Auction-Based Resource Allocation in Digital Ecosystems

Moreno Marzolla, Stefano Ferretti, Gabriele D'Angelo
Department of Computer Science and Engineering, University of Bologna
Mura A. Zamboni 7, I-40127 Bologna, Italy
Email: moreno.marzolla@unibo.it, sferrett@cs.unibo.it, g.dangelo@unibo.it

Abstract—The proliferation of portable devices (PDAs, smart-phones, digital multimedia players, and so forth) allows mobile users to carry around a pool of computing, storage and communication resources. Sharing these resources with other users ("Digital Organisms" – DOs) opens the door to novel interesting scenarios, where people trade resources to allow the execution, anytime and anywhere, of applications that require a mix of capabilities. In this paper we present a fully distributed approach for resource sharing among multiple devices owned by different mobile users. Our scheme enables DOs to trade computing/networking facilities through an auction-based mechanism, without the need of a central control. We use a set of numerical experiments to compare our approach with an optimal (centralized) allocation strategy that, given the set of resource demands and offers, maximizes the number of matches. Results confirm the effectiveness of our approach since it produces a fair allocation of resources with low computational cost, providing DOs with the means to form an altruistic digital ecosystem.

Keywords—*Resource Allocation, Optimization, Peer-to-Peer Systems, Ad-hoc Networks*

I. INTRODUCTION

Mobile users are evolving: while in the recent past people used their mobile devices just for "simple" tasks such as checking e-mail or browsing the Web, the rise of novel social applications fosters a massive use of ubiquitous services. Current Operating Systems for mobile devices (e.g., Android, iOS, Windows Phone) allow the execution of applications that, for instance, publish user's geographical position and other context-aware information on social networking services. However, these forms of interaction are usually based on the classic client-server approach, i.e., the mobile device connects to a central service through its own Internet connection. The proliferation of heterogeneous devices with different capabilities (computation, communication, data storage, sensors and actuators) gives rise to new scenarios that promote the cooperation among individuals in order to guarantee the provision of "always on" services [1].

As an example, consider a medical doctor who receives an urgent call while attending a meeting with other colleagues. He needs to check some medical data to diagnose a particular illness; unfortunately, the tablet PC he carried with him has not enough memory and computational power to execute the job. Therefore, he "rents" CPU power from one of his colleagues high-end laptop to carry on the analysis.

As another example, consider a user that wants to upload some pictures made with her smart-phone/camera on Flickr or on the wall of her social Web application, but she is not provided with network connectivity. Hence she exploits the 3G mobile connection of a neighboring friend using that as a gateway to the net via an ad-hoc short range connection (e.g., Bluetooth).

In general, mobile users have many different devices in their pockets and suitcase, each of them with specific hardware and software characteristics. Quite often such devices are not enabled for seamless interaction with other devices belonging to the same owner; sharing resources among different people is even more challenging. The possibility for a user to exploit, in a Peer-to-Peer (P2P) and altruistic way, computing facilities owned by (known and trusted) neighbors requires mechanisms for automatic service discovery and negotiation, and for access control.

A system architecture supporting the scenarios above has been proposed [1]. Each mobile user is considered as a "digital organism" (DO), composed by many different devices belonging to the same human being. Each DO may share resources with peer DOs using auto-configuration strategies. Then, a community of interacting DOs can be thought of as a "digital ecosystem". Each DO in the ecosystem contributes by providing its own unused resources to its (friend/trusted) neighbors. Therefore, the community of DOs exploits self-organization protocols and P2P strategies to create a cooperating, altruistic ecosystem.

Of course, privacy and security issues should be carefully considered, especially if data are to be distributed to other users. Addressing these issues in a mobile environment is highly nontrivial, especially when the remote nodes are completely untrusted. Therefore, we assume that each DO will preferably connect to other DOs towards which there is explicit trust (e.g., because users know each other). For example, a user traveling by train shall be willing to share a network connection with some (known and trusted) traveling companions.

In this paper we address the problem of optimizing the allocation of resources in a digital ecosystem, by matching resource requests and offers. We consider a P2P overlay which connects DOs; each DO may offer and/or require resources, which are traded with neighboring DOs. We present a fully distributed scheme that can match demands and offers, allowing resources to be provisioned efficiently and fairly. We use a market-based approach in which requests are handled through ascending clock auctions [2]. We assume that each DO can use some form of "virtual currency" (tokens) as a form of payment for resources usage: this allows a fair

allocation that balances supply and demand [3], [4]. First, we describe a distributed algorithm to carry on the auction; then, we formulate the resource allocation problem as a Mixed Integer Programming (MIP) optimization problem, which is used to compute the maximum number of requests that can be matched. We compare the optimal allocation with the one provided by our our cheap, local strategy. Results show that our approach produces good allocations and requires low computational cost: this makes the auction-based allocation strategy particularly appropriate for sharing resources among devices with very limited computational power.

The rest of this paper is organized as follows. In Section II we give a precise formulation of the problem we are addressing. Section III presents our auction-based resource allocation scheme. In Section IV we discuss numerical results obtained from a set of synthetic simulation experiments. In Section V we revise the literature and contrast our approach with some related work. Finally, concluding remarks are provided in Section VI.

II. PROBLEM FORMULATION

We consider a set $\mathbf{R} = \{1, \ldots, R\}$ of R different resource types (e.g., network connectivity, processing power, storage, and so on). $\mathbf{N} = \{1, \ldots, N\}$ denotes a set of N users trading these resources: each user can be either a *buyer* (if he requests resources) or a *seller* (if he offers resources). The same user may play the role of buyer and seller at the same time, offering surplus resources while buying those he needs.

