Towards P2P-Based Resource Allocation in Competitive Environments

Yoni Peleg and Jeffrey S. Rosenschein

School of Engineering and Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel
{jonip, jeff}@cs.huji.ac.il

Abstract. Peer-to-peer frameworks are known to be robust and scalable to large numbers of agents. Recent resource allocation studies have leveraged this by using peer-to-peer frameworks for the implementation of resource matching algorithms. In this paper, we present a matching protocol for multiagent resource allocation in a competitive peer-to-peer environment; this work marks the first solution to the resource matching problem in this type of environment. Our approach makes use of micropayment techniques, along with concepts from random graph theory and game theory. We provide an analytical characterization of our protocol, and specify how an agent should choose optimal values for the protocol parameters.

1 Introduction

Resource allocation [1] is a fundamental problem in multiagent systems, allowing agents to share resources so as to complete their tasks. In this paper, we consider one important aspect of the problem, namely the *matching* problem. The goal is to match a supplier (an agent that can provide a resource) to a customer (an agent that needs the resource). We present a protocol that solves this problem in competitive environments; the protocol itself is based on a peer-to-peer (P2P) architecture, along with a central player.

P2P systems have become popular in recent years as a means of disseminating information in distributed environments. An important property of P2P systems is that control is distributed symmetrically among peers, increasing the systems' robustness and scalability.

Many practical P2P solutions are actually based on hybrid architectures: message distribution relies on a distributed peer architecture, while solutions for certain specific problems rely on some centralized entity (e.g., "BitTorrent Trackers" in BitTorrent, and "Rendezvous peers" in JXTA). In this paper we adopt this hybrid approach, and solve certain security problems in our P2P network using a centralized entity.

We assume that all agents in our system are self-interested, and thus concerned with maximizing their private utilities. An agent will not share its resources for free, and it will deviate from our protocol if by doing so it will increase its utility. The assumption of self-interested agents operating in the P2P environment is realistic for many scenarios, but introduces several challenges for incentivizing adherence to the protocol. This work marks the first solution to the resource matching problem in this type of environment.

1.1 Structure of the Paper

We begin our discussion by providing an informal overview of the problem and our solution in Section 2. In Section 3 we explain our approach to the challenges alluded to above, using cryptographic techniques, random graph theory, and game theory. In Section 4 we give a formal description of the protocol, and in Section 5 we present an analysis of it. Section 6 discusses related work, and we conclude in Section 7.

2 An Overview of our Approach

To help ensure that self-interested agents follow our protocol, we incorporate a payment mechanism into it. Each agent in the system has an account with some amount of virtual money. When a customer looks for a resource, it initiates an auction for that resource. Each possible supplier may send a bid to the customer; the customer then selects the best bid.

Our protocol makes use o Vickrey Auctions [2]. As in first price auctions, the supplier that offers the lowest bid wins; however, in Vickrey Auctions the customer pays the winner according to the offer of the second-lowest bid. It is well-known that in Vickrey Auctions, all bidders have a dominant strategy to bid their true private valuations. Additionally, Vickrey Auctions give suppliers an incentive to participate—if a supplier wins the auction, it benefits from the difference between its bid and the second-lowest bid.

To implement the auction, the customer sends agents *Auction Messages* informing them that an auction is taking place. The customer may not know all agents in the P2P network, and relies on its neighbors to keep forwarding the auction messages to *their* neighbors. We call each agent that needs to forward an auction message a *middle agent*; these middle agents will be paid for forwarding auction messages (this compensates middle agents for the effort they expend in transmitting those messages). Payments to middle agents are actually a critical part of the protocol, since each potential supplier has a natural incentive *not* to forward auction messages: the fewer participants in an auction, the higher its probability of winning. Paying for message forwarding is thus a necessity, to incentivize adherence to the protocol. This problem is discussed further in Section 3.4.

2.1 The Search Algorithm

We do not wish to restrict the definition of a resource, but want to allow it to be anything: CPU time, disk space, or even some task that an agent wants to outsource. Thus, we cannot use sophisticated *structured searches* [3] (like Distributed Hash Tables), but rather need to use *unstructured searches*.

Unstructured searches are usually based on flooding techniques or on random-walk techniques (see [4] for an excellent survey). Random-walk techniques are known to have lower message complexity than flooding techniques [5]. We, however, have chosen to use a flooding technique [6]; using this approach gives us a higher probability that more than one path will exist from the customer to a possible supplier. Thus, a potential supplier will not be able, on its own, to block many other possible suppliers. This subject is discussed further in Section 3.4.

Our search algorithm can be considered as an extension to Vanzin's work [7]; the following is a short overview of the algorithm:

1. The customer chooses auction parameters: TTL (Time To Live), l, and $TimeOut$ and it pays for the auction accordingly.
2. The customer forwards auction messages to l of its neighbors (chosen uniformly at random among its neighbors) with the TTL, l parameters attached.
3. Each middle agent subtracts 1 from the message's TTL value, and forwards the message to l of its neighbors. When $TTL = 0$, middle agents no longer forward the message. If the middle agent is a possible supplier, it sends a bid to the customer.
4. The customer waits for $TimeOut$ seconds. It then either chooses the lowest bid, or starts a new auction.

2.2 Challenges to Overcome

The following list describes situations where a rational agent might deviate from the protocol:

- A middle agent gets paid for forwarding an auction message, and then does not forward the message. The customer cannot tell whether a middle agent has forwarded the auction message;
- A middle agent forwards the auction message, but the customer then denies that the auction ever existed, and refuses to pay for it;
- A middle agent falsely claims that another agent is searching for a resource, and charges that other agent for forwarding its auction message;
- A middle agent tries to get paid more than once;
- Using fake IDs—a middle agent may try to add itself more than once to the system, and then make money from forwarding the auction messages to itself;
- The customer claims that the value of the second-lowest bid was lower than it really was, paying the auction winner less.

2.3 Comparison with Existing Solutions

Our solution is a hybrid P2P protocol; to overcome the above difficulties, our solution uses a centralized trusted third party (Section 3.1). The number of messages that this third party entity receives depends only on the number of customers in the system. The number of messages in previous centralized solutions (like centralized match-makers, centralized internet directories, mechanism design algorithms, and "stable marriage problem" algorithms) also depends on the number of suppliers in the system, since each supplier needs to inform the centralized entity about the goods it can supply. Our solution is thus more scalable than those previous centralized solutions, despite our use of a centralized player in the protocol. In addition, our solution does not restrict the definition of a resource. Previous solutions (like centralized match-makers) had to make prior assumptions about the resources' definition.

3 Protocol Incentives

This section describes the incentives added to the protocol in order to overcome the difficulties listed above in Section 2.2. We make the following assumptions: agents are rational, are only aware of their immediate neighbors, and cannot fake identities nor form coalitions. The agents' utilities are determined by the amounts of money they have—each agent receives money for each task it completes, and receives $1 for each message it forwards.

3.1 The Central Bank

Our solution uses a centralized trusted third party called the bank, a secure entity that always follows the protocol. It handles the agents' accounts (that initially contain $0), and is in charge of money transfers. The bank also performs sporadic checks to help ensure that agents won't deviate from the protocol (details below).

3.2 Cryptographic Solutions

Giving incentives to agents to forward messages is a problem that has been studied extensively in recent years in the field of multi-hop cellular networks. In this paper, we follow a micro-payment solution that was presented by Jakobsson [8]. We begin by presenting a naive algorithm that solves the problem.

When an agent receives an auction message, it sends a message to the bank that includes a receipt (a cryptographic-hash signature of the message), and the ID of the agent from which it got the message (the previous agent in the search path). The bank pays $1 to the agent and $1 to the previous agent in the search path. In this way, each agent has an incentive to forward the message, since once the next agent gets the message the sending agent will also get paid (in its role as the previous agent in the search path).

The naive approach has a clear disadvantage: the bank is overloaded with messages. Jakobsson's solution to this problem is to use *Lottery Tickets*: each time an agent enters the system, it receives from the bank a private random integer s_i. Each time a customer starts an auction, it receives from the bank a random integer L which is the "lottery ticket". The bank signs L using its private RSA key [9], so that each agent can verify the signature. When a middle agent gets an auction message, it can check with a known cryptographic-hash function H whether it won the lottery (for example, the agent wins the lottery if and only if $H(s_i, L) = 1$). An agent contacts the bank only if it won the lottery.

We can set the probability of winning to be small enough, so that the expected number of times the bank will be contacted in an auction will be 2. Once an agent wins the lottery, the bank gives it a very high payment, so in expectation the agent gets that same \$1, as it got in the naive solution above. If the agent participates in many auctions, the average payment for forwarding an auction message will be close to the expectation.

Since we are using a cryptographic hash function, an agent cannot reveal another agent's secret number from previous auctions. Therefore, a middle agent cannot know in advance if the next agent will win the lottery, and thus it has an incentive to forward the message.

3.3 Solutions Based on the Bank

In the cryptographic solution from Section 3.2, the bank does not keep track of the entire message distribution process. Instead, the bank is prompted only when an agent wins the lottery. This gives rise to some additional challenges that our protocol must confront: agents may continue to forward the auction message when the TTL parameter is 0; they may send the auction message to more than l of their neighbors; they may lie about the identity of the previous agent in the search path. All these problems can be solved using cryptographic techniques; the bank needs to run sporadic checks to make sure that agents do not deviate in these ways (if the bank detects that an agent has deviated, it punishes that agent). We omit details due to lack of space.

3.4 Game Theory Solutions

A Two-Player Coordination Game Consider the example network in Figure 1. The network contains four agents: a, b, c, and d. a is a customer that starts an auction for a resource. b, c, and d may supply the resource. Since a possible supplier will make a profit from winning an auction, b and c have a dilemma: if any of them forward the auction message, all three suppliers will participate in the auction; the probability of each of the agents to win decreases (and thus the expected profit from participating in the auction also decreases). If both of them do not forward the auction message, only b and c will participate in the auction, and their expected profit from participating in the auction will increase. b is worse off if c forwards the auction message while it does not—this way it will not get money for forwarding the auction message, and it will not increase its expected profit from participating in the auction. c has a symmetric situation. Thus we see that b and c are playing a coordination game: each of them can Cooperate (forward the auction message) or Defect (deviate from the protocol).

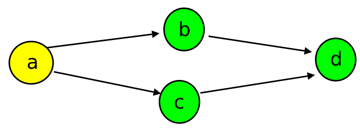

Fig. 1. A Sample Network

The Distribution of Private Valuations over Resources To make the discussion above more concrete, we give a formal description of the agents' private valuation of resources. We assume that the agent's private valuation of a resource r is drawn i.i.d. from a distribution D_r. Thus, if there are m possible suppliers, each of them has probability $\frac{1}{m}$ of winning the auction.[1] The winner is expected to offer the first-order statistic of the distribution D_r (given m participants); its expected profit will be the difference between the second and the first-order statistic.

To simplify our presentation, for the rest of the paper we take D_r to be the continuous uniform distribution on $[A_r, B_r]$. Nevertheless, all the calculations we present can be applied to any arbitrary distribution. We use the following known theorem [10]:

Theorem 1. *Let $X_1, X_2, \ldots, X_n$ be n random variables that were drawn from a continuous uniform distribution on the section $[A, B]$. Let Y_k be the k-th statistic order of $(X_i)_{i=1}^{n}$ (i.e., the k-th smallest value that was drawn). Then:*

$$E\left(Y_k\right) = A + (B - A)\frac{k}{n+1}$$

Furthermore, throughout the rest of the paper we assume that $A_r = 0$. A different value of A_r simply means that a customer would need to pay at least A_r dollars in order to use the resource; it will not influence the expected profit of the suppliers, nor the customer's search strategy. Assuming that $A_r = 0$, both the expected bid of the winner and the expected profit of the winner are $\frac{B_r}{m+1}$.

Analysis of the 2-Player Coordination Game Using the continuous uniform distribution gives the game in Table 1 (below) for agents b and c in the Figure 1 example.

If $B_r \leq 12$, then by Table 1 both agents have the dominant strategy of following the protocol. Unfortunately, usually we would expect the price of a resource to be much larger than the cost of forwarding a single message. If $B_r > 12$, we will get a coordination game where both CC and DD are Nash equilibria.[2]

[1] We make a simplifying assumption that agents are not familiar with D_r, and thus cannot make a better estimation of their probability of winning.

[2] For the rest of the paper we only consider ex-ante Nash equilibria (we make the simplifying assumption that agents have full knowledge of the network structure, and of other suppliers, a standard assumption, e.g., in papers on "byzantine and rational" network behavior [11]). Evaluating ex-post Nash equilibria is left for future research.

	C	D
C	(B/12)+1 (B/12)+1	B/12 (B/12)+1
D	(B/12)+1 B/12	B/6 B/6

Table 1. The Sample Network Game Table

The DD strategy is Pareto optimal. Recall that we assume no coalitions, so agents b and c cannot coordinate with one another.

Boyer [12] has shown analytically that without prior conventions that help the agents coordinate, the system might converge to either a CC or DD equilibrium (even though DD is Pareto optimal). Cooper [13] got the same results empirically (in experiments with humans). Furthermore, the following conventions might help the system converge to the CC equilibrium in practice:

– A rational agent might prefer a *Minimax* strategy that promises it the $1 payment, rather than playing a strategy that in the worst case would leave it with nothing.
– An agent that is not a supplier has a dominant strategy to cooperate. This may provide a convention in favor of the CC equilibrium for unpopular resources.
– It is possible that there will be more than one path from the customer to the supplier (e.g., from a to d in Figure 1). In such a case, it is enough that the agents in one of the paths cooperate so as to turn cooperation into the best strategy for other agents.
– If all agents do not cooperate, the system will eventually crash. This can provide the agents a prior-convention to cooperate [14].

Multi-Player Coordination Games In the network of Figure 1, we have a relatively simple coordination game with only two participants. Real networks are obviously far more complicated. As in the two-player game, multiple-player coordination games [15] may have more than one Nash equilibrium, and without further assumptions one cannot give a mathematical argument regarding which of the equilibria the system will converge to (even when one of the equilibria is Pareto optimal). We will show in Section 5.3 that in some cases, it is possible to derive sufficient conditions for cooperation even without further assumptions.

If one accepts our hypothesis that the conventions that were mentioned above will cause the agents to play a minimax strategy, stronger sufficient conditions can be deduced. Let T be a random number that represents the total number of agents that will accept the auction message if an agent cooperates. Let V be a random number that represents the total number of agents that will *surely* be blocked if the agent deviates (i.e., all the paths of length at most TTL to those V agents pass through the agent). According to our hypothesis, the agent will surely cooperate if: $B * \frac{2*V*T-V^2}{(T-V)^2*T^2} \geq 1$. The values of T and V depend on the random graph model of the network. We show in Section 5.3 how these values can be approximated using "node coverage".

Why Don't We Use Payments Instead? Could we not simply pay more to middle agents for forwarding messages, such that cooperation becomes a dominant strategy? Unfortunately, that approach increases the cost of each auction significantly, and reduces the probability of finding a resource. The details are omitted due to lack of space.

4 Our Protocol

We now introduce our detailed protocol (Algorithm 1, below) for carrying out an auction.

Algorithm 1 The protocol that is executed when a customer v_i holds an auction for a resource

1: The customer sets the auction parameters: $TimeOut$, l, and TTL. It connects to the bank and informs the bank that it wants to start an auction for a resource. The bank calculates the expected payment and charges the customer.
2: The bank generates a random number (the lottery ticket) L, signs it using its private key, and sends the signed message to the customer.
3: The customer sends the auction message (including the parameters and L) to l of its neighbors.
4: Each middle agent j that receives the auction message verifies the bank's signature on L; it checks if it won the lottery (if so, it informs the bank); it checks if it is a possible supplier (and submits a bid if it is). Eventually, it decreases the TTL value by 1 and forwards the message to l of its neighbors.
5: If the bank receives a winner announcement, it verifies that the agent indeed won, and it pays the agent and the previous agent in the search path. The bank does sporadic checks that the win is legitimate, i.e., the agent is indeed at distance TTL from the customer (otherwise, the bank punishes the previous agent in the search path).
6: The customer waits $TimeOut$ seconds, and then informs the bank of the best two bids. It uses the resource of the winner. The bank charges the customer and pays the winner.

4.1 Optimal Parameters for an Auction

The customer needs to choose three parameters for each auction: TTL, $TimeOut$, and l. Vanzin [7] describes how to choose the optimal value for the $TimeOut$ parameter. A rational customer always chooses l to be as large as possible,[3] since a

[3] We assume that the degree of the vertices in the P2P network is bounded by a constant; otherwise, we would have broadcast capabilities.

larger l implies that each agent can block a smaller group of agents by itself. The optimal value of the TTL parameter depends on the level of cooperation among agents. Let us assume that all agents cooperate. We follow Wu's definition [5] of *node coverage* and let $N(j)$ be a random number that represents the number of agents at distance at most j from the customer. The optimal TTL is the value that minimizes the following expression: $|N(TTL) - \frac{\sqrt{2(B_r - A_r) * P_r} - 1}{P_r}|$.

The above expression holds if all agents follow the protocol. In Section 5.3 we show that this expression is correct in many cases even if the agents are simply rational. In Section 5.2 we give expressions for the node coverage ($N(j)$) of two popular random graphs.

4.2 Known Disadvantages

Our protocol does not demonstrate "fairness": if an agent is slow, it may not hear about/participate in an auction before the auction is over. Our protocol is also not budget-balanced. Since the environment is asynchronous, we cannot know in advance how many agents will participate in an auction (even if we have full knowledge of the network, and even if all agents cooperate). Thus, the customer might be overcharged for an auction. Furthermore, we cannot return the overcharged money back into the environment, since that would give agents an incentive to deviate from the protocol. The details are omitted due to lack of space.

5 Analysis of the Protocol

We here present a short analysis of the protocol. The analysis considers two graph models [16]: the *d-out* model, and the random geometric graph (RGG) model. We assume that each agent holds a resource with equal probability p. Due to lack of space, we omit proofs of theorems and most simulation results.

5.1 Graph Models

In the random d-out model, each vertex is connected to exactly d neighbors that are chosen uniformly. Duplicate edges and self-loops are not allowed. In the RGG model, the vertices' locations are distributed uniformly on an $L \times L$ board. There is an edge between two agents if and only if the distance between them is smaller than r.[4]

5.2 Node Coverage

The following two theorems approximate the node coverage in the d-out and RGG models:

[4] We use the L_∞ norm in this presentation because it enables cleaner expressions.

Theorem 2. *Node coverage in the d-out model is (where $L_{d,n} = log_d(\sqrt{n})$):*
$E(X_j) \geq d^j$ *if* $0 \leq j \leq L_{d,n}$
$E(X_j) \geq \sqrt{n} * (d - \frac{2}{d})^{j-L_{d,n}}$ *if* $j \leq log_d(n) - 2$
$E(X_j) \geq \sqrt{n} * (\frac{d^2-2}{d})^{j-L_{d,n}}((d-2) * (1 - e^{-1})$ *if* $j < 2L_{d,n}$
with high probability and $E(X_j) \leq d^j$*,*
where $X_j = N(j) - N(j-1)$*.*

Theorem 3. *Let $G(n, r, L)$ be an RGG where $L = 1$, $4r^2n = d$, $d \geq 30$. Let $j < \frac{L}{2r}$. Then with high probability:* $\frac{n}{4} * (2 * j * (r - \frac{r}{\alpha}))^2 \leq N(j) \leq n * (2 * j * r)^2$.

To evaluate our approximation, we compared it with simulation results. The simulation and analytic results were quite similar (9% average error on the RGG model, with standard deviation of 1%, 6.38% average error on the d-out model, with standard deviation of 0.01%). Our approximation is tighter than the approximation given by Wu [5].

5.3 Agent Cooperation

In the previous section, we demonstrated how to calculate the optimal TTL value for the case where all agents follow the protocol. A natural question that arises is, if we use that TTL value, will rational agents follow the protocol?

Theorem 4 gives a sufficient condition for cooperation without using assumptions about prior conventions. It relies on the fact that agents that cannot supply the resource have a dominant strategy to follow the protocol. Thus, agents cooperate if the resource is not popular.

Theorem 4. *A sufficient condition for cooperation in the d-out graph model is that:* $\frac{2}{d(d-1)^2} \leq \frac{p^2[d(1-p)]^{2TTL-2}}{B_r}$.

Unfortunately, our simulation results show that for popular resources, if we assume that there are no prior conventions then the optimal TTL for the auction is larger than the optimal TTL for a cooperative environment.

The next two theorems assume that agents have a prior convention to play minimax (Section 3.4).

Theorem 5. *In the d-out model, with high probability all agents will cooperate if:* $\frac{2}{d}\left(B_r + P_r^2\sqrt{\frac{2B}{P}}\right) exp\left(-\frac{1}{n}\sqrt{\frac{2B_r}{P_r}}\right) \leq 2B_r * P_r$.

Theorem 6. *In the RGG model, if $r \geq \sqrt{\frac{30L^2}{4n}}$ then all agents will follow the protocol.*

The idea behind Theorem 5 is that if the resource is expensive enough or if the resource is not popular, the customer will choose an auction with a large TTL. Thus, the number of possible suppliers that can be effectively blocked by a single agent is small, and the minimax strategy will cause the agents to cooperate. Clearly, if the resource is cheap, cooperation also becomes a dominant strategy.

That leaves us with an (exponentially small) gap of intermediate prices, where an agent may deviate from the protocol. Simulations on the d-out model show that even for "problematic" intermediate prices, in about 95% of auctions all the agents in the system cooperate (recall that Theorem 5 gives only sufficient conditions for cooperation). From Theorem 6 we see that in the RGG model there is a "connectivity gap"; above it, agents will always follow the protocol. Unfortunately, the message complexity of our protocol in an RGG is much higher than in the d-out model.

6 Related Work

Chevaleyre et al. [1] is an excellent resource allocation survey; here, we briefly mention some related work on resource allocation matching. Much prior research used a centralized agent (e.g., [17]), which causes scalability problems. Mullender and Vitanyi [18] used a distributed matching process, and proved a $\sqrt{n}$ lower bound for the number of agents with which each matchmaker is familiar, assuming that matchmakers cannot pass information among themselves; thus, that work also has scalability problems when n is very large. Recent studies (e.g., Vanzin [7], Ogston and Vassiliadis [19]) have used a P2P matchmaking framework. Unlike our work, they assume a cooperative environment, and identical resources. The well-known Contract Net protocol [20] used auctions for matchmaking, and was later extended to non-cooperative environments. However, the Contract Net relied on broadcasts for carrying out auctions, which is very inefficient if a natural broadcast channel does not exist.

7 Conclusions

We have presented a protocol that addresses the resource allocation matching problem. The protocol is based on a P2P framework with a centralized bank; the central bank plays a role different in our protocol from its role in previous solutions, allowing our technique to be more scalable than those previous approaches.

The protocol is designed for non-cooperative environments—incentives are provided for the agents to follow the protocol, and we showed sufficient conditions that ensure that the incentives will indeed cause all agents to follow the protocol.

Three important topics are open for future work. First, simulation results show that our protocol withstands coalitions of constant size; a formal analysis of this remains open. Another interesting question is whether we can distribute the role of the central bank, doing away with the one centralized aspect of our solution (a similar approach was taken in [21]). Finally, dealing with "whitewashing" (detecting and punishing participants that use fake IDs) is one of the most important open problems in P2P research.

References

1. Chevaleyre, Y., Dunne, P., Endriss, U., Lang, J., Lemaitre, M., Maudet, N., Padget, J., Phelps, S., Rodriguez-Aguilar, J., Sousa, P.: Issues in multiagent resource allocation. Informatica **30** (2005)
2. Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders. The Journal of Finance **16**(1) (1961) 8–37
3. Risson, J., Moors, T.: Survey of research towards robust peer-to-peer networks: search methods. Comput. Netw. **50**(17) (2006) 3485–3521
4. Li, X., Wu, J.: Searching techniques in peer-to-peer networks. In: Handbook of Theoretical and Algorithmic Aspects of Ad Hoc, Sensor, and Peer-to-Peer Networks. (2006) 1–28
5. Wu, B., Kshemkalyani, A.D.: Analysis models for blind search in unstructured overlays. In: NCA. (2006)
6. Kalogeraki, V., Gunopulos, D., Zeinalipour-Yazti, D.: A local search mechanism for peer-to-peer networks. In: CIKM. (2002) 300–307
7. Vanzin, Barber: Decentralized partner finding in multiagent systems. Coordination of Large-Scale Multiagent Systems (2006)
8. Jakobsson, M., Hubaux, J., Buttyan, L.: A micropayment scheme encouraging collaboration in multi-hop cellular networks (2003) In Proceedings of Financial Crypto.
9. Rivest, R., Shamir, A., Adelman, L.: A method for obtaining digital signatures and public-key cryptosystems (1977)
10. Ahsanullah, M.: Order Statistics. Nova (2005)
11. Clement, A., Li, H., Napper, J., Martin, J., Alvisi, L., Dahlin, M.: BAR Primer. In: IEEE/IFIP DSN-2008. (June 2008) 287–296
12. Boyer, R., Orlean, A.: How do conventions evolve? Journal of Evolutionary Economics **2**(3) (October 1992) 165–77
13. Cooper, R., De Jong, D.V., Forsythe, R., Ross, T.W.: Forward induction in coordination games. Economics Letters **40**(2) (October 1992) 167–172
14. Feldman, M., Papadimitriou, C., Chuang, J., Stocia, I.: Free-riding and whitewashing in peer to peer systems. In: SigCom '04: Workshop, Portland, Oregon, USA (2004)
15. Huyck, J.B.V., Battalio, R.C., Beil, R.O.: Tacit coordination games, strategic uncertainty, and coordination failure. The American Economic Review **80**(1) (1990) 234–248
16. Zegura, E.W., Calvert, K.L., Bhattacharjee, S.: How to model an internetwork. In: INFOCOM (2). (1996) 594–602
17. Bertels, K., Panchanathan, N., Vassiliadis, S., Ebrahimi, B.P.: Centralized matchmaking for minimal agents. In: ICPADS. (2004)
18. Mullender, S.J., Vitanyi, P.M.B.: Distributed match-making. Algorithmica **3**(1) (June 1987) 367–391
19. Ogston, E., Vassiliadis, S.: Matchmaking among minimal agents without a facilitator. In: International Conference on Autonomous Agents. (2001)
20. Smith, R.G.: The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Transactions on Computers **12** (December 1980) 1104–1113
21. Zhong, S., Chen, J., Yang, Y.: Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. INFOCOM 2003. (2003)