Scalability and Robustness of a Market-Based Network Resource Allocation System*

Nadim Haque, Nicholas R. Jennings, Luc Moreau
School of Electronics and Computer Science
University of Southampton
Southampton, UK.
{n.a.haque,n.r.jennings,l.moreau}@ecs.soton.ac.uk

Abstract

In this paper, we consider issues related to scalability and robustness in designing a market-based multi-agent system that allocates bandwidth in a communications network. Specifically, an empirical evaluation is carried out to assess the system performance under a variety of design configurations in order to provide an insight into network deployment issues. This extends our previous work in which we developed an application that makes use of market-based software agents that compete in decentralised marketplaces to buy and sell bandwidth resources. Our new results show that given a light to moderate network traffic load, partitioning the network into a few regions, each with a separate market server, gives a higher call success rate than by using a single market. Moreover, a trade-off in the number of regions was also noted between the average call success rate and the number of messages received per market server. Finally, given the possibility of market failures, we observe that the average call success rates increase with an increasing number of markets until a maximum is reached.

1. Introduction

Resource allocation is a central problem in effectively managing networks. Specifically, this covers the process by which network elements try to meet the competing demands that applications have for network resources — primarily link bandwidth and buffer space in routers or switches [12]. This is a challenging problem since resources become scarce when there is a high demand for them. In this work, we consider a circuit switched meshed network where nodes communicate with their immediate neighbours using radio links [11], where by design, they consume as little power as possible and are targeted for rapid and cost-efficient deployment in poor countries. Such low power consumption implies that there is limited bandwidth available in the network. In this context, we provide an empirical evaluation of a multi-agent system that allocates end-to-end (source-to-destination) bandwidth in such communications networks to set up calls. More specifically, based on our previous solution [7], in which we developed an application using markets, we consider a network which is partitioned into regions, each with a market server, from where resources are allocated. Using regions and decentralised markets in this way means that there is no central point of failure from where all resources are allocated. In more detail, we look at the region scalability in a fixed size network, as well as robustness where failures are induced in the network. That is, we evaluate how the system performance changes with varying number of markets. By investigating these issues, we provide a network designer with an insight into how such a network can be deployed in practice.

In recent years, market-based approaches have been investigated and used to solve various problems in areas of computing, where applications and systems have also successfully been developed. Areas in which markets have been studied include allocating resources in computational grids [1], development of peer-to-peer systems [14], supply-chain management [10], scheduling [17], congestion control [8], routing [6], workflow automation [9] and recommender systems [16], amongst other areas. The solution that we developed consists of software agents that compete in a marketplace to buy and sell bandwidth. Our previous work [7] described the system and indicated the broad feasibility of our approach. Here, buyer agents represent callers and seller agents represent the owners of the resources. We decided to base our solution on agents for a number of reasons. First, their autonomous behaviour allows them to carry out their tasks in the decentralised control regime

*The research in this paper is part of the EPSRC funded Mohican Project (Reference no: GR/R32697/01). We would also like to acknowledge the contribution of Steve Braithwaite who provided us with domain expertise.

of distributed marketplaces. Second, the reactive nature of agents is needed to respond to requests quickly so that calls within the network can be made with minimum delay. Third, agents have the ability to flexibly interact which is important in our system because the agents need to bid against a variety of different opponents in an environment where the available resources vary dynamically. A market-based approach was chosen for the following reasons. First, markets are effective mechanisms for allocating scarce resources in a decentralised fashion [2]. Second, they achieve this based on the exchange of small amounts of information (such as prices). Finally, they provide a natural way of viewing the resource allocation problem because, generally speaking, they ensure the individual who values the resources the most will obtain them.

Our system in [7] was a distributed market mechanism in which allocations of *interrelated* resource bundles were sold in multiple markets. The marketplace protocol incorporates a reservation and commitment mechanism that provides a guarantee that resources will not be bought unnecessarily. Now in this paper, we extend our previous work by analysing the system behaviour after performing scalability and robustness tests, with respect to increasing the number of regions in a fixed size network. The remainder of this paper is structured as follows: the marketplace design and components are recapped in section 2. The system evaluation and experimental results are presented in section 3. Section 4 describes related work and, finally, section 5 concludes.

2. Marketplace design

This section describes the design of the system. Specifically, the basic components and the network model are outlined in section 2.1. Section 2.2 describes the constituent agents and, finally, section 2.3 provides a brief description of how resources are acquired in a multi-region call.

2.1. Network model

The system consists of three types of agents: seller, buyer and auctioneer (see figure 1). Seller agents are responsible for selling node bandwidth capacity resources and buyer agents are responsible for buying these resources. The auctioneer agents accept *asks* from seller agents and *bids* from buyer agents and conduct auctions so that resources can be allocated using a market-based protocol. As can be seen in the figure, the overall network is divided into a number of regions (3 in this case). Callers are used to initiate calls via the use of handsets. When a call request takes place, the destination location to where the caller wishes to make the call is passed to the buyer agent on the local node. This agent then starts the process of setting up the call. For each

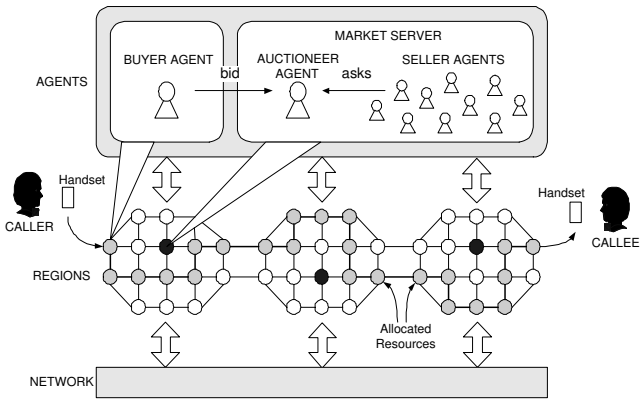


Figure 1. An overview of the system architecture. Black nodes in regions represent market servers and grey nodes represent allocated resources for a particular call from the caller to the callee.

call attempt, a buyer agent in each required region tries to reserve a resource bundle (i.e. set of interrelated resources in a single region) from its local market server. Buyer agents work together to collectively make a complete source-to-destination path across the regions using the bundles. If a resource bundle cannot be obtained, then a backtracking mechanism is used which allows alternative allocations to be made if currently reserved bundles cannot lead to the final destination.

It is desirable for resources to be bought and sold in the network from various points and not from a central location (since a single server would constitute a central point of failure). With this in mind, we partitioned the network into regions where only resources within those regions are sold (i.e. there are multiple market servers in the network, one placed in each region). Thus, if we are to look into the scalability of the number of regions within a fixed size network (i.e. the region scalability), we must consider how the network should be divided. Here we regard a network region as a group of nodes that are situated geographically close together. In this application, we consider a static network configuration as defined at deployment time. Nodes on the edge of regions can communicate with other edge nodes in neighbouring regions. In using local markets, resource information does not have to be replicated across all markets in the network. This is good since the market server that receives a bid from a buyer does not need to contact *all* other markets to make sure that the same resources are not being sold elsewhere, for each bid placed.

To model the network, each node has a fixed total bandwidth capacity that is split logically into several *equal* parts, where these are the resources that are bought and sold by

the market mechanism. These resources are used in relaying several calls at the same time through the nodes. Each node has a fixed number of handsets attached from where calls originate. A handset that is currently in use is assumed to be engaged and, thus, cannot be used for any other calls at the same time. Our current work assumes that control capacity is separate from the bandwidth capacity used for relaying calls. The resources we consider are for calls and not for control messages. In this work, we do not look into the usage of control capacity and leave this investigation for future work. However, in section 3.3, we do look at the number of messages received per market server, since it is on these servers that the majority of the processing takes place for running the auctions.

2.2. The agents and the markets

Auctioneer agents conduct auctions using a combinatorial reverse auction protocol [15] to allocate goods (units of node bandwidth) to buyers (i.e. they allocate a *combination of goods* that consist of the cheapest possible bundles). There is one auctioneer agent per region in the network, each on their respective market server nodes. Auctioneer agents execute a *winner determination protocol* that determines which resources are allocated to which parties, for each bid submitted.

There are several seller agents per region, one owning each node, where they each submit an individual ask price to their local markets. The implication of each seller agent owning a node is that they can attempt to compete against each other by pricing their respective resources competitively. To minimise communication in the network, all seller agents are physically deployed on their local market server nodes and we assume that, currently, they all use the same pricing strategy. A seller agent begins with a total of y resource units initially priced at one price unit each. For each unit sold, the price increases by one price unit (i.e. when there is only one resource unit left, it should cost y price units). Conversely, for each unit reclaimed by a seller, the price reduces by one price unit.

The initial low price of one price unit is chosen so that sellers can sell resources more easily to begin with. As demand for resources increases, the price per unit increases so that buyer agents have to bid more for resources. Given this, seller agents can maximise their utilities by making as much profit as possible. They also reduce the price of resources by one price unit when they have reclaimed the resource so that they can lure more buyers to purchase resources from them in the future. This allows seller agents to remain competitive against each other when pricing their resources.

A buyer submits a bid composed of several bundles, of which only one is required. The winner determination algorithm then attempts to allocate resources by minimising the amount spent. From these bundles, the cheapest available one is allocated to the buyer agent. If a buyer agent's bid is successful, resources are sold at the asking prices of the seller agents. There is one buyer agent placed on each individual node where they await call requests, from callers, from any point in the network. We assume that all buyer agents use the same purchasing strategy. Thus, when a buyer agent receives a request for purchasing node bandwidth, it formulates its bid. It attempts to find the cheapest set of routes that lead from its current node to a destination node within its own region. Now, in this work, we assume that buyer agents select a set of bundles that minimise the length of their desired routes. The intuition here is that the buyer believes shorter routes are generally cheaper since they contain fewer resources.

We make the assumption that buyer agents are only allowed to submit up to a certain number of bundles for each bid. The value chosen here must be enough to allow some flexibility in the bundle that a buyer could be allocated, but it should not be so high that the market algorithm has to do significant amounts of unnecessary processing. If the final destination node is within the same region, that node is the destination node. The bundles selected by a buyer agent are sent as a bid to the buyer's local market. Finally, if the buyer agent is successful in reserving resources, it is informed by the local market. Callers are assumed to pay a fixed amount per region for calls that are successfully established where the cost per region is proportional to the number of resources in the region.

2.3. Acquiring resources across regions

The number of buyer agents required in setting up a call is the same as the number of regions in which resources are required for a given call. If the final destination for a call is in a different region from the one in which a buyer is located in, then this buyer will attempt to find routes that lead to a node within its region that is connected to a node in a neighbouring region that leads to the final destination. Then, in a multi-region call, once a buyer agent has successfully reserved a bundle of resources, the market server in that region is responsible for contacting a buyer agent that is on the edge of the next region. The node on which this second buyer agent resides must be in reach of the last node in the bundle of resources that have been reserved in the previous region. Thus, these boundary nodes will relay the call from the initiating source node to the final destination node, such that there is a continuous path when/if the call eventually takes place.

Figure 2 illustrates part of the reservation process. This figure shows a buyer agent, b1, which has already successfully been allocated a resource bundle (shown by the grey nodes) by its local market server within its region (region 2

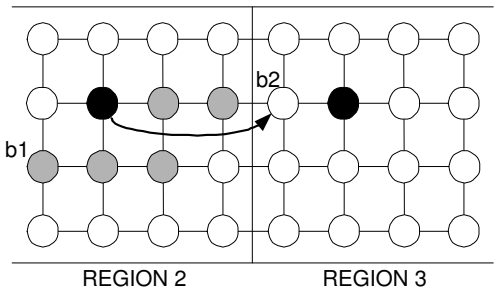


Figure 2. A market server in region 2 (black node) contacting a buyer agent, b2, in the following region for setting up a multi-region call. Grey nodes represent a reserved resource bundle for b1 in region 2.

in this case). This market server in region 2 then attempts to contact another buyer agent, b2, which is on a boundary node in the following region so that b2 can then bid for a resource bundle within its own region (region 3). The reservation process continues in each required region until the final node is reached and the call is set up. In general, the system algorithm allows buyers to choose which region to contact next when there is a choice of following regions in a multi-region call. In such a case, the regional route that involves the use of the fewest regions for the set up of a call is preferred (i.e. the shortest regional route).

Once the final destination has been reached, the market server in the last region sends a *commit* message to the buyer agent within its own region. This buyer agent then contacts the market server in the previous region which, in turn, informs its buyer agent and so on, until the initial region is reached. Eventually, the originating buyer agent receives the commit message and the call can be placed. Thus, there is a reservation and commitment process that takes place in the system, where payment for resources only takes place during the commit phase once all of the necessary resources have been acquired. When the call has completed, a message is sent from the initial buyer to its local market (and to all other markets and buyers involved in this call in the direction of the final region) to signal that resources can be released. The markets then resell the resources to buyers that place bids for them in the future.

The system uses a backtracking mechanism that allows alternative allocations to be made if currently reserved resource bundles cannot lead to the final destination. Thus, if a buyer agent in an intermediate region fails in reserving a bundle of resources, it can resubmit another bid to its local market which contains bundles that lead to another destination node within its own region (i.e. to a different boundary node). This continues until a bundle has been reserved or there are none available. Thus, agents perform a distributed search for resource bundles. An example of the backtracking mechanism used to find alternative routes via different boundary nodes and regions was given in our previous work [7].

3. Experimental evaluation

This section describes the experimental work that was carried out in evaluating the scalability of the system with respect to increasing number of regions as well as the robustness of the system. Section 3.1 describes the methodology and parameters used, while results are outlined in sections 3.2 to 3.5.

3.1. Experimental methodology and settings

Previously [7], we looked at the initial results of the system algorithm (i.e. the *average call success rate* and *average call set up time* when compared against optimum and random strategies). This provided us with an insight into a fundamental measure of the percentage of successful calls and set up times, respectively, given one particular network setting. However, the aim of our current work is to evaluate the system in terms of region scalability and robustness. Looking at the former, will give us an understanding of how the structuring of the network into regions impacts the performance for a given network size. Testing for the latter, will show how well the system performs with market failures.

In more detail, to test for scalability, we measured the *average call success rate* when progressively increasing the number of regions in a fixed size network (see section 3.2). This was chosen as a measure because it provides a fundamental insight into the percentage of calls that are successfully established from all calls attempted, with call establishment being the primary aim of our system. We also look at the number of messages received per market server in the network when scaling up the number of regions (described in section 3.3). This is also important since this is where resources are auctioned and thus, where the majority of processing occurs. Thus, these are two measures that a network designer would be interested in considering when deploying a network. By obtaining results from these two experiments, we then look at the trade-off between the average call success rate and the number of messages received per market server in order to find an optimum number of regions in which the network should be partitioned into (see section 3.4).

Finally, the average call success rate was also measured for robustness testing, after a market failure was introduced within the network (described in section 3.5). Again, as with scalability testing, the average call success rate is an equally applicable measure to investigate for robustness

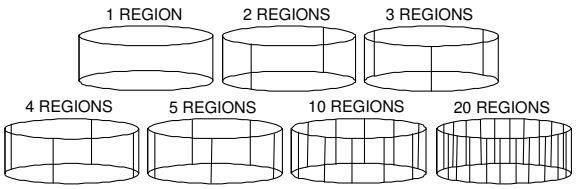


Figure 3. Set of seven tori used where each consists of an 80-node (20-by-4) network but is split into a different number of regions.

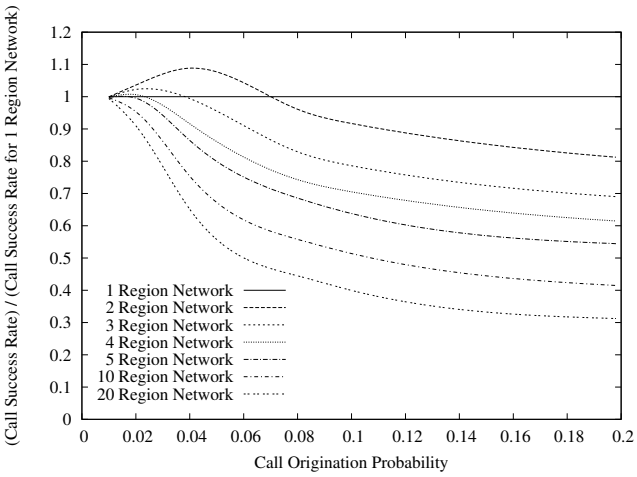


Figure 4. Average call success rate over the 1 region set up average call success rate, as the call origination probability is varied.

testing, since looking at the successful establishment of calls is just as important given a market failure. For the sake of consistency, we chose to investigate a *single* market failure throughout all of the robustness testing in this particular experiment. Here, we assume that all buyers have knowledge of the failed market server and, thus, we do not deal with failure detection. In all experiments discussed in sections 3.2 to 3.5, the network load was increased by varying the *call origination probability* (i.e. the probability of a call originating from any given unused handset).

For our experiments, several different network set ups were used. In each set up, the same underlying network topology was used but it was partitioned into a different number of regions with a market server in each region (market servers are placed manually within a central location in their regions where there is a high connectivity of neighbouring nodes). Thus, all of the experimental set ups use an 80-node (20-by-4) network. This was chosen because it demonstrates a topology which can be partitioned vertically with ease into several regions such that it is easier to evaluate the system for scalability. For each set up, the network was wrapped around and shaped into a taurus, as shown in figure 3. Also, all calls made were unidirectional (i.e. calls travel in only one direction around the taurus). The combination of joining the beginning and end of the network to form a taurus and allowing calls to traverse the network in one direction ensures fairness in our experiments for two reasons. Firstly, the number of types of different distance calls made in any set up are the same. For example, given the five region set up, there are exactly five different types of calls that require a single region, five different types that span across two regions, five types of triple region calls, and so on. Secondly, by using a taurus, the average load in any region is the same, since there is no central region through which any extra calls traverse. This provides an even setting for testing the system.

The experimental settings we used in this evaluation were obtained from a domain expert. Specifically, each experiment was run for a total of 100,000 time steps and probed after every 1,000 time steps. The duration of a call was set to 1,000 time steps. We assume that each node has 2 handsets attached to it and has a total of 10 units of bandwidth capacity available, allowing each node to relay up to 10 simultaneous calls at any one time. Calls were made to originate after every 50 time steps. Buyer agents were allowed to submit bids that contained a choice of 5 bundles from which an attempt was made to allocate the cheapest one available. Finally, the number of simulation runs for each experiment was sufficient for the results to be *statistically significant* at the 95% confidence level.

3.2. Average call success rate

The purpose of the first experiment was to investigate the proportion of calls that could successfully be connected for each network set up, shown in figure 3, when varying the call origination probability. (The network is regarded as being heavily loaded when the call origination probability is 0.2 (20%). This gives a typical network occupancy of 82%). As can be seen from figure 4, in general, call success rates are higher when there are fewer regions. (For the sake of clarity in figure 4, all call success rates are divided by the 1 region call success rate). However, another observation can be made from figure 4. Given a network load that is light to average where the call origination probability is below 0.07 (7%), we can see that the 2 region set up gives a higher call success rate than the single region case of up to 10% more. Also, when the call origination probability is below 0.04 (4%), the 3 region network set up provides a higher average call success rate than the 1 region centralised case,

by about 3% more.

The reason for our observations can be explained as follows. When the network is light to moderately loaded (i.e. when the call origination probability is below 0.07), there are sufficient resources available for setting up a large proportion of the calls. For the multi-region set ups (i.e. when there are 2 or more regions in the network), resources need to be acquired across *several* regions. Here, the search for resource bundles is exhaustive across boundary nodes in intermediate regions — buyers will continue to bid until either a bundle is found or there are none available, except in the final region where only a single bid is made to the final destination node. The search for resources in the 1 region set up is similar to the search that takes place in the final region of a multi-region call (except that on average, the required resource bundle is larger in the 1 region case because the size of the region is larger). Thus, no exhaustive searching takes place for resource bundles across boundary nodes for the 1 region case since there is only a single region and, therefore, if the initial attempt fails, then the call set up is unsuccessful.

When the call origination probability increases, resource contention increases for all network set ups (i.e. node bandwidth becomes more scarce). For multi-region cases, the search becomes more exhaustive, where resources are reserved for longer periods of time, while a call is being set up. Consequently, this reduces the chance of other calls being made. This is not an issue with the 1 region network and therefore, it is only at this point that the single region network set up begins to perform better than all of the multi-region network set ups.

3.3. Market server load

Our next experiment investigates the number of messages received per market server, as the number of regions is scaled up. There are several different types of messages that are received by the market servers. These include, buyer bids, seller asks, commit messages and messages that tell the market servers to release resources when the resource usage is complete. We do not differentiate between these, but rather simply sum them across all market servers and divide by the number of market servers, for each network set up, to find the average number of messages received by each one. Our hypothesis was that there would be a lower number of messages received by a market server, per simulation run, as the number of network regions increases. This is important because if market servers receive fewer messages, the load on a server is less and, therefore, less auction processing needs to take place.

Figure 5 shows that our hypothesis is true and that when there are more market servers in the network, on average, the number of messages received per market server does

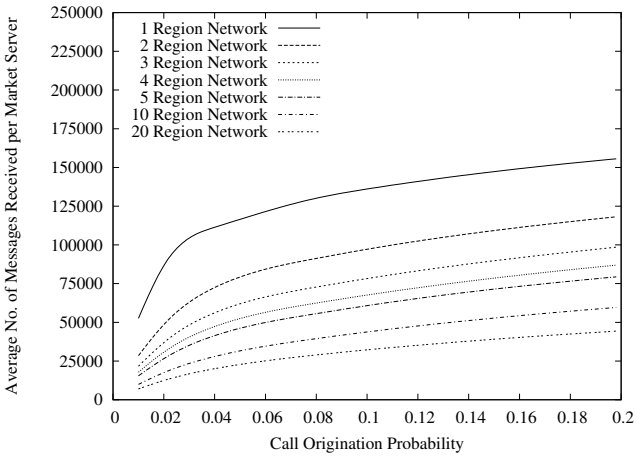
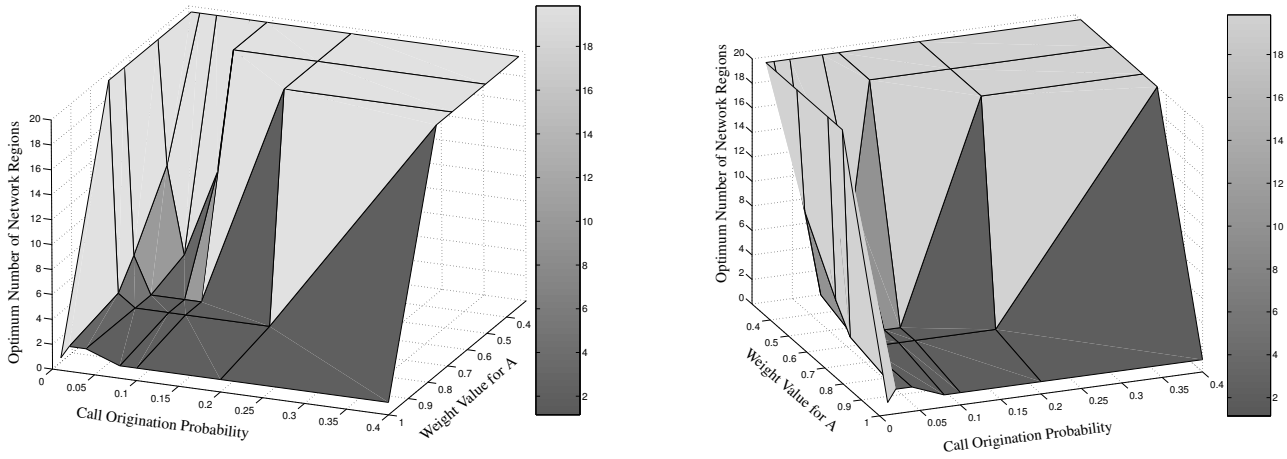


Figure 5. Average number of messages received per market server as the call origination probability is varied.

indeed decrease. Using a 1 region network, with a call origination probability of 0.1, 136,000 messages are received by the single market server whereas, in contrast, with 2 regions, there are 97,000 messages per server and with 20 regions, there are just 32,000 messages per server. A similar pattern is observed at all call origination probability values. The reason for this result can be explained as follows. As the number of regions increases, the number of nodes per region decreases, since the overall size of the network remains fixed. Since buyers and sellers submit bids and asks respectively, only to their own local market servers, the number of such messages becomes less per market. Thus, with more regions, the number of messages received per market is less (i.e. the load is distributed more amongst market servers, which is desirable, since less processing is required per server).

Whilst the total number of messages received by *all* market servers in each network set up is higher with an increasing number of regions, this sum of messages does not increase by a disproportionate amount when, say, the number of regions doubles. For example, at a call origination probability of 0.04, there is only a 35% increase in the total number of messages received by all market servers from the 5 region network set up to the 10 region case. Also, at a lower call origination probability of 0.01, there is only an 8% increase in the total number of messages from the single region case to the 2 region network set up. However, it still remains the case that the average load per market server decreases with increasing number of regions and this is a useful result, since it is the processing *per* market server that we wish to minimise.



(a) 3-Dimensional, left-sided view

(b) 3-Dimensional, right-sided view

Figure 6. Optimum number of regions, as call origination probability and weight value for A are varied.

3.4. Optimum number of network regions

Our result from figure 4 shows that, given specific call origination probabilities, the average call success rate is generally higher when the network is partitioned into a smaller number of regions. Alternatively, figure 5 showed that the number of messages received per market server decreases per market when there are more regions in the network. Receiving less messages is better because fewer auctions need to take place and this, therefore, requires less processing power and fewer input buffers. Thus, figure 4 (average call success rate), shows that less regions is better and figure 5 (messages received per market server), indicates that more regions gives a better performance.

Given these results, we would like to find the trade-off between the average call success rate and the number of messages received per market server. Our motivation is to help engineers in deploying such a network to find an optimum number of regions in which to partition the network. In order to achieve this, we first normalise the y-axis on both figures 4 and 5 so that they both range between 0 and 1, where we consider these as the utilities for each measurement. (In the case of figure 5, we subtract each normalised value from 1 to obtain the utility values, since receiving fewer messages per market server should give a higher utility). Then, for each of the network set ups and varying call origination probabilities, we consider the overall utility defined as the weighted sum of the utilities of the average call success rate and the number of messages received per market server: A $\times$ u(c) + B $\times$ u(m) such that u(c) is the utility of the average call success rate with weight A, u(m) is the utility of the number of messages received per market server with weight B and where A + B = 1. The values for A and B reflect the importance that a network designer would assign to the two different measures. Figure 6 shows a 3-dimensional surface plot where the optimum number of regions is plotted against the call origination probability with varying values of weights A and B. For the sake of clarity, two different views are shown of the same plot in figure 6. Several observations can be made from this figure.

When the network designer believes that receiving fewer messages per market server is more important than the average call success rate in the system (i.e. when B $>$ A), the optimum number of regions to deploy is 20, across all call origination probabilities. This observation can be explained by the fact that a lower value for A means more importance is given to the number of messages received per market server, where the utility is higher when there are more regions.

When the network designer gives an equal importance to the average call success rate in the system and to receiving fewer messages per market server (i.e. when A = B = 0.5), the optimum number of regions that should be deployed decreases from 20 to 10, 3, 10 and then back to 20, given call origination probabilities of 0.01, 0.02, 0.04, 0.08 and 0.1, respectively. Thus, a minimum is seen across the optimum set of regions when A is 0.5. A similar pattern is observed when the importance given to average call success rate increases, thereby preserving the minimum. When a higher weighting is given to the average call success rate, the optimum number of regions drops further. The reason for this is that with intermediate call origination probabilities between 0.02 and 0.08, the difference in the average call success rates begins to widen with an increasing number of regions.

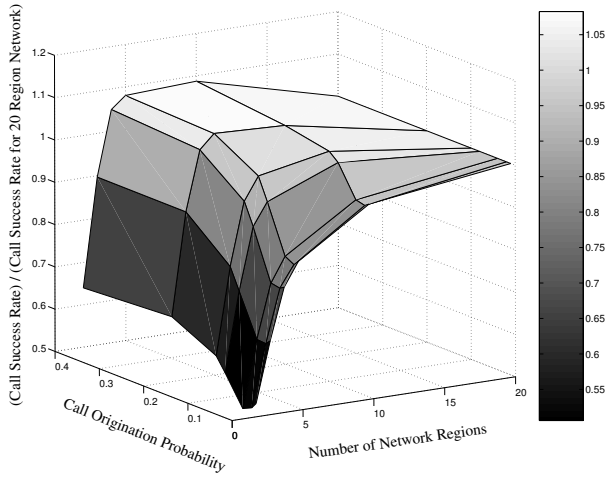


Figure 7. Average call success rate over the 20 region set up average call success rate, as the call origination probability is varied and where there is a single market failure.

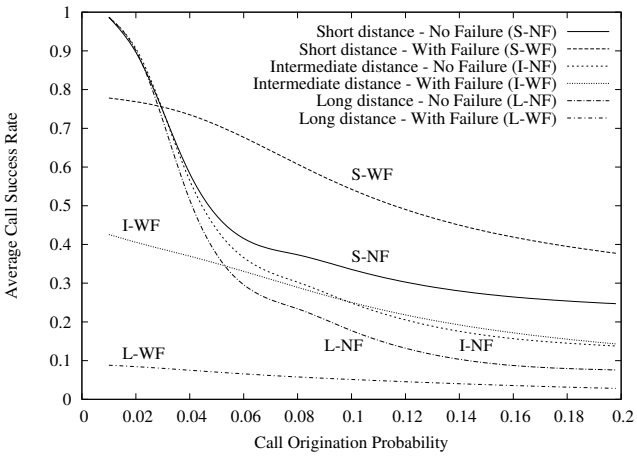


Figure 8. Average call success rate for different distance calls for the 10 region network, with and without a single market failure, as the call origination probability is varied.

This is also shown in figure 4 where the largest change in average call success rate occurs between these values of call origination probability. Thus, it becomes more evident that the fewer regions there are, the higher the average call success rate. In addition to this, because there is a higher weight value for A than there is for B, the optimum number of regions to deploy begins to decrease.

Finally, when exclusive importance is given by the network designer to the average call success rate (i.e. when A = 1 and B = 0), deploying a single region gives the best overall performance at all call origination probabilities except at 0.07 or below, where the 2 region network set up is best. Figure 6 shows that at these call origination probabilities, there is a maximum when the 2 region network provides the optimum solution. (Section 3.2 provided an explanation for why the 2 region network gave a higher average call success rate at these call origination probabilities).

3.5. Average call success rate with failures

The purpose of our final experiment was to investigate how a single market failure in the network can affect the average call success rate when the number of regions is scaled up. Our hypothesis was that the average call success rate would be higher as the number of regions is scaled up, for any value of call origination probability. Figure 7 shows that this was indeed the case when the number of regions was increased between 2 and 10 regions. (For the sake of clarity in figure 7, all call success rates are divided by the 20 region call success rate). In the 1 region set up, there are no calls set up since there is no market server functioning. As the number of regions is scaled up between 2 and 10, the average call success rates increase, albeit at a progressively lower rate (i.e. the difference in the average call success rates between the 5 and 10 region set ups is very close when the call origination probability exceeds 0.12). When doubling the number of regions from 10 to 20, given a call origination probability below 0.06, the average call success rate increases. However, when the call origination probability exceeds 0.06, the call success rate decreases from the 10 region set up to the 20 region set up. As a result of this, a maximum can be seen between 5 and 10 regions (as shown in figure 7).

These results can be explained by the following. When the number of regions increases and there is a *single* market server failure, the *maximum* percentage of calls that can take place increases, but at a progressively lower rate. For example, with a 2 region set up, up to 25% of calls can take place when there is a market server failure, a maximum of 40% of calls can take place with 5 regions and 45% with 10 regions. Thus, there is a steady increase in the call success rates between 2 to 10 regions, with the difference between 5 and 10 regions being marginal. This shows that the impact on the performance is affected more, with a market failure, when there are fewer regions in the first instance. However, in the 20 region case, even though a maximum of 47.5% of calls can be made, the 10 region case performs better because contention for resources increases more in the 20 re-

gion set up than in the 10 region case. Thus, given that there is a maximum between the 5 and 10 region set ups, a network designer should ideally deploy a network which has between 5 and 10 regions if a market failure is anticipated.

During our investigation, we also discovered an additional result when looking at the 10 and 20 region set ups, in the absence and presence of failures. For the 10 and 20 region set ups, we found that the overall call success rate is marginally higher with a failure than when there is no failure. This can be explained by means of figure 8, which shows calls of different distances in the network for the 10 region set up, with and without failures. Short distance calls span across 1 and 3 regions, intermediate calls between 4 and 7 regions and long distance calls cover 8 to 10 regions. The percentage of long distance calls for the failure case is less than without failures because the number of these calls is restricted more in the failure case (i.e. when there is a single failure, no calls which span across all 10 regions can be made, for example). Thus, there are more resources available in the network for shorter distance calls to be more successful and indeed, figure 8 shows that the success rate of such calls is greater for the failure case beyond a call origination probability of 0.03. Finally, figure 8 also shows that the number of intermediate distance calls are approximately the same when the call origination probability exceeds 0.08.

4. Related work

There are several market-based architectures that have been proposed for allocating resources in a distributed environment. Gibney and Jennings [5] describe a system in which agents compete for network resources in distributed markets so that calls can be routed in a telecommunications network. The system used a double auction protocol [18] with sealed bids and provided good utilisation of the network where the load was also balanced. However, a drawback of this system was that if some resources on a path were already bought and the next desired resource could not be obtained, then the resources already bought could become redundant and money would be spent unnecessarily. In contrast, our reserve/commit mechanism ensures that this situation is avoided by releasing unused resources immediately and allowing payment to occur only after all necessary resources have been successfully reserved.

The *Global Electronic Market System* (GEM) [13] is a framework for decentralised markets across the Internet. GEM has a single market which is distributed on which goods are sold. In GEM, the markets are replicated and the order for goods is distributed across these markets. Looking at GEM provided an insight into one method of how servers in a market-based resource allocation system could be distributed. However, the approach taken by GEM of replicating the resource information is not suitable for our system because it would induce more messages in the network than our partitioned approach (as was outlined in section 2.1).

MIDAS [3] is an auction-based mechanism that allocates link bandwidth in a network for making paths. Simultaneous multi-unit Dutch auctions were used as the protocol for allocating resources. This protocol would be inadequate for our requirements since it is not capable of allocating several interrelated goods at the same time.

Finally, Ezhilchelvan and Morgan [4] have looked at how an auction system can be distributed across several servers in a network of servers. However, their approach assumes that communication takes place using a high-bandwidth network which is an assumption that does not hold within our work.

5. Conclusions and future work

In this paper, a system was described that allocates end-to-end bandwidth to set up calls in a network using market-based agents. The system used a combinatorial reverse auction where bundles of interrelated resources were allocated. Scalability testing was performed with respect to increasing the number of regions in a fixed sized network. In conclusion, results show that if light to average network loads are anticipated, then the network designer should consider deploying a decentralised approach with a few number of regions (i.e. 2 or 3 regions) rather than opting for a completely centralised system with a single market. We also see that the average number of messages received per market server is less as the number of regions is scaled up, and thus, allowing the processing of bids and the message load to be distributed better amongst the market servers when there are more of them.

In addition, we found the trade-off between the average call success rate and the number of messages received per market server, with respect to the optimum number of regions in which the network should be partitioned. Results showed that if the network designer assigns a higher priority to receiving fewer messages per market server than the average call success rate in the system, then the optimum number of regions to deploy is higher, at all values of call origination probability. When the level of importance for the average call success rate and the number of messages received per market server are considered to be the same by the designer, the optimum number of regions to deploy decreases progressively, at intermediate call origination probabilities. If a network designer wishes to associate a greater importance to the average call success rate than the number of messages received per market server, the general trend that shows this minimum continues.

Robustness testing was also performed by inducing a single market server failure. This showed that the call success rate was higher as the number of regions increased until a

maximum was reached between 5 and 10 regions, beyond which the call success rate decreased. Therefore, if it is envisaged that a market server failure is likely to occur, then the network designer should consider deploying a network which has between 5 and 10 regions. Finally, during the course of our investigations, we also found that the average call success rate is slightly higher given a single market failure than when there is no market server failure in the network at all. This was explained by the fact that more longer distance calls were restricted in the failure case, thereby allowing many other shorter distance calls to take place with the remaining resources.

There are several investigations that we would like to look into for future work. Firstly, we aim to perform further scalability tests but using networks with various different topologies. Currently, we are investigating the use of other network topologies where it is envisaged that this will provide an insight into how well the system scales, in terms of regions, given these different topologies. Secondly, we plan to look at more robustness tests by introducing multiple market failures in the network to see what additional affect this has on the call success rate. We will also investigate the amount of control capacity used on the nodes in the network, since this will also be in contention. Performing such an investigation would provide a network designer with insight into how much control capacity nodes in the network should with deployed with in the first instance. Thirdly, we aim to impose a restriction on the number of messages that a market server can receive and process within a given amount of time. The purpose of this will be to see how well the system performs, with varying number of markets, when such a limit is imposed. Finally, we would also like to look into issues related to resource pricing in more detail in order to investigate the gains made by buyer and seller agents as a result of calls being set up.

References

[1] L. ChunLin and L. Layuan. A two level market model for resource allocation optimization in computational grid. In *CF '05: Proceedings of the 2nd conference on Computing frontiers*, pages 66–71, New York, NY, USA, 2005. ACM Press.

[2] S. H. Clearwater. *Market-Based Control: A Paradigm For Distributed Resource Allocation*. World Scientific Publishing Co. Pte. Ltd, Covent Garden, London, 1996.

[3] C. Courcoubetis, M. Dramitinos, and G. D. Stamoulis. An auction mechanism for bandwidth allocation over paths: New results. Technical report, London, UK, June 2001.

[4] P. D. Ezhilchelvan and G. Morgan. A dependable distributed auction system: Architecture and an implementation framework. In *International Symposium on Autonomous Decentralized Systems*, pages 3–10, 2001.

[5] M. A. Gibney and N. R. Jennings. Dynamic resource allocation by market-based routing in telecommunications networks. In S. Albayrak and F. J. Garijo, editors, *Intelligent Agents for Telecommunication Applications — Proceedings of the Second International Workshop on Intelligent Agents for Telecommunication (IATA'98)*, volume 1437, pages 102–117. Springer-Verlag: Heidelberg, Germany, 1998.

[6] M. Goemans, L. E. Li, V. S. Mirrokni, and M. Thottan. Market sharing games applied to content distribution in ad-hoc networks. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 55–66, New York, NY, USA, 2004. ACM Press.

[7] N. Haque, N. R. Jennings, and L. Moreau. Resource allocation in communication networks using market-based agents. *International Journal of Knowledge Based Systems*, 18(4-5):163–170, August 2005.

[8] T. M. Heikkinen. On congestion pricing in a wireless network. *Wireless Networks*, 8(4):347–354, 2002.

[9] J. Hulaas, H. Stormer, and M. Schnhoff. Anaisoft: An agent-based architecture for distributed market-based workflow management. In *Proceedings of the Software Agents and Workflows for Systems Interoperability workshop of the Sixth International Conference on CSCW in Design*, June 2001.

[10] P. W. Keller, F. Duguay, and D. Precup. Redagent-2003: An autonomous market-based supply-chain management agent. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1182–1189, Washington, DC, USA, 2004. IEEE Computer Society.

[11] P. Nicopolitidis, M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis. *Wireless Networks*. John Wiley & Sons Ltd, Chichester, England, 2003.

[12] L. L. Peterson and B. S. Davie. *Computer Networks: A Systems Approach*. Morgan Kaufmann Publishers Inc, San Francisco, California, 2003.

[13] B. Rachlevsky-Reich, I. Ben-Shaul, N. T. Chan, A. W. Lo, and T. Poggio. GEM: A global electronic market system. *Information Systems*, 24(6):495–518, 1999.

[14] O. Ratsimor, T. Finin, A. Joshi, and Y. Yesha. encentive: a framework for intelligent marketing in mobile peer-to-peer environments. In *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*, pages 87–94, New York, NY, USA, 2003. ACM Press.

[15] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Winner determination in combinatorial auction generalizations. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 69–76, Bologna, Italy, July 2002.

[16] Y. Z. Wei, L. Moreau, and N. R. Jennings. A market-based approach to recommender systems. *ACM Transactions on Information Systems (TOIS)*, 23(3), 2005.

[17] M. P. Wellman, J. K. MacKie-Mason, D. M. Reeves, and S. Swaminathan. Exploring bidding strategies for market-based scheduling. In *EC '03: Proceedings of the 4th ACM conference on Electronic commerce*, pages 115–124, New York, NY, USA, 2003. ACM Press.

[18] P. Wurman, W. Walsh, and M. P. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24:17–27, 1998.