

A Framework for Autonomous Networked Auctions

Antonio Di Ferdinando
Intelligent Systems &
Networks
Imperial College London
London SW7 2BT, UK
a.di-
ferdinando@imperial.ac.uk

Ricardo Lent
Intelligent Systems &
Networks
Imperial College London
London SW7 2BT, UK
r.lent@imperial.ac.uk

Erol Gelenbe
Intelligent Systems &
Networks
Imperial College London
London SW7 2BT, UK
e.gelenbe@imperial.ac.uk

ABSTRACT

Business companies are showing a growing interest towards the use of the Internet as part of their business. In particular, networked auctions exploit characteristics that are particularly attractive, and are therefore gaining interest as mean for business. This is especially true when the management efforts, to handle such auctions, are reduced by the use of automatic trading agents. The *automatic* nature of agents, however, does not allow them to account for the non-stationariness of the market, with the result of inevitably limiting optimization of auction economic factors and, finally, the overall utility.

We outline a platform where trading agents behave in an *autonomous* fashion. Agents employ biologically-inspired techniques, to adapt auctioning strategies to present environmental conditions, through exploitation of *self-** properties. We put particular emphasis on *self-configuration* and *self-adaptation*, presenting results of preliminary experimentations, and showing through simulations how autonomous behaviour is expected to impact current results.

1. INTRODUCTION

In recent years, business companies have shown a growing interest in the use of the Internet as a mean for doing commerce, in what it is usually called *e-commerce*. As an example, consider networked auctions. Their nature allows to hammer geographical market boundaries, with a consequent widening of possibly interested audience. This enhances competition, and has important repercussions on the final utility. Furthermore, it is becoming common to rely on auctions as a mean for doing real-time business, thanks to the reduction in management efforts that the use of automatic trading *agents*[24] bring. As a consequence of these trends, the ideal duration of auctions can be reduced to minutes, rather than hours or days as for traditional electronic auctions, thus resulting suitable for serial use[19].

Automatic trading agents typically act in the basis of decision policies which are statically defined. Thus, they do not usually account for non-stationariness of the market, such

as for instance the stock market[22]. In other words, they tend not to account for market and environmental parameters when taking critical decisions concerning auctions. This has inevitable repercussions on auction factors, especially of financial nature.

Accounting for the non-stationary behavior of a market involves agents to be able to dynamically adapt their behavior based on the characteristics of the market in a specific moment. This, in turn, requires agents to be *aware* of the market in that specific moment and behave so as to optimize auction factors adapting to such circumstances.

Context-awareness leads to exploitation of a set of properties, collectively defined as *self-** properties[14], typical of autonomous computing[5]. Application of such technology to the networked auction scenario is thus expected to improve trading agents with means to adapt their strategy based on the current market moment.

In this new scenario, *autonomous* trading agents try to exploit the most appropriate behaviour in the present situation without human intervention. Decisions are taken according to factors that change dynamically based on self-awareness and adaptation to present environmental market. This makes agents sensitive to dynamic changes in the environment which, in turn, would allow them to pursue the goal they are given in a more accurate way.

A novel computing platform with such characteristics is the main objective of the CASCADAS[3] project (EU FP6-027807). The material here presented is based on an ongoing research partially funded by the above project, that authors here are glad to acknowledge.

The overall goal of CASCADAS is identifying, developing, and evaluating architectures and solutions based on a general-purpose component model for autonomous communication services. The project is driven by the ambition of identifying a fundamental, uniform abstraction for situated and autonomous communication entities, at all levels of granularity. This abstraction is called an *Autonomous Communication Element* (ACE), and represents the cornerstone of the component model, in which the four driving scientific project principles (situation awareness, semantic self-organisation, self-similarity, autonomous component-ware) will properly converge. Concepts here presented find a strong convergence in the objectives of the CASCADAS project, and consequently objects and entities here presented can easily be integrated into an ACE.

This paper is structured as follows: in section 2 we describe fundamental concepts for understanding the platform. Section 3 describes the fundamental autonomous units present

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AUTONOMICS 2007, 28-30 October 2007, Rome, Italy

Copyright © 2007 ICST 978-963-9799-09-7

DOI 10.4108/ICST.AUTONOMICS2007.2304

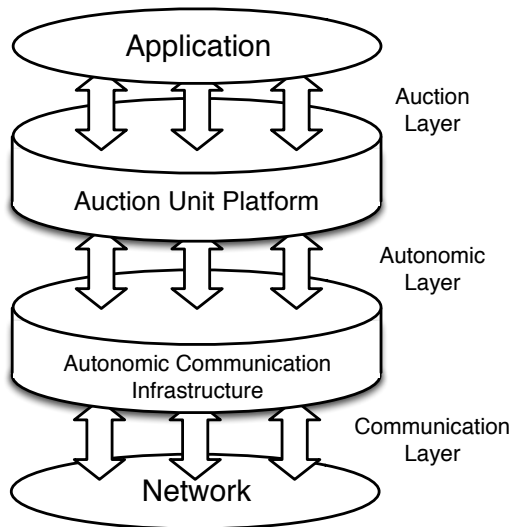


Figure 1: Layered platform structure.

in the platform. Autonomic units are supported by an autonomic communication infrastructure, which is presented in section 4. Some preliminary measurements are then presented in section 5, while section 6 outlines how introduction of autonomic features is expected to impact the platform. Finally in section 7 we draw some conclusions.

2. PLATFORM DESCRIPTION

We consider a platform populated by users and services[6], which play a symmetric role. Hence, users of some services can become services for other users. The platform allows users to dynamically request services by specifying a set of parameters such as desired *Quality of Service* (QoS), price for the service and duration. Such a network can monitor service provision and inform users of the service level actually being delivered.

User and services are supported at various levels by a layered architecture, shown in figure 1. At the bottom of the platform, an *Autonomic Communication Infrastructure* (ACI) interacts with the standard network infrastructure to provide sophisticated communication facilities that are sensible to specified QoS attributes. Besides, the ACI also provides information about behavior of the network to the above layer. The ACI is capable of autonomously optimizing the coordination among several parties involved in a communication process according to a set of configuration parameters that can be specified in advance, yet dynamically change during the service provision time through the use of a Self-Aware Network[8] (SAN). This latter is capable of adapting communications to present network conditions, according to specified QoS attributes. In our experiments, we use the *Cognitive Packet Network*[7, 10] (CPN).

On top of the ACI, the *Auction Unit Platform* (AUP) defines and realizes the environment where users and services interact. These latter are realized through agents, that we named *Auction Units* (AUs), and the platform is thus a multi-agent platform. The topmost layer in this structure is an application that allows the user to visually interact with the platform underneath. In this way, the user can visually observe how the platform evolves or being directly involved

in the auctioning process, overriding the autonomic logic underneath. It is worth noting that the user can be a human user as well as another, application-level, agent.

AUs in the platform dynamically change their behavior based on the context they are currently operating in. As for auctions, they can behave according to the following roles:

- *Bidder*. Bidders are AUs that notify their will to buy an item of interest, and for this reason they are users in the platform. They engage in a competition to the extent of acquiring the rights to buy items under auction, and might employ strategies aimed to reduce the financial efforts to achieve so.
- *Auctioneer*. Auctioneers are AUs that own one or more items and are willing to sell them through an auction. For this reason, they are services in the platform. Auctioneers typically start an auction by notifying own intention to sell an item according to a set of rules, which collectively form the *Auction Model*. The choice of the auction model through which auctioning an item is made by the auctioneer. In our work, we chose *English Auction*[15] as the referring auction model, due to its wide use. According to rules of the model, thus, the auctioneer acts as a *seller*.
It is worth noting that, unlike traditional face-to-face auctions where the term "auctioneer" refers to an intermediate entity, such as an auction house, in charge of conducting the auction on behalf of the owner of the item, in our platform auctioneers are owners of items that handle auctions directly.
- *Auction Centre* (AC). ACs provide meeting points for auctioneers and bidders. Through ACs, in fact, sellers may advertise their "wares" and indicate their asking price, and buyers may advertise their will to pay a certain amount for particular items. ACs have an influence that can be related (or not) to a geographic area, or to a particular type of products or services. Through the use of self-* properties, ACs might decide to form a coalition to the extent of reaching a wider

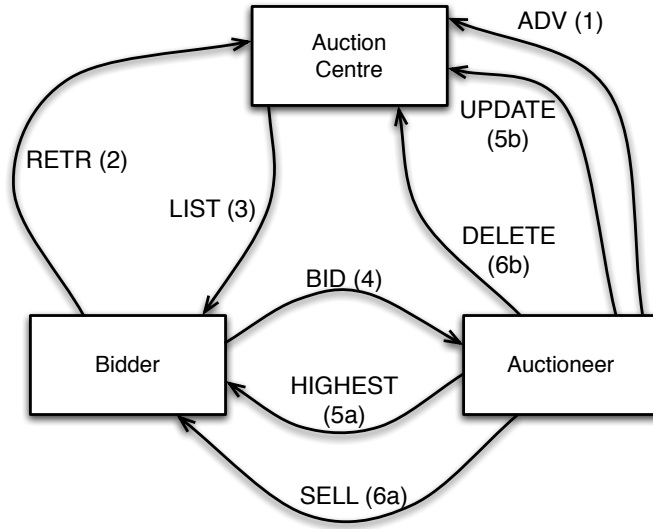


Figure 2: English Auction transaction model

spectrum of users and/or services, or to specialize in certain categories if the relevant interested audience can be localized in the platform. Obviously, given the delicate position they occupy, ACs will need to provide a set of trust management, fairness and security guarantees. Besides, they will enable some level of control over the auctions, for example by regulating access through verification of users' credentials.

ACs contain an *Auction Web Page* (AWP) that bidders can consult in order to find out who is selling goods of interest. An AWP is essentially an overall representation of a set (or subset) of goods under auction. Advertisement of an item or service on one or more ACs by sellers equals to a notification of seller's intention to auction the good under the set of rules there specified by the seller itself. In the context of the platform, bidders represent users, and sellers represent services. Services can become users, for instance, by allowing agents acting as sellers in certain auctions to act as bidders for goods in other auctions and vice versa.

3. AUCTION UNITS

The AUP is populated with Auction Units (AUs), which represent physical users in the platform. AUs are agents capable of behaving as a bidder, auctioneer and/or AC based on context-awareness and environmental self-knowledge. Communication in the AUP is supported by the ACI, which allows communication to be dynamically adapted to specific QoS goals.

AUs exploit the concept of mobility by being *self-mobile*: AUs are associated with a *Physical Location* and a *Virtual Location*[20] in the network. Consider a physical user B . The Physical Location $PL(B)$ defines the location of the physical user B . This can be mobile if the physical user connects to the platform through a mobile device, and therefore the use of a middleware for construction of mobile applications such as [4] can be envisaged. The Virtual Location $VL(B)$, on the other hand, defines the position of the AU that represents B in the overlay network. The physical location moves based

on physical movements, while the virtual location moves as a function of a set of parameters such as context awareness, security, QoS, etc. We term movements of the VL as *migrations*.

3.1 Migrations

A migration is the process of delegating auction responsibilities to another node in the platform, and can be regarded as a form of *self-organization*. The original, i.e. the *delegating*, node performs a migration by creating a virtual location expressly for a specific auction. This, in turn, is moved to another node in the network, named the *delegated* node, to the extent of conducting the auction from this new location. Migrations allow AUs to have multiple virtual locations VL_i (AU). These are instantiated and migrated to the delegated node they have been instantiated for, to keep a channel between VLs at delegating and delegated nodes. The exact rationale for migrating depends on the specific auction context. However, mobility of AUs in the network is directed as a function of its QoS goals and based on environmental self-knowledge. Thus, exploiting self-mobility allows the AU to act opportunistically in the context of one or more auctions. For example, AUs may use mobility to position themselves to reach more easily markets with a wider audience (as a seller and/or an AC) or a market with a more consistent presence of items of interest (as a bidder).

Migrations involve transferring a certain amount of auction-related data from the delegating node. This data will then be used and updated by the delegated node in the process of conducting the auction on behalf of the delegating node. Once the auction terminates, all data concerning that particular auction, i.e. the original data updated and any other data originated in the process of conducting the auction, will need to be transferred back to the delegating node.

Migrations might also involve *aggregation*, intended as the dynamic formation of a new, composite, AU by the joining of the VL of delegating and delegated nodes. Aggregations allow an AU to be present in both the old and the new position in the network, and are useful when the delegating AU needs to migrate into a new location while maintaining

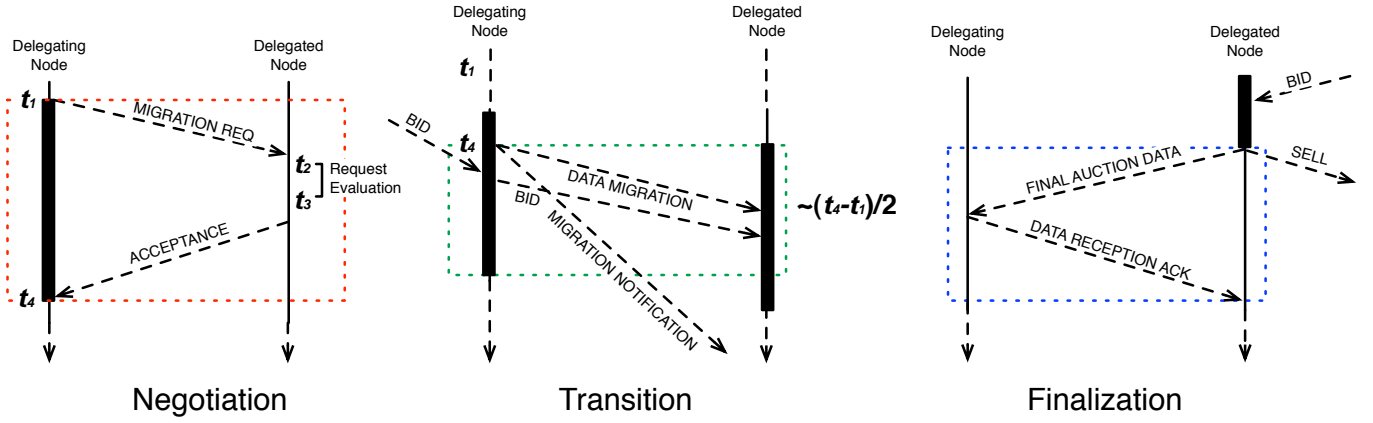


Figure 3: The 3-phases migration protocol.

its presence in the old location. Aggregations are decided based on calculation of the tradeoff between gains and efforts. In addition, the decision on whether to aggregate or not is taken autonomically, so that this property can be named *self-aggregation*.

Consider a scenario in which several nodes located on physically different networks are interconnected in the auction platform. Bidders find goods of interest by contacting one or more ACs. Through them, they learn the virtual location of the auctioneers and start bidding. In order to gain advantage over other bidders, the AU of a bidder might decide to migrate to another node. The reason for migrating lies in the advantage gained in terms of QoS, perceived at the new location, towards the auctioneer. For instance, considering packet delay as a QoS attribute, the bidder might want to migrate because of a large packet delay suffered towards one or more auctioneers of interest. The autonomic communication infrastructure determines the ideal node to migrate to, i.e. the one maximizing the tradeoff between the QoS perceived towards those auctioneers of interest and communication efforts needed for migration. Then, the AU evaluates *pros* and *cons* of the migration, for instance by calculating a tradeoff between the expected gain and the financial effort needed to migrate, whereas the delegated node is willing to charge the delegating node for taking the auction responsibilities. If the evaluation is positive, the bidder opportunistically migrates to the delegated node. From this, the delay towards the auctioneers of interest is reduced and this might give the bidder a considerable advantage over other bidders. Similarly, an auctioneer might decide to migrate opportunistically to another node. Based on certain specified parameters (the perspective to expand its market, proximity to a larger number of ACs and/or bidders, etc.), the autonomic platform determines the ideal node to migrate to. In this case, the auctioneer would generate a VL with which to aggregate to the delegated node. By doing so, the auctioneer would in fact be able to migrate to the node while maintaining its presence in the original position. This allows the auctioneer to start interacting with new AUs while preserving interactions with previous ones.

Agents in the platform communicate through *Auction Messages*, which allow parties involved in the auction process to exchange information relevant to the auction. The auction starts with the seller advertising the auction to an AC. Along

with item-related information (such as name, initial price, etc.), the seller specifies the auction model under which the item will be auctioned. The AC then updates its AWP with data about the new auction. Some time after, bidders ask the AC for retrieval of a specific item, and obtain back a list of auctions of interest. Once decided the right item(s) to bid for, a bidder places a bid, based on modes and terms specified in the auction model, by contacting the seller via a point-to-point communication. If the bid results to be the highest, the bidder is as such acknowledged by the seller, who also notifies other competitors, in particular the former highest bidder, and the AC about the new highest price. When the seller decides that the auction is over, notifies all other competitors, as well as the winner, about the termination of the auction.

Virtual locations require the ACI to determine the ideal location for migration in the network. In addition, autonomic elements will be necessary to facilitate the exchange of Auction Messages. In fact, the process of finding offers and conducting the rational process of auctions involves complex communications (including migrations) with stringent and highly complex requirements. As an example, consider location in a time-constrained auction. Bidders capable of reaching the auctioneer quicker (e.g. closer, in terms of end-to-end delay) will have better chances to snipe the auction than bidders a few milliseconds further away from the same auctioneer. This creates an undesirable position to certain participants, for instance users of wireless devices which may have higher loss probabilities. The complexity scales up further if we consider that users will often interact with more than one AC to try to increase their profits.

3.2 Migration Protocol

Migrations take place through a three-phase protocol, depicted in figure 3 and described below.

- Phase 1 (*Negotiation*). The delegating node negotiates terms and conditions of a possible migration to the delegated node, already individuated as the node that maximizes the tradeoff between expected gain and efforts needed for migrating. To this end, the delegating node contacts the delegated node and requests a migration to handle a specific auction. The delegated

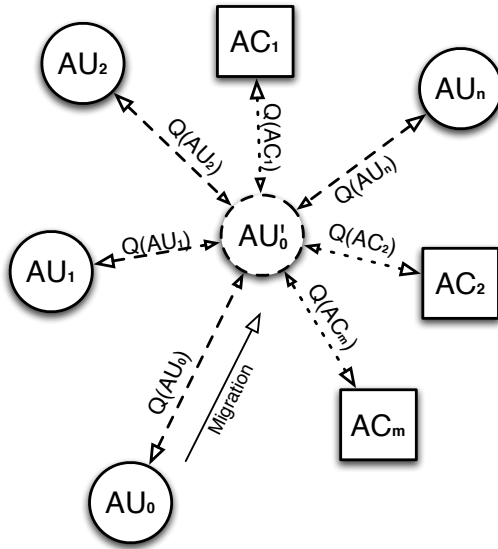


Figure 4: Positioning of virtual AU in the ACI.

node will evaluate whether is convenient or not to conduct the auction on behalf of the delegating node. As a side remark, this might involve considering to abandon an auction the delegated node is involved in. For instance consider the case where the delegated node is competing itself in the same auction; accepting the migration would involve the delegated node to abandon the auction it is actually involved in (as long as it is not the actual highest bidder) and thus it decide to charge the delegating node as a sort of compensation for abandoning the auction.

The negotiation is said to fail if both parts do not reach an agreement. In this case, the delegating node, based on the current situation of the auction and the market, evaluates *pros* and *cons* of migrating to another node, which might be the second optimal node, if the migration is for the same auction, or another node if the migration concerns another auction. On the other hand, if a satisfactory agreement is reached by both parts, the negotiation is said to be successful. In this case, the agreement has legal document (for instance through a *Service Level Agreement*[13](SLA) and, as such, it binds both parts to respect the terms and conditions found on the agreement itself. In particular, the contract will specify a clause that will enforce the delegated node to transfer back all auction data once this terminates. In both cases, however, the delegated node notifies the delegating node through a notification message and, in case the notification is positive, the migration process progresses to phase 2, described below.

- Phase 2 (*Transition*). If negotiation is successful, the delegating node broadcasts all competitors a notification and starts transferring auction data to the delegated node. From that moment on, the delegated node is responsible for the auction duties.

This transition phase is critical to the migration process, and thus foresees both delegating and delegated nodes to be responsible for the auction. This is espe-

cially important when the delegating node is a seller (or an AC). It might in fact happen that, while in the process of transferring data to the delegated node, the delegating node receives a bid (an item retrieval request) from a new bidder. This occurrence is handled by the delegating node simply forwarding the message to the delegated node, as this latter is the new referee for the auction. When the data has been transferred completely, the transition phase comes to a completion. From this moment on, the delegated node will conduct the auction on behalf of the delegating node.

- Phase 3 (*Finalization*). This phase starts when the auction terminates. The delegated node, who has conducted the auction on behalf of the delegating node, is due to transfer updated auction data back to the delegating node. When the transfer operation terminates, correct reception of data is then acknowledged by the delegating node. Then, the delegated node is requested to permanently delete its local copy of auction data and, once done, the contract is exhausted and the two parts are no longer in business.

4. AUTONOMIC COMMUNICATIONS

The auction platform supports autonomic communication through the use of a self-aware network[8]. In particular, the *Cognitive Packet Network* (CPN)[7] is used in the context of our work. This technology allows to achieve an *a priori* specified QoS goal through communications which are characterized by flows of packets.

Internally, the CPN uses an intelligent routing engine, based on a *Random Neural Network*[9] (RNN), that in turn uses reinforcement learning algorithms to provide routing paths based on some specified QoS criteria. Criteria may be simple or composite, including many simple goals (e.g. low delay and low packet loss). Packets flow over *self-adaptive* destination paths. Self-adaptation is achieved by introducing *Smart Packets* (SPs) in the network, which carry QoS goals information and incrementally build a path towards a

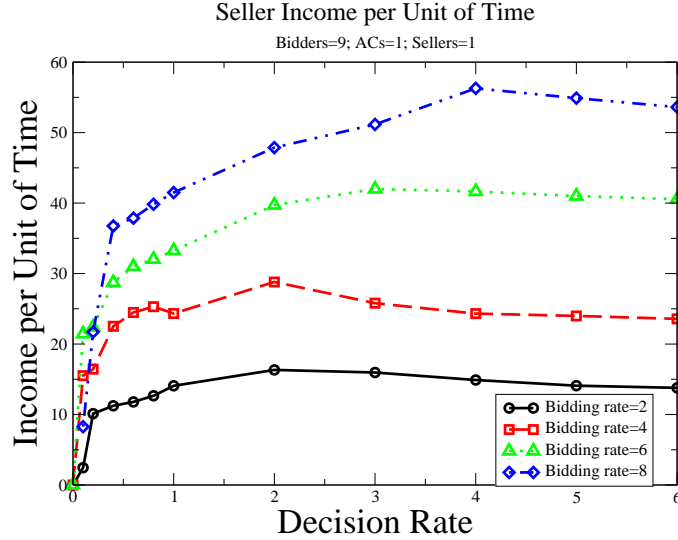


Figure 5: Seller income per unit of time as a function of the decision rate.

destination under respect of the QoS goals. Smart packets are sent as a ratio of regular communication packets, so that changes happen autonomically. To be efficient, smart packets rely on network monitoring information carried on by all packets in the network. Information collected by packets is stored in a virtual board distributed among all participating nodes in a path.

Paths obtained in this way result self-configured in a way that they reach the specified destination in respect of QoS constraints. In addition, the engine autonomously reconfigures paths when unforeseen events, such as performance drops or nodes internal to the path crash, make the selected path no longer available or suitable for the specified QoS constraints. In other words, CPN paths are self-organizing. The use of smart packets to build self-adaptive paths also enhances self-aggregation in the network, intended as the capability to autonomically aggregate to other nodes, with the purpose of providing a comparative advantage with respect of one or more auctions. In addition, as more aggregations take place, the virtual board will constitute a larger knowledge network to be used by nodes in the platform as a base for understanding the network context they operate in and learning patterns for more stable and robust future aggregations.

Ideal nodes for possible migrations are individuated, at this level, by evaluating a function that considers the unified QoS level. Consider a networked auction scenario consisting of a number of AUs and ACs, connected through the ACI (Figure 4). Suppose that AC_0 is aware of m auction centers AC_1, \dots, AC_m . Suppose also that AC_0 wishes to participate as a bidder of an auction involving AU_1, AU_2, \dots, AU_n . To handle the request, AU_0 produces AU'_0 , a virtual auction unit responsible for this specific operation. In fact, AU'_0 is normally expected to interact with the virtual counterparts of AC_1, \dots, AC_m and AU_1, AU_2, \dots, AU_n as the same opportunistic process can be applied to them. However, these virtual operations, external to AU'_0 , need to be hidden by the

autonomic infrastructure to AU'_0 . To simplify the description, we simply refer to them using their physical notation. The autonomic communication infrastructure provides AU'_0 the means to establish communication paths towards its counterparts, involved in the auction, to allow the exchange of auction messages. Specifically, assume that the current location of AU'_0 is node i . The network provides a set of paths

$$P_i = \{P_{AU_0}, \dots, P_{AU_n}, P_{AC_1}, \dots, P_{AC_m}\}, \quad (1)$$

one for each destination, and a set of QoS metrics

$$Q = \{Q(AU_0), \dots, Q(AU_n), Q(AC_1), \dots, Q(AC_m)\}, \quad (2)$$

corresponding to current observations of the relevant performance of each path. Note that $Q(AU_0)$ denotes the QoS observation of the communication channel between i and the original auction unit where the bidder resides. Note also that the elements of Q provide an indication of the communication performance and other parameters between AU'_0 and another auction element (AU or AC) in either (or both) direction. In general, each of the elements of Q are combined quantities of basic QoS metrics of interest and other parameters (e.g. monetary cost).

A unified evaluation of the current location of AU'_0 with respect to the auction is a function:

$$G_i = f(Q(AU_0), \dots, Q(AU_n), Q(AC_1), \dots, Q(AC_m)) \quad (3)$$

where f is defined to match the type of auction being used, the preferences of the user and specific goals for the auction. A migration to a node j might occur only if $G_j > G_i$ and an ideal migration would occur when $G_j = \max(G_k) > G_i$ for all known nodes k .

Knowledge of the estimated performance G_j from different locations j (other than the current location of AU'_0) may be inferred from P and Q by composition techniques.

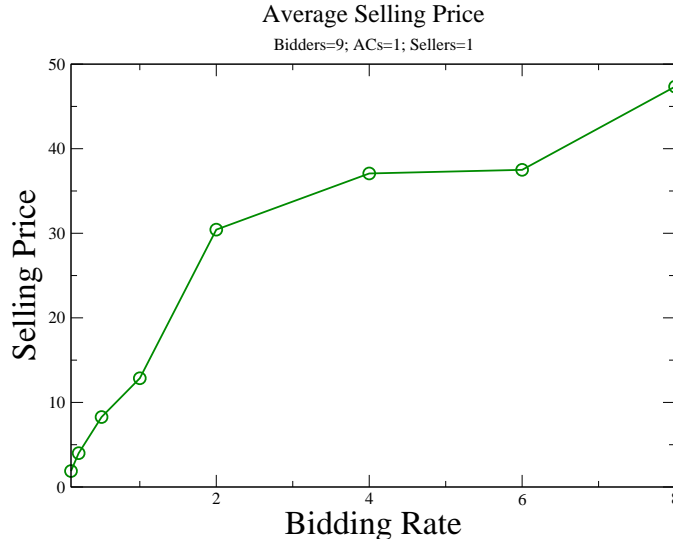


Figure 6: Selling price as a function of the bidding rate.

5. PRELIMINARY MEASUREMENTS

The auction platform just described is under development. However, a first implementation does exist and on this we conducted some preliminary experiments. The process of auctioning items follows the model for the analysis of single networked auctions[11], where a seller continuously auctions a single item through an English Auction model. Bids are assumed to increase the price offered of one unit of currency. The seller receives one bid at a time and, upon reception, takes some time to evaluate goodness of the bid. The time spent in this thinking activity is based to a *decision time*, specified *a priori* and generated according to an exponential distribution of rate d .

While evaluating the goodness of a bid, the seller might receive another bid offering a higher price. In this case, the seller discards the previous offer and repeats the same process, i.e. spends some time considering the offer for final acceptance, for the new offer. When the evaluation time finishes, the seller terminates the auction by selling to the owner of the bid with the highest price.

At bidding side, bidders contact the AC for retrieval of auctions for items of interest. The timing for the retrieval is established according to a *bidding time*, and the interval time is generated exponentially with a *bidding rate* b . The model for single networked auctions foresees the presence of a single seller auctioning one item at a time, and therefore the list of items received will always contain one auction.

Once retrieved the auction of interest, the bidder decides whether to bid or not. Provided that the bidder is not already leading the auction, in which case no action can be taken, this decision is taken by watching at the price that should be offered for item. Each bidder contains, in fact, an internal representation of the maximum value attributed to the item, V . This value is randomly generated on each bidder, according to a uniform distribution, and essentially represents the maximum price the bidder is willing to offer for that particular item. If the price to be paid in the next

possible bid exceeds the value V of the item, then the bidder retains that the item is not actually worth the price to be paid, and decides to terminate its involvement in that particular auction. On the contrary, if the price to be offered with the next possible bid does not exceed the value V for the local maximum value, the bidder places a bid. In this case, the bidder contacts the seller directly, without involvement of the AC through point-to-point communication, and places the bid instantaneously.

In the scenario described above, it is of interest to estimate the average seller income per unit of time. In fact, given that the maximum achievable price is bounded by the maximum among bidders internal values, trying to achieve such maximum value will result in the seller spending too much time in evaluating bids and, in turn, a decrease in its average income per unit of time. On the other hand, forcing decisions to be too fast in time will result in the seller not exploiting bidders' competition, thus deciding on low offers and overall not reaching the optimal income per unit of time. The timing for decisions is directly influenced by the decision rate d mentioned above. Therefore, setting d to be small will result in the average time spent in deciding to be high, whereas setting d with higher values will increase the speed of decision time.

Experiments were conducted in sets of 30 minutes-long sessions with fixed decision and bidding rates. In each session, we recorded the start time and termination time of each auction, and calculated the seller income, for the units of time corresponding to duration of the auction, by dividing the final selling price by the auction duration. At the end of the session, the average seller income per unit of time was calculated by dividing the seller income per unit of time calculated in each auction by the duration of the session.

The result of sessions with different data set are shown in figure 5. There, the the average seller income per unit or time is drawn as a function of the decision rate. Symbols in each line refer to the decision rates used as sample points for

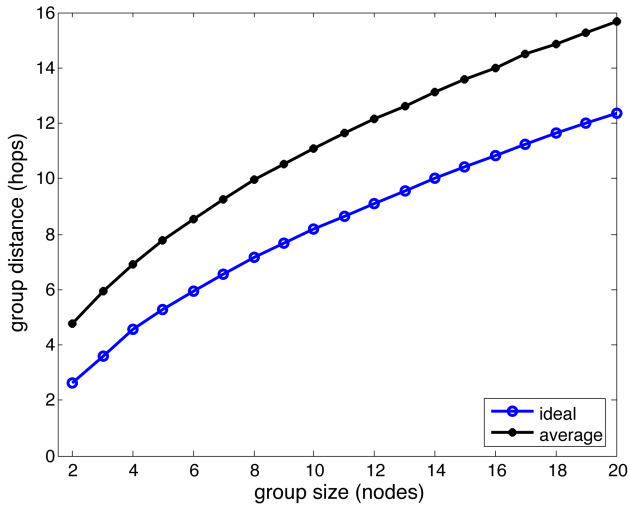


Figure 7: Simulation of average and ideal group distances in a 5×4 grid.

the data fathering operation, and the four lines represent income obtained with different bidding rates. As expected, the income is seen to start at very low figures when the decision rate d is set to small values as to indicate that the seller has spent too much time in evaluating bids. The income is seen to grow when bidders place bids at a faster rate, but still the income is low. As the decision rate is set to higher values, the income per unit of time increases. However, for very high values the curve shows a tendency to decrease. This is due, as mentioned earlier, to decisions becoming too fast, resulting in terminating the auction before bidders reaching values close to their corresponding V and effectively limiting bidders' competition.

The graph clearly shows that the seller income per unit of time reaches a peak and, therefore, that the decision rate can be balanced so as to achieve an optimum value. It is also possible to note how such optimal value is shifted towards higher decision rates, when the bidding rate increases. In addition to what said, the graph shows how the selling price increases as the bidding rate increases.

This concept is better studied in figure 6, that shows the selling price as a function of the bidding rate. Data in this graph is calculated by simply averaging the selling price of each auction in a session, and running sessions with various bidding rate. The decision time is kept fixed at $d = 0.5$ in all sessions.

Unsurprisingly, the graph shows that the higher the bidding rate, the higher the selling price. However, it can be noted that the difference in the average price is bigger when increasing the bidding rate from $b = 1$ to $b = 2$, and this seems to suggest proximity to the optimal seller income per unit of time.

6. EXPECTED IMPACT OF SELF-MOBILITY

The graphs in figures 5 and 6 clearly show that increasing the rate at which bidders place their bids increases the possibility to achieve higher final selling prices, also increasing the seller income per unit of time. Albeit in our experiments such rate is generated based on an exponentially distributed

rate, in real scenarios the rate at which bidders place their bids is influenced by many factors. Some of them, such as the worthiness of the price for that specific item and the impact on the current budget, depend on internal strategies, which are typically employed to the extent of trading an increase in own chances to win an auction against the financial efforts needed. However, some other factors are influenced by interactions that take place externally to the actual bidder. A typical example is communication. In auctions whose average duration is in the order of minutes, such as the ones under consideration, the capability to place timely bids assumes importance at both selling and bidding sides. While, in fact, allowing to increase the bidding rate and consequently the selling price for an item at selling side, timely bids allow to increase own chances to win by hammering competition.

As an example, consider two bidders bid_1 and bid_2 which are the last remaining ones to compete for an item under auction whose current price is x . Suppose they are willing to pay the same amount of money for the same item, i.e. $V_{bid_1} = V_{bid_2}$, and suppose they place a bid, y_{bid_1} and y_{bid_2} (with $y_{bid_1} = y_{bid_2} = x + 1$) at the same time. Finally, suppose the bid they are about to place is the maximum price they are willing to pay for that item, i.e. $y_{bid_1} = y_{bid_2} = x + 1 = V_{bid_1} = V_{bid_2}$. Then, in this situation the winning bid will be the "fastest" one, i.e. the one reaching the seller more quickly.

Communication is therefore a factor that can contribute to increasing the bidding rate. In our platform, communication optimization can be achieved through migrations. In fact, as mentioned earlier, bidders can decide to migrate to new locations based on the QoS perceived towards the seller of a specific auction (or sellers of a set of auctions). In order to show how migrations contribute to such extent, we simulated such a scenario.

We expect that our experimental activities will show the effectiveness of migrations in handling time-sensitive communications for auctions. The current test-bed consist of 20 nodes arranged in a grid of 5×4 and connected with 100 Mbps links. Experiments were conducted to investigate the benefit of migrations in reducing group distance to destinations for auction traffic. For the sake of simplicity, we conducted the experiments using hop count as the basic metric in all calculations of group distance. However, more interesting metrics, such as latency, can be quickly adopted by the system as the underlying CPN already has support for them.

The experiments modeled auction activity with synthetic traffic and they consisted in randomly selecting a source of traffic and a number k of destinations. In addition, simulated auction parties are assumed to migrate according to a *migration rate*. Such rate was *a priori* fixed for each experiment run.

Figure 7 compares the group distance configurations obtained from the initial traffic configurations, i.e. random locations of source and destinations with ideal configurations. The comparison is then expressed as a function of the group size. Ideal configurations were obtained assuming perfect knowledge of the network at each point of the grid so as to calculate an optimal migration of traffic source. The results show that significant gains could be obtained by an adequate mechanism to handle virtual mobility.

A set of test-bed experiments were carried out to illustrate

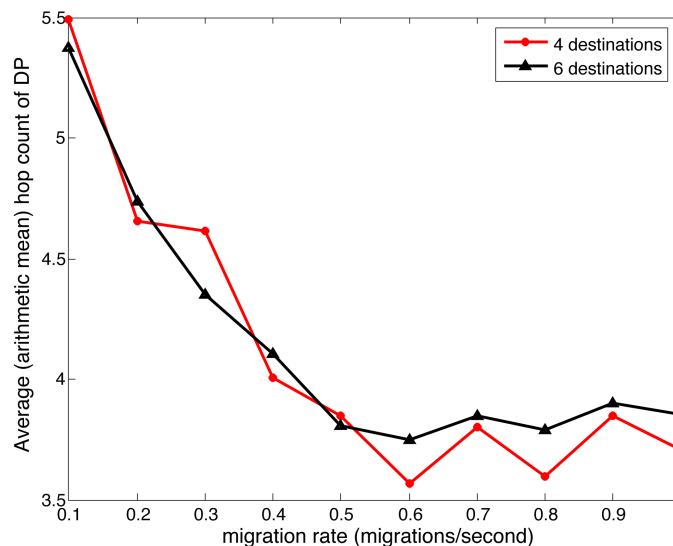


Figure 8: Measured group distances in network test-bed.

the benefits gained by virtual mobility as driven by network information made available by the underlying CPN. The experiments consisted in randomly selecting a source and k destinations on the grid test-bed, and transmitting a flow packets (representing auction traffic), at a rate of 100 Kbps for 10 seconds with a pre-selected migration rate. The smart packet rate was kept to 10% during all experiments so that both self-adaptation and self-configuration occurred during experiments. A high migration rate indicates a higher chance for finding a better location in the network as migration calculations occur more frequently. Figure 8 shows the average group distance of 4000 instances of the experiment against the migration rate (in migrations/second). The two curves represent groups of four and six destinations, and they clearly show a decrease in the average group distance as the migration rate increase.

7. CONCLUSIONS

We have discussed the structure of an autonomic networked auction system that provides participants an opportunistic advantage in terms of QoS communications. The system offers participants the benefit of self-adapting communications and self-configuring location selection via virtual mobility of the application, both of which will allow users gain comparative advantage over other participants to maximize utility. Our on-going work includes further development and experimentation with more realistic traffic sources and extended performance evaluation. Also, we expect to develop a comparable application to operate with autonomic communication elements (ACE) and extended autonomic functionality, which is currently being investigated by the IST CASCADAS project.

8. REFERENCES

- [1] G. Apostolopoulos, D. Williams, S. Kamat, R. guerin, A. Orda and T. Przygienda. *QoS Routing Mechanisms and OSPF Extensions*. RFC 2676, August 1999.
- [2] P. Bouquet, L. Serafini, S. Zanobini. *Peer-to-Peer Semantic Coordination*. Journal of Web Semantics, 2 (1), 2005.
- [3] CASCADAS project web page <http://www.cascadas-project.org>
- [4] L. Capra, W. Emmerich, C. Mascolo. *CARISMA: Context-Aware Reflective mIddleware System for Mobile Applications*, IEEE Transactions of Software Engineering Journal (TSE) 29(10):929-945, 2003.
- [5] S. Dobson, S. Denazis, A. Fernndez, D. Gati, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt and F. Zambonelli. *A survey of autonomic communications*. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 1(2):223259, 2006.
- [6] E. Gelenbe. *Users and services in intelligent networks*. In Lecture Notes in Computer Science, 3837 (3045), December 2005.
- [7] E. Gelenbe. *Cognitive Packet Network*. U.S. Patent No. 6,804,201 B1, Oct. 12, 2004.
- [8] E. Gelenbe, R. Lent, A. Nunez. *Self-aware networks and QoS*. Proceedings of the IEEE, 92 (9), pp. 1478-1489, 2004.
- [9] E. Gelenbe. *Random neural networks with negative and positive signals and product form solution*. Neural Computation, 1 (4), pp. 502-511, 1989.
- [10] E. Gelenbe, R. Lent, A. Montuori, and Z. Xu. *Cognitive packet networks: QoS and performance*. In Proceedings of the IEEE MASCOTS Conference, pages 312, Ft. Worth, TX, October 2002. Opening Keynote Paper.
- [11] E. Gelenbe. *Analysis of Automated Auctions*. 21th International Symposium on Computer and Information Sciences, 2006.
- [12] E. Gelenbe, M. Gellman, R. Lent, P. Liu, Pu Su. *Autonomous smart routing for network QoS*. In Proceedings of the First International Conference on Autonomic Computing, IEEE Computer Society, ISBN 0-7695-2114-2, 232-239, New York, 2004.
- [13] A. Hiles. *The Complete Guide to IT Service Level*

Agreements: Aligning IT Service to Business Needs.
Rothstein Associates Press, 2002.

- [14] J. O. Kephart and D. M. Chess. *The Vision of Autonomic Computing*. ACM Computer, Vol. 36, Issue 1, pp. 41-50, January 2003.
- [15] P. Klemperer. *Auctions: Theory and Practice*. Princeton University Press, 2004.
- [16] H. Liu, M. Parashar. *Component-based Programming Model for Autonomic Applications*. In Proceedings of the First International Conference on Autonomic Computing, New York, NY, USA, 2004.
- [17] A. Manzalini and F. Zambonelli, *Towards Autonomic and Situation-Aware Communication Services: the CASCADAS Vision*. IEEE Workshop on Distributed Intelligent Systems, Prague (CZ), June 2006.
- [18] M. Mikic-Rakic, N. Medvidovic. *Support for Disconnected Operation via Architectural Self-Reconfiguration*. In Proceedings of the First International Conference on Autonomic Computing, IEEE Computer Society, New York, 2004.
- [19] D. Nissanoff. *FutureShop: How the New Auction Culture Will Revolutionize the Way We Buy, Sell and Get the Things We Really Want*. The Penguin Press, 2006.
- [20] G. P. Picco, A. L. Murphy, G. C. Roman. *LIME: a Middleware for Logical and Physical Mobility*. 22nd IEEE Intl. Conference Distributed Computing Systems, 2001.
- [21] F. Sestini *Situated and Autonomic Communication an EC FET European initiative*. ACM SIGCOMM Computer Communication Review, Vol. 36, Issue 2, pp. 17-20, April 2006.
- [22] A. Sherstov and P. Stone. *Three Automated Stock-Trading Agents: A Comparative Study*. In Agent Mediated Electronic Commerce VI: Theories for and Engineering of Distributed Mechanisms and Systems (AMEC 2004), P. Faratin and J.A. Rodriguez-Aguilar, eds. Lecture Notes in Artificial Intelligence, pp. 173187, Springer Verlag, Berlin, 2005.
- [23] C. Tschudin, H. Lundgren, H. Gulbrandsen. *Active Routing for Ad Hoc Networks*. IEEE Communications Magazine, April 2000.
- [24] F. Zambonelli, N. Jennings, M. Wooldridge. *Developing Multiagent Systems: the Gaia Methodology*, ACM Transactions on Software Engineering and Methodology, 12 (3):317-370, 2003.