For each user $i \in \mathbf{N}$, we denote with Req_{ri} the amount of type r resource requested by i; for each $j \in \mathbf{N}$, we denote with Off_{rj} the amount of type r resource offered by j; quantities are not restricted to be integer. The vectors $Req_{\bullet i}$ and $Off_{\bullet j}$ are called *resource bundles*[1].

As an example, if there are two resource types ("CPU" and "Network Bandwidth"), then a resource bundle $(0.1 \text{ MIPS}, 200 \text{ KB}/s)$ can be interpreted as a request (or an offer) for 0.1 MIPS of CPU power and 200 KB/s of network bandwidth. Unit of measures will be omitted in the following.

We assume that each user (node) is equipped with some form of wireless connectivity which enables short range interaction with a set of neighbors using an ad-hoc network topology. We model this with an $N \times N$ adjacency matrix M_{ij}, where users i and j can interact iff $M_{ij} = M_{ji} = 1$; matrix M_{ij} is symmetric, so that interactions are always bidirectional.

Each user can get resources from, or provide resources to, one of his direct neighbors; multi-hop interactions are not allowed. Multi-hop interactions would be much harder to handle, since appropriate routing strategies should be employed to ensure connectivity in spite of individual users moving and losing contact with neighbors. We introduce the binary decision variable X_{irj} which equals 1 iff user i obtains resource r from user j. If $X_{irj} = 1$, then i must obtain exactly Req_{ri} items of resource r from j. The allocation X_{irj} must satisfy the following constraints:

---

[1]We use the symbol $\bullet$ as a shortcut to denote a slice of a multi-dimensional vector; therefore, $Req_{\bullet i}$ denotes the slice $(Req_{1i}, Req_{2i}, \ldots, Req_{Ri})$

TABLE I.   NOTATION USED IN THIS PAPER

$\mathbf{N} :=$	$\{1, \ldots, N\}$ set of users	
$\mathbf{R} :=$	$\{1, \ldots, R\}$ set of resource types	
$Req_{ri} :=$	Amount of resource r requested by user i	
$Off_{rj} :=$	Amount of resource r offered by user j	
$RP_i :=$	Reserve price of buyer i: maximum amount i is willing to pay for his requested bundle	
$SP_{rj} :=$	Unitary selling price of resource r offered by user j	
$M_{ij} :=$	1 iff i can interact with j	
$X_{irj} :=$	1 iff i obtains resource type r from j	

1) Each buyer i must obtain the requested quantities Req_{ri} of all resources r in his bundle, or none at all. Partially fulfilled requests are not allowed.
2) For each r, the requested quantity Req_{ri} must be provided by a single seller j (if the request is satisfied at all).
3) For each r, the offered quantity Off_{rj} can be fractioned across multiple buyers (i.e., a seller is not forced to provide all items of resource r to a single buyer).
4) If user i gets resource r from j, then the amount requested by i must not exceed the amount offered by j: $Req_{ri} \leq Off_{rj}$.
5) For all $r \in \mathbf{R}$, $\sum_{i \in \mathbf{N}} Req_{ri} X_{irj} \leq Off_{rj}$, where the left-hand side represents the total amount of resource r provided by seller j. This means that the total amount of resource r provided by j to all buyers must not exceed his capacity.
6) X_{irj} can be 1 only if $M_{ij} = 1$: interactions are only allowed between neighbors.

The notation used in this paper is summarized in Table I (additional symbols shown in the table will be introduced in the next section).

The problem of finding an optimal allocation X_{irj} which maximizes the number of matched requests (i.e., maximizing the number of requests which can be satisfied by some seller) can be formulated as a MIP optimization problem, as follows:

Given:

$$\mathbf{N} := \{1, \ldots, N\} \text{ Set of users}$$
$$\mathbf{R} := \{1, \ldots, R\} \text{ Set of resource types}$$
$$Req_{ri} := \text{Amount of resource } r \text{ requested by } i$$
$$Off_{rj} := \text{Amount of resource } r \text{ offered by } j$$
$$M_{ij} := 1 \text{ iff user } i \text{ can interact with user } j$$

Define:

$$X_{irj} = 1 \text{ iff user } i \text{ gets resource } r \text{ from user } j$$

Maximize:

$$\sum_{i \in \mathbf{N}} \sum_{r \in \mathbf{R}} \sum_{j \in \mathbf{N}} X_{irj} \tag{1}$$

Subject to:

$$\sum_{i \in \mathbf{N}} Req_{ri} X_{irj} \leq Off_{rj} \qquad r \in \mathbf{R}, j \in \mathbf{N} \qquad (2)$$

$$\sum_{j \in \mathbf{N}} X_{irj} \leq 1 \qquad i \in \mathbf{N}, r \in \mathbf{R} \qquad (3)$$

$$\sum_{j \in \mathbf{N}} X_{irj} = \sum_{j \in \mathbf{N}} X_{i1j} \qquad i \in \mathbf{N}, r \in \mathbf{R} \qquad (4)$$

$$X_{irj} \leq M_{ij} \qquad i \in \mathbf{N}, r \in \mathbf{R}, j \in \mathbf{N} \qquad (5)$$

The optimization problem above is a MIP problem since it involves binary decision variables X_{irj}.

Constraint (2) ensures that the total capacity of each seller j is not violated: the *total* amount of type r resource provided by j to all other users must not exceed its capacity Off_{rj}. Constraint (3) requires that each user i acquires resource type r from at most a single provider j; note that i may be unable to acquire any resource at all, so the constraint is an inequality rather than an equality. Constraint (4) requires that, for each user i, either all the resources it needs are obtained, or none at all. Finally, constraint (5) requires that user i can request resources from user j ($X_{irj} = 1$) only if i and j are neighbors ($M_{ij} = 1$).

From constraint (4) we have that

$$\sum_{r \in \mathbf{R}} \sum_{j \in \mathbf{N}} X_{irj} = 0 \text{ or } R, \quad \text{for all } i \in \mathbf{N}$$

therefore, the total number of matches, i.e., the total number of buyers which can be satisfied, is

$$\sum_{i \in \mathbf{N}} \sum_{r \in \mathbf{R}} \sum_{j \in \mathbf{N}} \frac{X_{irj}}{R} \qquad (6)$$

Since R is a constant, each assignment of X_{irj} maximizing (6) also maximizes the somewhat simpler expression (1) which we use as objective function.

Unfortunately, solving the optimization problem is impractical for several reasons: (i) global knowledge of all parameters is required, whereas each peer has only local knowledge; (ii) solving large instances of the optimization problem using general-purpose MIP solvers is time cosuming; (iii) the optimal allocation may not even be desirable, since the constraints above do not take into account any measure of fairness between users. The lack of fairness is a particularly serious limitation, since it gives users no incentive to share their resources. While it would be possible to extend the optimization problem to take fairness into account, the other limitations would still apply.

Therefore, in the next section we propose a distributed algorithm for binding resource requests with resource availability; our algorithm is fully decentralized and uses an economic model based on ascending auctions; "virtual currency" is used to compensate transactions and stop free riders.

III. AUCTION-BASED RESOURCE ALLOCATION

We propose a fully distributed algorithm to compute a fair allocation of the resources offered by sellers. The algorithm is lightweight and fully decentralized, since it will be executed on
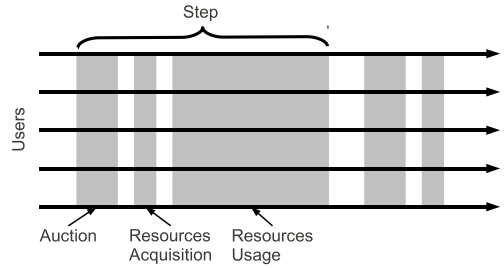

Fig. 1. Auction-based resource allocation

portable devices (smartphones, PDAs, notebooks) connected through an ad-hoc network infrastructure.

Our algorithm is based on ascending clock auctions [2]; in this type of auctions, the auctioneer defines the price of goods, and the bidders reply with the quantities they want to buy. In case of excess demand, the auctioneer raises the price and calls for new bids. This mechanism is iterated until there is no excess demand. Successful bidders pay the last announced price. In our scenario, each seller engages an auction with all potential buyers and adjusts the resource prices independently of other sellers.

Resource allocation works as a sequence of steps, as depicted in Figure 1. Each step is divided into three phases. During the auction phase, all users engage an auction which consists of multiple rounds; during each round, sellers broadcast the unitary prices and quantities of the resources they offer, and potential buyers place bids. The sellers which receive excess demand raise the selling price and start a new round. At the end of the auction, successful buyers pay their dues to the sellers and can access the provided resources. At this point, usage is granted for some amount of time, after which a new step is initiated.

Each user has some amount of virtual currency (tokens), which can be spent to acquire resources provided by other peers, or earned by selling access to his resources. In the following we assume that an adequate scheme exists to employ virtual concurrency; dealing with the details is out of the scope of this paper, since several proposals exist such as those described in [5], [6], [7], [8].

Each node i autonomously computes a slice of the allocation matrix X_{irj}, i.e., decides from which sellers to buy the resources in his request bundle. Each seller j broadcasts the vector $Off_{\bullet i} = \left(Off_{1j}, \ldots, Off_{Rj} \right)$ of the amounts of resources offered, together with the unitary selling prices $SP_{\bullet j} = (SP_{1j}, \ldots, SP_{Rj})$. Each seller can define initial prices at the beginning of an auction phase; such prices are called *reserve prices*, and represent the minimum prices at which a seller is willing to offer his resources.

Each buyer i can place *bids* to all sellers j in his neighborhood from which he wants to acquire resources. A bid is a R-dimensional vector with elements $Req_{ri} X_{irj}$. Each element of the bid is a proposal to acquire $Req_{ri} X_{irj}$ items of resource r from seller j at the unitary price SP_{rj}. Buyer i places a bid for resource r of j provided that (i) $Off_{rj} \geq Req_{ri}$ (the amount requested does not exceed the amount offered), and (ii) the unitary price SP_{rj} of seller j is the minimum over all
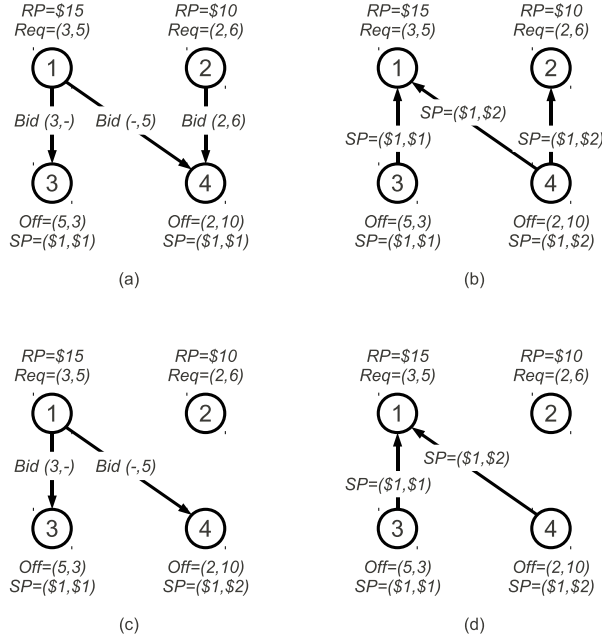
Fig. 2. Example of resource allocation through ascending auction

potential sellers. Each buyer i tries to acquire resource r from the seller providing a sufficient quantity, at the lower price.

Since there is a finite amount of each resource type, it is necessary to handle the situation in which the demand is larger than the supply provided by a seller. We employ a mechanism based on *ascending auctions*, which eventually produces an allocation X_{irj} satisfying the constraints described in Section II.

If all bids placed by i were to be accepted from the sellers, the total cost for the buyer would be

$$\sum_{r \in \mathbf{R}} \sum_{j \in \mathbf{N}} SP_{rj} Req_{ri} X_{irj}$$

Each buyer i has a maximum reserve price RP_i, which represents the total maximum amount of money he is willing to spend for the bundle $Req_{\bullet i}$[2]. Buyer i places bids as long as the cost (III) is not greater than his reserve price RP_i.

Each seller j collects all bids he receives, and replies back to the buyers with a new vector of (possibly updated) prices $\left(SP'_{1j}, \ldots, SP'_{Rj}\right)$, where:

$$SP'_{rj} = \begin{cases} SP_{rj} & \text{if } \sum_{i \in \mathbf{N}} Req_{rj} X_{irj} \leq Off_{rj} \\ SP_{rj} + \Delta P & \text{otherwise} \end{cases}$$

This means that j increases the price by some quantity ΔP of each resource r for which there is excess demand. Potential buyers who placed bids resulting in excess demands must either issue a new bid at the new prices, or give up. The pseudo-code of the seller and buyer algorithms will be described shortly.

Figure 2 shows a simple example with $N = 4$ nodes and $R = 2$ resource types. Nodes 1 and 2 are buyers, and can interact with sellers $\{3, 4\}$, and 4 respectively. The amounts

[2]Note that the meaning of *reserve price* is different for buyers and sellers.

Algorithm 1 BUYER(i)

Require: RP_i Reserve price of buyer i
Require: $Req_{\bullet i}$ Resource bundle requested by i
1: Receive $SP_{\bullet j}, Off_{\bullet j}$ from all $j \in \text{neighbors}(i)$
2: **loop**
3:    $p := 0$ {Total cost of the bundle}
4:    $b_r := 0$, for all $r \in \mathbf{R}$
5:    **for all** $r \in \mathbf{R}$ **do**
6:       $S_r := \{j \in \text{neighbors}(i) \mid Off_{rj} \geq Req_{ri}\}$
7:       **if** $(S_r = \emptyset)$ **then**
8:          **Abort**   {Not enough items of r available}
9:       **end if**
10:      $k := \arg\min_j \{SP_{rj} \mid j \in S_r\}$
11:      $b_r := k$ {This means $X_{irk} := 1$}
12:      $p := p + SP_{rb_r} \times Req_{ri}$
13:      **if** $(p > RP_i)$ **then**
14:         **Abort**   {Reserve price RP_i exceeded}
15:      **end if**
16:   **end for**
17:   Send bid Req_{ri} to b_r, for all $r \in \mathbf{R}$
18:   Receive $SP'_{\bullet j}$ from all $j \in \text{neighbors}(i)$
19:   **if** $(SP_{rb_r} = SP'_{rb_r}$, for all $r \in \mathbf{R})$ **then**
20:      Break   {Bid successful}
21:   **end if**
22:   $SP_{\bullet j} := SP'_{\bullet j}$, for all $j \in \text{neighbors}(i)$
23: **end loop**
24: Send $Req_{rb_r} \times SP_{rb_r}$ tokens to, and use resource r from seller b_r, $r \in \mathbf{R}$

---

of resources requested or offered is shown next to each node: for example, node 1 requests 3 items of resource 1 and 5 items of resource 2.

After the sellers broadcast the offered quantities and selling prices (in this case, unitary prices are initially set to 1), buyers make their initial bids (Figure 2(a)). Specifically, user 1 requests 3 items of resource 1 to user 3, and 5 items of resource 2 to user 4. User 2 requests 2 items of resource 1 and 10 items of resource 2 to user 4. If all bids were accepted, both nodes 1 and 2 would spend $8.

According to the bids above, user 4 has excess demand on resource 2 since 11 items are requested but only 10 are available. Therefore, node 4 increases the unitary price of resource 2 to $2 (we use $\Delta P = \$1$); seller 3 has no excess demand, so he replies with the same unitary prices (Figure 2(b)).

With the new prices, the bundle requested by user 1 would cost $\$3 \times 1 + \$2 \times 5 = \$13$ (which is below the reserve price $RP_1 = \$15$), and the requested bundle of user 2 would cost $14, which is above the reserve price $RP_2 = \$10$. Therefore, user 2 gives up, while user 1 resubmit his bids (Figure 2(c)). Since there is no excess demands, the auction ends and user 1 can finally acquire his bundle (Figure 2(d)).

Algorithms 1 and 2 show the behavior of the generic buyer i and seller j, respectively. We remark that the same user may act as buyer and seller at the same time, so the algorithms above could also be executed concurrently by the same node (for clarity, we neglect synchronization issues in the pseudocode).

Algorithm 2 SELLER(j)

Require: $Off_{\bullet j}$ Resource bundle offered by j
Require: $SP_{\bullet j}$ Reserve prices of seller j
Require: ΔP price increment
1: Send $SP_{\bullet j}, Off_{\bullet j}$ to all $i \in \text{neighbors}(j)$
2: **repeat**
3:     $d_r := 0$ for all $r \in \mathbf{R}$ {Demand for resource r}
4:     **for all** bids Req_{ri} received from i **do**
5:        $d_r := d_r + Req_{ri}$
6:     **end for**
7:     $exc := false$
8:     **for all** $r \in \mathbf{R}$ **do**
9:        **if** $(d_r > Off_{rj})$ **then**   {Excess demand?}
10:          $SP_{rj} := SP_{rj} + \Delta P$
11:          $exc := true$
12:        **end if**
13:     **end for**
14:     Send $SP_{\bullet j}, Off_{\bullet j}$ to all $i \in \text{neighbors}(j)$
15: **until** $(exc = true)$
16: Receive $Req_{ri} \times SP_{rj}$ tokens and allocate Req_{ri} items of resource r to buyer i, $r \in \mathbf{R}$

Algorithm BUYER(i) requires some additional parameters, namely the reserve price RP_i of user i, and the number of items of the requested bundle $Req_{\bullet i}$. After receiving the initial price and offered resource amounts from all neighbors (line 1), the main loop starts. At each iteration, we use variable p to keep track of the cost of the bid; if p becomes larger than the reserve price RP_i, then i resigns and the procedure stops (line 14). We use the variable b_r to denote the index j of the seller from which i gets resource r. To place a bid, user i first identifies the set S_r of neighbors which are offering enough items of resource r (line 6). If S_r is empty, buyer i gives up since we require that all items in the requested bundle be available. If S_r is not empty, buyer i will bid for resource r to the seller $k \in S_r$ which advertises the minimum unitary price SP_{rk} for r (line 10). After all bids are placed, buyer i listens for the new selling prices $SP'_{\bullet j}$ from each neighbor j. If the new prices advertised by sellers to which i placed a bid match the previous prices, (line 19), the auction is successful; otherwise, a new round is performed using the updated selling prices.

Algorithm SELLER(j) describes the behavior of the generic seller j. The required input parameters are the initial reserve prices SP_{rj} (the minimum unitary price at which j is willing to sell resource r), the offered quantity Req_{rj} and the price increment ΔP. First, selling prices SP_{rj} are advertised to all neighbors (line 1). Then, the main loop starts; we use the vector d_r to keep track of the total demand of resource r, so that if excess demand for r is detected, its price SP_{rj} is incremented (line 10). The loop breaks only when there is no longer excess demand.

IV. EXPERIMENTAL EVALUATION

In this section we analyze the performance of the auction-based resource allocation algorithm using a set of numerical experiments. We consider different network sizes (with $N = 10, 20, 50$ users, respectively) and different numbers of resource types ($R = 3, 5, 7$). For each combination of N and

TABLE II.   NUMBER OF MATCHES (HIGHER IS BETTER).

		Matches		
N	R	Auction	Optimal	Auction/Optimal
10	3	14.60 ± 1.48	14.70 ± 1.45	0.99
20	3	46.10 ± 2.33	47.80 ± 2.41	0.96
50	3	162.50 ± 4.55	172.50 ± 3.14	0.94
10	5	10.40 ± 1.07	10.40 ± 1.07	1.00
20	5	33.50 ± 2.93	34.10 ± 3.12	0.98
50	5	149.80 ± 4.44	161.20 ± 2.58	0.93
10	7	6.30 ± 1.64	6.30 ± 1.64	1.00
20	7	26.40 ± 3.30	27.80 ± 3.50	0.95
50	7	126.80 ± 6.41	141.70 ± 5.38	0.89

R we perform $T = 10$ allocation steps. Each step involves the definition of requested and offered bundles (see below), and running an auction to match them. At the very beginning, each user is given a budget of 100 tokens; furthermore, before starting each step several initializations are performed, as follows.

First, we generate a random network with link density 0.3 (this means that on average, 30% of the elements of the adjacency matrix M_{ij} are nonzero). 20% of the users are randomly assigned the role of pure buyers (the offered bundles are set to zero), while the others are pure sellers (the requested bundles are set to zero). Each user is also assigned a random demand or offer vector: the number of items of each resource type that are offered/requested are drawn with uniform probability from the discrete set $\{1, 2, \ldots, 10\}$. The initial reserve prices for sellers are set to 1, and the price increment is $\Delta P = 1$. The reserve price for buyer i is set to $\epsilon_i \times \sum_{r \in \mathbf{R}} Req_{ri}$, where ϵ_i is uniformly chosen in $[1.5, 2]$; $\sum_{r \in \mathbf{R}} Req_{ri}$ is the cost of bundle $Req_{\bullet i}$ when all items have unitary cost. With the setup above we ensure that each node has sufficient liquidity to satisfy requests for resources, since each user will act most of the time as seller. In order to cope with statistical fluctuations, we executed 20 independent replications of each sequence of T steps; at the end of all replications, average values and confidence intervals at $(1 - \alpha) = 0.9$ confidence level are computed.

Number of matches.: We first analyze the total number of matches, i.e., the total number of requests which can be satisfied at the end of the sequence of T allocation steps. We compare the value obtained using the auction with the optimal value obtained by matching requests using the MIP formulation from Section II; the optimization problem has been solved using GLPK [9].

Raw results are reported in Table II. The column labeled "Auction" shows the total number of matches at the end of the T auctions, computed with the ascending auction algorithm proposed in this paper. Column labeled "Optimal" shows the maximum number of matches computed using the optimization problem. The results has been plotted in Figure 3): we can see that the number of matches produced by the auction algorithm is only slightly less than the optimum value. It is important to report that for the larger systems ($N = 50$, $R = 7$), GLPK required up to $10s$ to compute the optimal allocation (we used GLPK v4.45 on an AMD Athlon 64 X2 3800+ Dual Core Processor with 4 GB of RAM running GNU/Linux 2.6.32). The auction, implemented as a script in GNU Octave 3.2.3 [10], consistently requires less than a second on the same
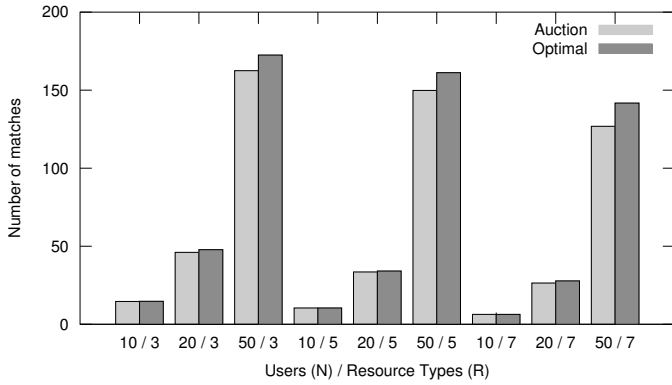
Fig. 3. Performance results: number of matches (higher is better)
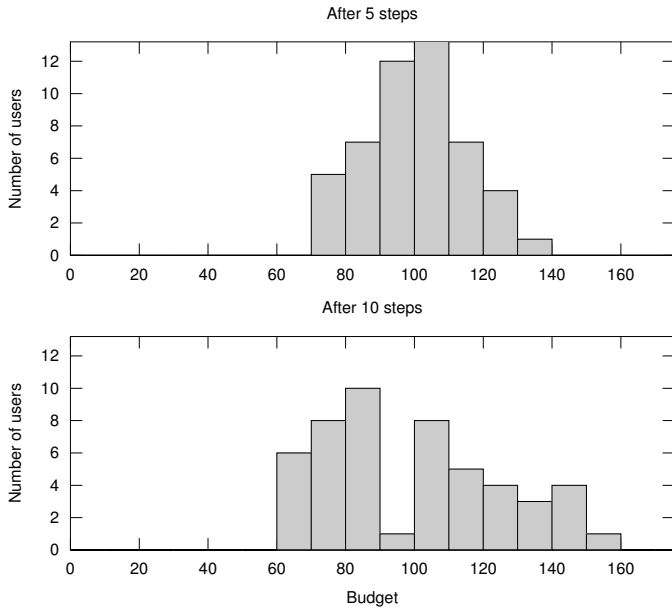
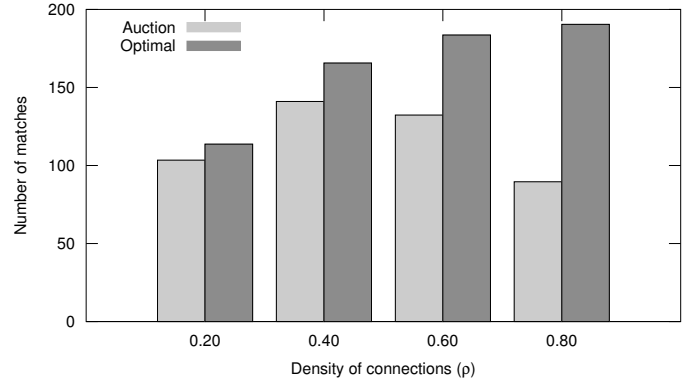| Density | Matches | | Price | Max Iter. |
	Auction	Optimal		
0.20	103.40 ± 4.96	113.70 ± 4.14	1.07	6
0.40	141.00 ± 3.98	165.60 ± 3.27	1.15	18
0.60	132.30 ± 3.21	183.60 ± 1.82	1.28	29
0.80	89.60 ± 2.78	190.40 ± 1.57	1.48	56

Fig. 5.   Number of matches for different connection densities; $N = 50$, $R = 7$

independently repeated 20 times.

The raw data is shown in Table III. As the connection density increases, each buyer can interact with more sellers, since each node has more neighbors. Therefore, we expect that the total number of matches increases because a buyer has more chances to find a seller with enough resources to match his demand.

As we can see from Figure 5, the maximum number of matches indeed increases as ρ becomes larger. However, the total number of matches obtained from the auction has a maximum at about $\rho \in [0.4, 0.6]$, and starts decreasing afterwards. The bad behavior of the auction-based allocation can be explained by the fact that for large values of ρ, many buyers are likely to share the same neighboring sellers. Therefore, many buyers are likely also to share with other buyers the seller offering the lowest price: given that all buyers will bid the best (lowest) price, this will cause contention to the "best" seller.

To substantiate this claim, we show in Figure 6 the average price for a single resource item, and the average number of rounds which are necessary for the auction to settle to the final prices. We observe that both the average resource price and the number of rounds increases as the connection density ρ becomes larger. For $\rho = 0.8$, about 60 rounds are requested to conclude the auctions, resulting in higher unitary prices on average.

Several strategies can be used to mitigate the problem below: raising the prices by a quantity proportional to the excess demand or adding randomization in the choice of sellers (such that a buyer may occasionally bid to sub-optimal sellers) are possible extensions which are currently under investigation. However, we remark that large connection densities are somewhat unlikely in the scenario we are considering, as it would require a large number of people sharing resources

Fig. 4.   Distribution of users budget, after 5 simulation steps (top) and at the end of the simulation (bottom); $N = 50$, $R = 7$

platform. We recall that the mechanism works on resource-constrained mobile devices composing DOs; hence, it is important to reduce as much as possible the overhead to perform the allocation. In practice we should expect some slowdown due to request-response messages sent through wireless links.

We show in Figure 4 the distribution of the users budget after 5 steps, and at the end of a simulation run (10 steps). Remember that each user is assigned an initial budget of 100 tokens. The budget distribution spreads over a larger interval as users trade resources, due to the nature of the experiments carried out. Each user has the same probability of being a buyer or a seller at each step as any other user.

Behavior on crowded markets.: We also investigated the impact of the connection density (i.e., number of links between users) on the total number of matches. We consider $N = 50$ users and $R = 7$ resource types, and different values for the connection density ρ of the ad-hoc network. The value of ρ is the fraction of nonzero elements of the adjacency matrix M_{ij}. We considered $\rho = 0.2, 0.4, 0.6, 0.8$; for each value, we executed $T = 10$ allocation, and each sequence was
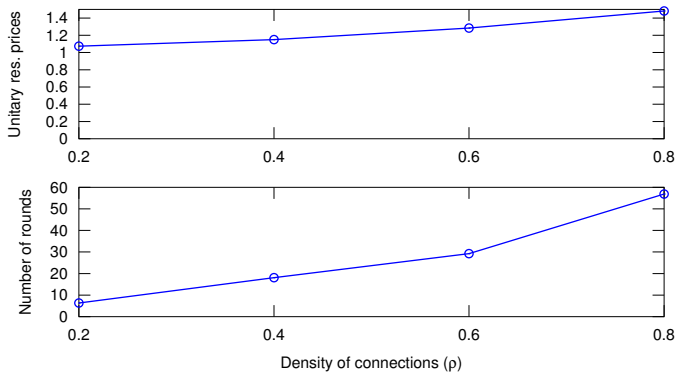
Fig. 6. Average price for any single resource item (top) and average number of rounds to complete an auction (bottom) as a function of the connection density

within a limited area. And furthermore, in such environments short-range communication technologies are preferred for their transmission power efficiency.

V. RELATED WORKS

The development of interaction mechanisms among wireless devices is a well studied area. For instance, the project "seamless computing" [11] proposed by Microsoft in 2003, and Jini, which is a part of the Java technology originally developed by Sun Microsystems, have addressed issues related to the definition and implementation of ubiquitous computing paradigms. Today, the notion of "digital ecosystem" is generally used to refer to any distributed system with properties of adaptability, self-organization, scalability and sustainability [12]. In this paper, we use this term in a slightly different acceptation, since we give more emphasis to issues concerned with mobile networks and interaction among constrained devices [1].

According to our view, there are two kinds of problems which need to be considered. The first one refers to the distributed resource utilization. In this sense, several works related to resource discovery, allocation and organization (mostly in ad-hoc fashion) are available in the literature, e.g. [13], [14], [15].

Another main problem is concerned with optimizing the communication capabilities of a DO. In general, this issue turns to allow a mobile node, having multiple wireless network interfaces, to change network points of attachment (handover) without disrupting existing connections, combined with the ability to disseminate messages in multi-hop transmissions (i.e., communication in a MANET). Examples of works on seamless host mobility are [16], [17], [18], [19]

As to the use of auctioning systems to allocate discrete computational resources, works have been already proposed, but usually employed on different use case scenarios, such as cluster and classic distributed systems [20], [21], [3], [22], [4]. Other works employ auction-based mechanisms in wireless networks; however, usually these are schemes that allow users to dynamically negotiate their agreed service levels with their service provider [23], [24], to define an optimal channel allocation or for scheduling. Hence, it is something very different from the P2P dynamic resource allocation we

are considering in this work. For example, in [25], the authors propose an auction-based, distributed coordination mechanism which can be used to determine which nodes in a wireless network act as relay nodes.

VI. CONCLUSIONS

In this paper we presented a fully distributed algorithm for resource allocation between DOs. Our algorithm is based on ascending clock auctions, and allows users to trade resources in exchange for some form of digital currency. Numerical results show that this approach represents a viable and effective strategy promoting sharing of resources, thus providing DOs with the means to form an altruistic digital ecosystem.

As concerns the general deployment of the proposed scheme in a real distributed system, there are some open problems that require further investigation. Security issues are particularly important: for instance, authentication must be enforced in order to verify the identity of users that try to utilize resources of other DOs. We will also consider more general notations to describe requests for resources, e.g., resource bundles as intervals rather than single values.

REFERENCES

[1] G. D'Angelo, S. Ferretti, V. Ghini, and F. Panzieri, "Mobile computing in digital ecosystems: Design issues and challenges," in *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011)*, IEEE, Ed., July 2011.

[2] L. M. Ausubel and P. Cramton, "Auctioning many divisible goods," *Journal of the European Economic Association*, vol. 2, no. 2-3, pp. 480–493, 2004.

[3] T. Kelly, "Generalized knapsack solvers for multi-unit combinatorial auctions: Analysis and application to computational resource allocation," in *Agent-Mediated Electronic Commerce VI, Theories for and Engineering of Distributed Mechanisms and Systems*, ser. Lecture Notes in Computer Science, vol. 3435. Springer, 2005, pp. 73–86.

[4] M. Stokely, J. Winget, E. Keyes, C. Grimes, and B. Yolken, "Using a Market Economy to Provision Compute Resources Across Planet-wide Clusters," in *Parallel and Distributed Processing Symposium*, 2009.

[5] S. Nair, E. Zentveld, B. Crispo, and A. Tanenbaum, "Floodgate: A micropayment incentivized p2p content delivery network," in *Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on*, aug. 2008, pp. 1–7.

[6] M. Sirivianos, J. H. Park, X. Yang, and S. Jarecki, "Dandelion: cooperative content distribution with robust incentives," in *2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2007, pp. 12:1–12:14.

[7] K. Wei, A. J. Smith, Y.-F. R. Chen, and B. Vo, "Whopay: A scalable and anonymous payment system for peer-to-peer environments," in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, ser. ICDCS '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 13–.

[8] B. Yang and H. Garcia-Molina, "Ppay: micropayments for peer-to-peer systems," in *Proceedings of the 10th ACM conference on Computer and communications security*, ser. CCS '03. New York, NY, USA: ACM, 2003, pp. 300–310.

[9] A. Makhorin, *GNU Linear Programming Kit–Reference Manual for GLPK version 4.45*, Dec. 2010, available at http://www.gnu.org/software/glpk/glpk.html.

[10] J. W. Eaton, *GNU Octave Manual*. Network Theory Limited, 2002.

[11] M. Chalmers and I. Maccoll, "Seamful and seamless design in ubiquitous computing," in *Proceedings of Workshop At the Crossroads: The Interaction of HCI and Systems Issues in UbiComp. 2003*, 2003.

[12] G. Briscoe, "Complex adaptive digital ecosystems," in *Proc. International Conference on Management of Emergent Digital EcoSystems*, ser. MEDES'10. New York, NY, USA: ACM, 2010, pp. 39–46.

[13] B. F. Cooper, "An optimal overlay topology for routing peer-to-peer searches," in *Middleware '05: Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware*. New York, NY, USA: Springer-Verlag New York, Inc., 2005, pp. 82–101.

[14] G. H. L. Fletcher and H. A. Sheth, "Unstructured peer-to-peer networks: Topological properties and search performance," in *3rd International Conference on Autonomous Agents and MUlti-Agent Systems*. Springer, 2004, pp. 14–27.

[15] I. Gruber, R. Schollmeier, and W. Kellerer, "Peer-to-peer communication in mobile ad hoc networks," *Ad Hoc & Sensor Wireless Networks*, vol. 2, no. 3, 2006.

[16] M. Bonola, S. Salsano, and A. Polidoro, "Upmt: universal per-application mobility management using tunnels," in *Proceedings of the 28th IEEE conference on Global telecommunications*, ser. GLOBE-COM'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 2811–2818.

[17] S. Ferretti and V. Ghini, "A web 2.0, location-based architecture for a seamless discovery of points of interests," in *Proceedings of the 2009 Fifth Advanced International Conference on Telecommunications*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 226–231.

[18] A. Mitseva, N. Prasad, P. Todorova, R. Aguero, A. Garcia Armada, C. Panayioto, A. Timm-Giel, and L. Maccari, "CRUISE research activities towards Ubiquitous Intelligent Sensing Environments," *IEEE Wireless Communications Magazine*, vol. 15, no. 4, pp. 52–60, August 2008.

[19] K. Tsukamoto, T. Yamaguchi, S. Kashihara, and Y. Oie, "Implementation design of handover management scheme for efficient multimodal communication," in *Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 684–689.

[20] D. Grosu and A. Das, "Auction-based resource allocation protocols in grids," in *In Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing and Systems*. ACTA Press, 2004, pp. 20–27.

[21] D. F. Ferguson, C. Nikolaou, J. Sairamesh, and Y. Yemini, *Economic models for allocating resources in computer systems*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1996, pp. 156–183.

[22] S. Reddy and A. Gupta, "Auction based resource allocation in grids," in *Distributed Computing and Networking*, ser. Lecture Notes in Computer Science, S. Chaudhuri, S. Das, H. Paul, and S. Tirthapura, Eds. Springer, 2006, vol. 4308, pp. 145–156.

[23] M. Dramitinos, G. D. Stamoulis, and C. Courcoubetis, "Auction-based resource allocation in umts high speed downlink packet access (hsdpa)," in *in Proceedings of the First EuroNGI Conference on Traffic Engineering (NGI 2005)*, Apr. 2005, pp. 434–441.

[24] T. Taleb and A. Nafaa, "A fair and dynamic auction-based resource allocation scheme for wireless mobile networks," in *Proceedings of IEEE International Conference on Communications, ICC 2008*. IEEE, 2008, pp. 306–310.

[25] J. Huang, Z. Han, M. Chiang, and H. V. Poor, "Auction-based distributed resource allocation for cooperation transmission in wireless networks," in *Proceedings of the Global Communications Conference, GLOBE-COM '07*. IEEE, 2007, pp. 4807–4812.