[Zhong et al. 2003] S. Zhong, J. Chen, Y. R. Yang, “Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks”, INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies.