Ant-like agents for load balancing in telecommunications networks

Ruud Schoonderwoerd[1,2], Owen Holland[3] and Janet Bruten[1]

[1]Hewlett-Packard Laboratories Bristol
Filton Road, Stoke Gifford, Bristol BS12 6QZ, United Kingdom

[2]Delft University of Technology
Julianalaan 132, 2628 BL Delft, The Netherlands

[3]University of the West of England
Coldharbour Lane, Bristol BS16 1QY, United Kingdom

ruud@hplb.hpl.hp.com o-hollan@uwe.ac.uk jlb@hplb.hpl.hp.com

Abstract

This paper describes a novel method of achieving load balancing in telecommunications networks. A simulated network models a typical distribution of calls between arbitrary nodes; nodes carrying an excess of traffic can become congested, causing calls to fail. In addition to calls, the network also supports a population of simple mobile agents with behaviours modelled on the trail laying abilities of ants. The agents move across the network between arbitrary pairs of nodes, selecting their path at each intermediate node according to the distribution of simulated pheromones at each node. As they move they deposit simulated pheromones as a function of their distance from their source node, and the congestion encountered on their journey. Calls between nodes are routed as a function of the pheromone distributions at each intermediate node. The performance of the network is measured by the proportion of calls which fail. The results are compared with those achieved by using fixed shortest-path routes, and also by using an alternative algorithmically-based type of mobile agent. The ant-based system is shown to drop fewer calls than the other methods, while exhibiting many attractive features of distributed control.

1 Introduction

This paper examines the potential for using mobile software agents modelled on ants for load balancing in telecommunications networks. It is organised as follows:

- load balancing is described
- a previous attempt at using mobile agents for load balancing is summarised
- the potential appropriateness of ant based models is noted
- an ant based model is derived from a principled minimalist standpoint
- the model is tested on a network simulation
- the load balancing abilities of the ant based model are analysed and compared with those of an alternative mobile agent model.

Agents'97 Marina del Rey CA USA

1.1 Load balancing

For economic and commercial reasons, circuit switched telecommunications networks are equipped not with a level of equipment which will guarantee successful call connection under all possible circumstances, but with some lower level which will give acceptable performance under most conditions of use. If there is some significant change in the conditions - for example, if the total call volume at any time is unusually high, or if some particular location is suddenly the origin or destination of an unusually large volume of calls - then these capacity limitations often lead to the system failing with calls unable to be connected.

Calls between two points are typically routed through a number of intermediate switching stations, or nodes, each with limited capacity; if an intermediate node is already fully occupied, no new call can be made through it, and any attempt to make such a call will fail. However, in a large network, there are many possible routes between two given points. It is therefore possible to avoid or relieve actual or potential local congestion by routing calls via parts of the network which have, or are likely to have, spare node capacity. Load balancing is essentially the construction of call-routing schemes which distribute the call traffic over the system in such a way that nodes are rarely fully occupied and failures in call placement occur only infrequently. Such schemes can either be static or dynamic: a static routing scheme fixes on a particular set of 'good' routes, and maintains them regardless of any variations in congestion; a dynamic routing scheme changes the routes from time to time as a function of previous and ongoing congestion.

Before looking at methods of load balancing, it will be useful to explain how routing is usually organised. There are two broad possibilities. In the first, a call from a node A to a node Z could be assigned a route to Z specifying at the outset the exact sequence of nodes to be visited on the way. In the second, the call could be tagged with the destination 'Z' and passed from A to the neighbour node specified by A as the one used for destination Z - say node D. D would read the tag and pass the call to whichever of its neighbour nodes it uses for traffic to Z, and so on, until the call reaches Z. This paper deals only with the second method, which is in widespread use. The structure within each node specifying the next node to be used for traffic for each destination is known as a routing table. Clearly, the set of routing tables within a given network must be complete, with a valid route for every possible call within the network. If every cell in every routing table contains an entry giving the address of a neighbour node, then every invalid route must terminate in a loop. An attempt to make a call on a looped route will be relayed round the loop until it uses all the capacity within one of the nodes within the loop, causing the call to fail, and temporarily interfering with the available capacity on

all nodes within the loop.

Managing a dynamic routing scheme by means of a single central controller has several disadvantages. The controller usually needs current knowledge about the entire system, necessitating communication links from every part of the system to the controller. These central control mechanisms scale badly, due to the rapid increase of processing and communication overheads with system size. Failure of the controller will often lead to failure of the complete system. There is the additional practical commercial requirement that centrally controlled systems may need to be owned by one single authority.

1.2 Appleby and Steward's Mobile Agents

Some form of distributed control seems natural for a system which essentially consists of a number of linked computational nodes. The case for implementing conventional distributed control by a number of static controllers was briefly reviewed and dismissed by Appleby and Steward (1994). As an alternative, they proposed the use of larger numbers of mobile software agents, claiming that this may produce benefits in robustness. Although their development of this idea included some experimental work, it was essentially a proof of concept study, and should not necessarily be taken to represent an advance on current methods of network routing.

In their mobile agents approach, there are two `species' of mobile agents: load management agents and parent agents. The lowest level of control is provided by the load management agents. Each such agent is launched from a particular node (its source node) and moves around the network, finding the current best routes from all nodes in the network to the source node using a clever adaptation of Dijkstra's shortest path algorithm (Dijkstra, 1959), and updating routing tables accordingly. The best route is defined as that which has the greatest minimum spare capacity of all possible routes; the aim is to distribute traffic evenly over the network to avoid high local loadings.

Parent agents provide the second level of control. They travel over the network, gathering information about which nodes are generating most traffic, and which nodes are more congested than others. Using heuristics, a parent agent can decide that network management at certain locations is needed to relieve congestion; it then travels to those locations to launch load agents. Mechanisms are proposed for regulating the numbers of load agents and parent agents to ensure that the number present is appropriate for the prevailing conditions, given that agents may crash, and that the requirements for agents may change as the network congestion comes and goes. Appleby and Steward simulated the effects on a network of rather limited traffic under two conditions: with fixed routing tables giving the shortest routes; and with routing tables subject to amendment by the mobile agents. The fixed routing produced severe local congestion, whereas the mobile agents successfully prevented congestion by spreading traffic more evenly across the network. However, Appleby and Steward present few details and only a single quantitative example.

Arguing from the standpoint of providing a programming discipline, Appleby and Steward proposed to achieve robustness with respect to 'the failure of an agent or the failure of a component in the distributed system' by following three precepts:

- there should be no direct inter-agent communication
- the agents should be present in reasonably large numbers
- the agents should be able to dynamically alter their task allocations and number

Their implementation rejected the ideas of distributed artificial intelligence, and instead drew on the principles of the Subsumption Architecture (Brooks, 1986) to support the idea of using relatively simple agents to achieve 'complex, intelligent behaviour...by exploiting the interaction of a simple control system with a complex environment'. However, their agents are essentially rooted in the methods of artificial intelligence, in that they are computationally based, using specialised and precise algorithms, maintaining elaborate records, and using complex sets of carefully crafted heuristics for decision making. We agree with their analysis, but propose the use of a rather different metaphor for the mobile agents: the behaviour of social insects such as ants.

1.3 The Abilities of Ants

Individual ants are in some ways very unsophisticated insects. They have a very limited memory and exhibit individual behaviour that appears to have a large random component. Acting as a collective, however, ants manage to perform a variety of complicated tasks with great reliability and consistency. Examples of collective behaviours that have been observed in several species of ants can be found in (Hölldobler & Wilson, 1994) and (Franks, 1989). These behaviours emerge from the interactions of large numbers of individual ants with one another and with their environment. In many cases, the principle of stigmergy (Grassé, 1959) is used. Stigmergy is a form of indirect communication through the environment. Like other insects, ants typically produce specific actions in response to specific local environmental stimuli, rather than as part of the execution of some central plan. If an ant's action changes the local environment in a way that affects one of these specific stimuli, this will influence the subsequent actions of ants at that location. Ants and other social insects have developed this principle to a high level, mainly by evolving actions which have no effect on the environment other than influencing the behaviour of passing ants; the mechanism of choice is the deposition of a variety of volatile chemical substances, pheromones, which have specific effects on behaviour. Stigmergy is a general method of control, and has recently been demonstrated in the domain of collective robotics (Beckers et al, 1994).

Social insect systems are impressive in many ways, but two aspects are of particular interest. First, they are outstandingly robust. They function smoothly even though the colony may be continuously growing, or may suffer a sudden traumatic reduction in numbers through accident, predation, or experimental manipulation, or may spontaneously split into two distinct colonies of half the size (Franks, 1989). They routinely cope with gross and minor disturbances of habitat, and with seasonal variations in food supply. Second, they are able to achieve an appropriate balance between the effort put into many parallel tasks, mainly by controlling the number of insects at he location at which the task is taking place. For example, if a number of breaches are made in a nest, all will soon be defended by soldiers clustering round them, and all will soon be under repair by teams of workers. Again, as the numbers of brood at different stages change with time, the numbers of workers caring for each stage will change appropriately.

These observations suggest that it may be possible to use very simple agents, with little or no memory or computational ability, to achieve an appropriate balance between a number of competing activities by interacting with the traces left in the environment by one another. A network supporting a variety of calls may be seen as a collection of competing activities. There is also a very obvious candidate for a suitable mechanism for routing: since the problem is to do with the routes taken between various locations, why not exploit some analogue of the main method used by ants - the laying and following of pheromone trails?

1.4 Trail laying

Depending on the species, ants may lay pheromone trails when travelling from the nest to food, or from food to the nest, or when travelling in either direction. They also follow these trails with a fidelity which is a function of the trail strength, among other variables. The strength of the trail laid by an ant may be modulated by internal state or by local circumstances. Since pheromones evaporate and diffuse away, the strength of the trail when it is encountered by another ant is a function of the original strength, and the time since the trail was laid. Most trails consist of several superimposed trails from many different ants, which may have been laid at different times; it is the composite trail strength which is sensed by the ants. Abstractions of these functional characteristics form the basis of the scheme for load balancing presented here.

1.5 Previous Work

The metaphor of trail laying by ants has previously been successfully applied to certain combinatorial optimisation problems such as the Traveling Salesman Problem and Job Shop Scheduling (Dorigo, Maniezzo & Colorni, 1996; Gambardella & Dorigo, 1995). These investigations were concerned with finding one good solution to a static problem. However, the problem of load balancing in telecommunication networks is essentially dynamic. The stochastic nature of calls, and the variations in call distributions, mean that the problem to be solved constantly changes with time, as different call combinations give rise to congestion in different areas of the network. It is essential to maintain network performance throughout the response of the load balancing system to a change in call distributions; we are therefore interested in the performance of the algorithm over a certain period of time, and not merely in the eventual performance of some fixed solution. The differences between our method and Dorigo's are largely determined by the need to cope with these dynamic aspects of the telecommunications routing problem.

2 Ant-like Mobile Agents

Consider a network with a certain amount of spare capacity at each node. Let A and B be two nodes in the network: we wish to exploit some analogy of trail laying to discover a route from A to B across the network which uses relatively little capacity (i.e. traverses few nodes) and which also avoids using capacity on heavily congested nodes. It will first be convenient to discuss the problem of simply finding short routes.

2.1 Finding shortest paths

Consider an ant-like mobile agent (an ant) released at A, which moves randomly from node to node until it arrives at B. If it moves at one node per time step, then the time it takes to reach B will reflect the length of its path. If a number of such ants are released at A, and follow different routes, then the length of each run will be reflected in the 'age' of each ant when it reaches B. A scheme for biasing the entries in call routing tables in favour of the routes followed by the youngest ants reaching B would therefore achieve the first objective. By analogy with trail laying, we would like each ant to leave some small influence on the routing table at each node it passes through along the way, and for these influences to combine to yield the desired effect. However, there is a problem: when an ant is on its way to B, it does not know how long it will take to reach B, and so does not know how good its route is, or even what its route will be, and so it cannot make any alteration to the routing tables to reflect the goodness of its route.

One way round this would be to allow the ant to reach B without leaving any influence, and then to allow it to retrace its steps, leaving an appropriate influence. However, this requires that the ant must both remember its path, and be able to invert it, or alternatively that it leaves temporary traces unique to itself which it can use to retrace its steps. A far more elegant and computationally undemanding scheme follows from the observation that at every intermediate node on the way from A to B, the ant's age reflects the length of the route from that node to A. It can therefore leave some influence at each node which reflects the time it would take to get to A from that node, via the inverse of the route taken by the ant. It is not practical to index the record of this influence at each node by some representation of the route, because this would require the ant to remember its route, and would require an enormous number of entries at each node, and a suitable encoding scheme. However, since the node can easily detect which link the ant arrived over, it requires no memory and little computation for the ant to make a contribution related to its age which is associated with 'all routes to A via that link'. At each node, the largest influence on the selection of the next node on the way to A would be deposited by the ant which had arrived there from A by the shortest route; this would apply at all nodes between node A and node B, and so by successively following these 'best' indicators at each node from B to A, the shortest route should be followed.

2.2 Getting down to detail

There are a number of problems with this naive scheme. The first is that we require new deposits to be combined with old deposits in some way which reflects a combination of monotonic increase and decay, rather than by implementing some max function. (Using a max function is very similar to the underlying idea of Dijkstra's algorithm.) However, any form of summation will suffer from the problem that there are a huge number of alternative routes from A to B, and that very few of them are short; this would lead to the influence from the short routes being overwhelmed by the effectively random influence from the longer routes. There are two obvious ways of counteracting this: weighting short paths very heavily compared to longer paths; and arranging for the ants to be somehow biased towards choosing shorter routes anyway, thereby increasing the relative proportion of ants arriving at B via the shorter routes. The first is trivial, requiring only the choice of a suitable function which is decreasing and positively accelerated with respect to age. The second is also potentially simple, because at each node the ant encounters on its way from A to B, there is information about the best routes from that node to B, left by ants launched from B.

From the point of view of economy, it is desirable that there should be only a single set of records at each node, which can be amended by ants, and which can be used both for guiding ants, and for routing calls. What form should such records take? For guiding ants, we can take an idea from real ants: when an ant is faced with two different pheromone trails, it chooses between them on a stochastic basis which is weighted by the strengths of the trails. The simplest form of this would be to represent the node information directly in terms of the probability of selection of each possible choice. For example, an ant at node G, with destination B, would find a table in the node indexed under 'B', with one entry for each of the neighbour nodes (say P, Q, R, and S). The four entries would sum to unity; the ant would select P, Q, R, or S by random selection, treating each entry as a probability. For routing calls, the highest of the cell entries could be selected deterministically.

A convenient form for representing the influence of an ant on the cell entry at a node is as the reciprocal of the age, plus a con-

stant. The following expression proved satisfactory:

$$\Delta p = \frac{0.08}{age} + 0.005$$

where Δp is the initial change to the cell. The cell entry thus should become $p = p_{old} + \Delta p$, where p_{old} is the original entry. However, the table must be normalised. Therefore the new cell entry becomes $p = (p_{old} + \Delta p)/(1 + \Delta p)$. The other entries become $p = p_{old}/(1 + \Delta p)$.

One possible problem with this simple approach is that a pheromone table may become frozen, with one entry almost unity, and the others vanishingly small; the simulated ants would then always make the same choice. Real ants do not appear to respond in this way, however strong the stimulation; there is always a chance that an individual ant will wander off the trail, apparently at random. This has potentially beneficial effects, in that it ensures that the whole environment is constantly being explored, although at a low level; however, it introduces an element of inefficiency, in that if nothing of use is discovered, the exploration energy has been wasted. A convenient way of preventing this freezing in the simulated system is to define a noise factor of f, such that at every time step an ant has probability f of choosing a purely random path, and probability (1-f) of choosing its path according to the pheromone tables on the nodes. The possibly beneficial effects of the addition of noise to ant-based algorithms were noted in (Deneubourg et. al., 1990): “Rather than simply tolerating a certain degree of error, it can even be desirable to deliberately add error where none or little exists.” Similar mechanisms were found useful by Sutton (1990) in the field of reinforcement learning.

2.3 Testing the ability to find short paths

Having developed this train of thought, we ran a simulation to check that this scheme would tend to produce the shortest paths on a typical network. We used the same model of the network as Appleby and Steward did (Figure 1).

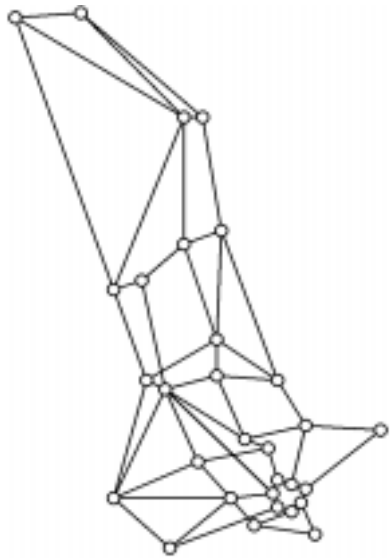

FIGURE 1. This network topology is the same as the interconnection structure of the Synchronous Digital Hierarchy network of British Telecom, and provides a realistic network topology

At each time step of the simulation, one ant was launched from each node; the destination of each ant was randomly selected. The pheromone tables were initialised with equal probabilities for each choice. No calls were placed on the network. After each time step we froze the pheromone tables, and ran a nested simulation in which a 'phantom ant' used the pheromone tables to travel once between all possible combinations of source and destination node, without affecting the pheromone tables. The total path length run by the phantom ant was divided by the square of the number of nodes to yield the average path length followed by an ant. The simulation was then restarted. Two simulations were run, one using ants with no noise (f=0) and one using ants with 5% noise (f=0.05). Figure 2 shows how for the ants with no noise the average path length declined over 1000 time steps from over 40 links to under 4; the minimum for this network is known to be 3.07 links (or 4.07 nodes). Since the phantom ant selected its route stochastically, this figure is bound to include some non-optimal choices, and so will be greater than the figure which would have been obtained from a deterministic choice mechanism. However, it demonstrates that the mechanism works. With 5% noise, the routes are very slightly longer, as would be expected. Of course, a smaller set of mobile agents using Dijkstra's algorithm would produce a guaranteed optimum result much more efficiently, but the point is that a good enough result can be produced by minimal agents using an analogue of trail laying.

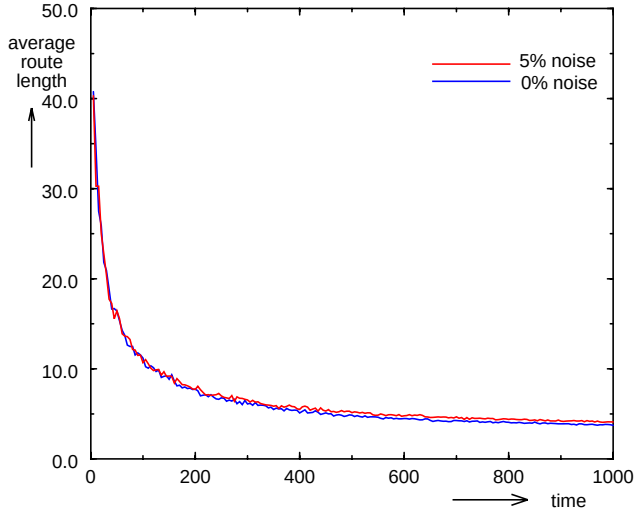

FIGURE 2. The average route length travelled by ants plotted against elapsed time for the network of Figure 1.

3 Dealing with Congestion

The next step is to modify the scheme to take congestion into account. Since the age of an ant is already used to modify the pheromone table entries, it seems sensible to allow the congestion encountered to affect the ant's record of its age.

3.1 Using congestion to affect age

There are two simple and related methods. The first is to add a quantity to the ant's age record at each node it encounters, with the quantity being some function of the degree of congestion of the node. This means that, at the end of its journey, the ant's age will have been increased by an amount reflecting to some extent the congestion found at every node, not just that found at the worst node as in Appleby and Steward. The second method is actually to

delay the ant by some amount. This has the obvious effect of increasing its age on arrival at subsequent nodes, just as in the first method. However, there is an additional and rather subtle effect: by preventing an ant from leaving a congested node, it prevents the ant from increasing the pheromone table entries pointing back to the node, which would have tended to increase traffic through the node. This breathing space will allow other ants arriving at the temporarily inaccessible node to reduce those pheromone table entries by the operation of the normalisation mechanism. The net effect will be a double reduction in the pheromone table entries pointing back to the congested node. This delay mechanism is also attractive from the commonsense point of view - it seems reasonable that it should take more time for the agent to pass through a congested node.

We require that the change produced by a given node should increase with increasing congestion. A suitable function is a negative exponential, with the exponent proportional to the degree of congestion. After some trial and error, the expression

$$delay = \lfloor 80 \cdot e^{-0.075 \cdot s} \rfloor$$

where s is the spare capacity of the node, was found to be suitable.

4 Modelling the Network

In order to assess the potential of the system for load management, and to compare it to other methods, it was necessary to develop a model network on which suitable distributions of calls could be placed. The SDH network was again used. Each node was given a capacity of 40 calls; links between nodes were assumed to have unlimited capacity.

4.1 Call distributions

An important idea in the area of networks which are not fully interconnected is that of the call distribution. In a real network, calls do not occur equally often between all possible pairs of nodes; at different times, and because of different external factors, certain pairs of nodes will generate much more traffic that other pairs, though of course the calls on those nodes may still begin at random times and last for randomly determined durations. If the busy nodes are far apart, they will consume more network capacity than if they are close. If they are in certain places in the network, they may unavoidably create certain bottlenecks leading to high congestion. A routing scheme suitable for one call distribution may be unsuitable for another. For these reasons, we decided to test the call routing systems not on a single call distribution (such as source and destination randomly selected with equal probability) but on a number of different call distributions.

Ten different call distributions were generated. Each was produced as follows: nodes were randomly assigned numbers between 0.01 and 0.07; the numbers across all 30 nodes were normalised to sum to 1; these numbers were taken to be the probabilities that a node would be the end point of a call generated at any instant under that distribution. From each call distribution, a call sequence lasting 15,000 time steps was generated as follows: at each time step, an average of one call is generated (Poisson distribution) with an average duration of 170 time steps (exponential distribution); the source and destination of each call are obtained by selecting nodes randomly with the probabilities determined as above. For the purposes of controlling the experimental design, each call sequence was numbered from 1 to 10, and split into two blocks 7,500 time steps long labelled A and B.

5 Modifying Appleby and Steward's Agents

We decided to evaluate the dynamic load balancing abilities of ants with no noise and with 5% noise. For comparison, we devised a static routing system consisting of the shortest routes, as determined by a backtracking algorithm. We also decided to evaluate the abilities of the mobile agents developed by Appleby and Steward. However, early tests of the mobile agents revealed some deficiencies, and some possibilities for improvements; the implementation was therefore modified as detailed below.

5.1 Eliminating circular routes

In (Appleby & Steward, 1994) load agents did not update the routing tables in the direction of their source node, but in the direction of an intermediate node, and from all nodes on the route between this node and the source node. In this way two load agents from different source nodes may at the same time do updates of routes to the same node. As these agents might have different data, because of constant network changes, we suspected that circular routes might occur in the network, and in early simulations repeating the work of Appleby and Steward we observed such circular routes. By making agents update routes in the direction of their source node, and by strictly limiting the number of load agents per source to one, we can avoid any possibility of circular routes. However, we lose the possibly beneficial effect that load agents also update routes to nodes in the network other than their own source node.

5.2 Eliminating unnecessarily long routes

We also investigated the use of a different criterion for `best route' because we had observed unnecessarily long routes in simulations where load agents maximise the minimum spare capacity. A call on such a long route occupies many nodes, and this additional demand on network resources may lead to congestion, causing subsequent calls to fail; other load agents may then respond to the congestion by amending the route to follow an even longer path. This can be counteracted by storing the total sum of squared utilisations of all nodes on the route from that node to the agent's source node, instead of the largest spare capacity of the route. (The node utilisation is the percentage of the node's capacity that is occupied by calls.) This introduces a bias towards shorter routes (routes with fewer nodes). Note that by squaring the utilisation, the relative influence of heavily utilised nodes is increased.

6 Experimental Design

Because we wished to make statistical performance comparisons using quantitative data, it was necessary to design the series of experiments quite carefully. The performance indicator chosen was the proportion of calls in a given period which could not be placed on the network; the proportion was used instead of the absolute number because the stochastic method of call generation produced different numbers of calls in each block. Early trials showed that there was enormous variability in the scores between blocks from different call distributions. The normal way of dealing with high variability is to take many samples; the variance of the sample mean is reduced by a factor of the square root of the number of samples. However, the simulation of telecommunications networks is computationally very expensive, and so this strategy cannot be used. The alternative is to take repeated measures of the same entities under the different experimental treatments. This was achieved by using the ten B sequences as the test sequences under all conditions, and comparing treatments by com-

paring the paired observations derived from each sequence under the two conditions of interest.

Data were taken under four different experimental conditions:

1. allowing each system to adapt to sequence A from given call distribution and testing it on sequence B from the same distribution
2. allowing each system to adapt on sequence A from a given call distribution, and testing it on sequence B from a different call distribution
3. as for (1), but disabling the dynamic load balancing for the test sequence
4. as for (2) but disabling the dynamic load balancing for the test sequence

For the ant simulations, initialisation was necessary to produce pheromone tables which did not contain any loops. This was done by allowing the network to run without calls for a period of 250 time steps for 0% noise and 500 time steps for 5% noise. Mobile agents were initialised by setting the routing tables to the same shortest path settings that were used for the no load balancing case.

7 Results

The experimental results, showing the percentage of call failures for each condition, are displayed in tables 1 to 4. Graphs 3 to 5 show the absolute numbers of call failures, measured in successive blocks of 500 time steps.

A number of planned comparisons were made among the call failure data using Student's t-test for related samples. The following findings reached the 1% significance level:

- All ant experiments gave better results than the corresponding experiments with the improved mobile agents
- All improved mobile experiments gave better results than the corresponding experiments with the original mobile agents
- All dynamic load balancing methods gave better results than no dynamic load balancing
- In the case of unchanged call probabilities, ants with no noise outperformed ants with 5% noise
- Ants with no noise perform better with unchanged call probabililties than with changed call probabilities
- Stopping launching load agents produces worse performance than continuing to launch them
- Stopping launching ants produces worse perforance.than continuing to launch them.

	Mean	**Standard dev.**
Fixed, shortest routes)	12.57%	2.16%
Original mobile agents	9.19%	0.78%
Improved mobile agents	4.22%	0.77%
Ants (0% noise)	1.79%	0.54%
Ants (5% noise)	1.99%	0.54%

TABLE 1. Results for unchanged call distributions

	Mean	**Standard dev.**
No improved mobile agents after 7500	6.43%	2.17%
No ants (0% noise) after 7500	2.11%	0.60%
No ants (5% noise) after 7500	2.48%	0.69%

TABLE 2. Results for unchanged call distributions, load balancing stopped

	Mean	**Standard dev.**
Fixed, shortest routes	12.53%	2.04%
Original mobile agents	9.24%	0.80%
Improved mobile agents	4.41%	0.85%
Ants (0% noise)	2.72%	1.24%
Ants (5% noise)	2.56%	1.05%

TABLE 3. Results for changed call distributions

	Mean	**Standard dev.**
No improved mobile agents after 7500	8.03%	2.88%
No ants (0% noise) after 7500	4.29%	2.06%
No ants (5% noise) after 7500	4.37%	2.27%

TABLE 4. Results for changed call distributions, load balancing stopped

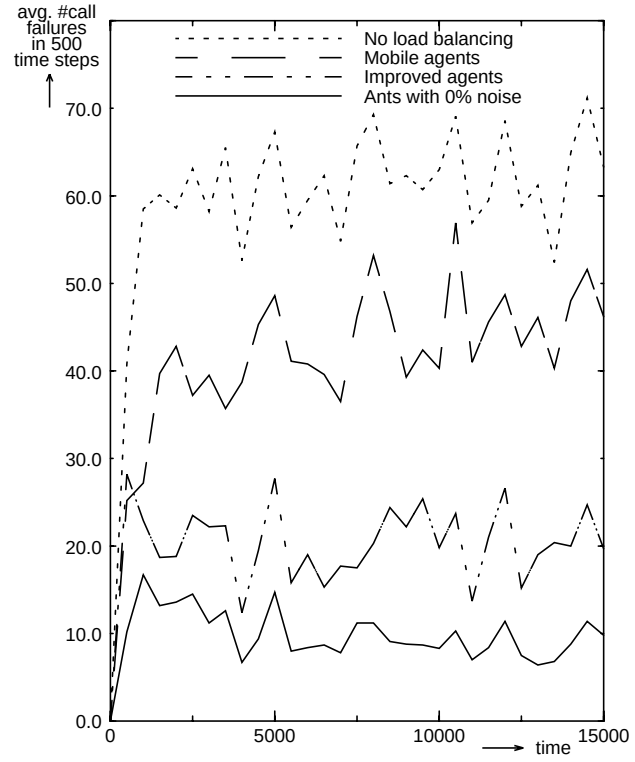

FIGURE 3. Temporal course of call failure rates for four load balancing techniques, with unchanging call probabilities.

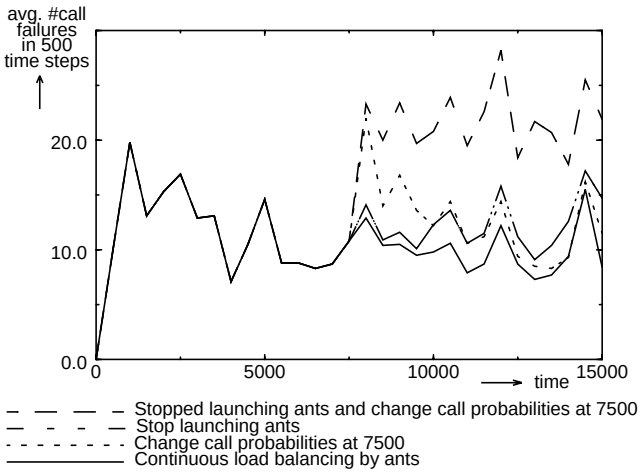

FIGURE 4. Temporal course of call failure rates for ants, with changed call probabilities.

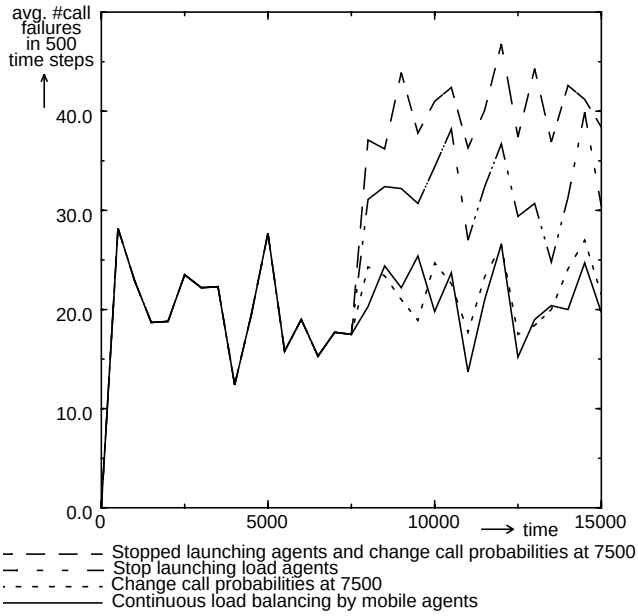

FIGURE 5. Temporal course of call failure rates for mobile agents, with changed call probabilities.

8 Discussion

The results clearly show that minimal mobile agents with behaviour modelled on the trail laying abilities of ants can perform useful load balancing, and that they can outperform mobile agents using sophisticated heuristics and algorithms. Both systems are shown to be able to cope with sudden changes in the distribution of call probabilities. It is easy to see that the ant based systems may also be robust with respect to agent failure, but this was not investigated.

One of the most interesting questions raised during the simulations was whether the ant-based and mobile agent systems were converging to a good static set of routing tables for each set of call statistics, or whether they were constantly adapting to changing situations. A good static set of routing tables would effectively combine information about the network topology and the call distribution statistics; routes would be sufficiently short, but would avoid the nodes likely to become congested with that topology and those particular call statistics. We think that three different forms of adaptation are possible:

- Adaptation to the network topology alone
- Adaptation to the call statistics within a given topology
- Adaptation to temporary situations produced by the randomness of the call patterns

8.1 Adaptation to network topology

Although adaptation to the distribution of network loads is a response to both the topology and the call probabilities, it is possible to get some insight into how well the system with an arbitrary set of call probabilities adapts to the topology alone. This insight can be obtained by inspecting the results of the experiments where dynamic load balancing is stopped at the same time as the call probabilities are changed. Any useful adaptation of the control system can then only be in relation to the topology, which is the only factor left unchanged.

For both the agents and the ants, the performance under these conditions is better than the experiments with fixed shortest path routing tables and no load balancing at all. Further one can clearly see that the ant based system performs better than the mobile agents (8.03% call failures for the improved agents versus 4.29% and 4.37% for the two ant experiments); this indicates that the ant based system is superior in adapting to the load distribution caused by the topology alone.

8.2 Adaptation to the call statistics within a given topology

To see how well a method performs when adapted to a combination of topology and call statistics, one can consider the results where the launching of agents or ants is suddenly stopped, but the call probabilities remain the same. This freezes the routing tables, and removes any contribution from short-term adaptations to local temporary situations. Here the ant based system again performs better than the mobile agents (6.43% call failures for the agents versus 2.11% and 2.48% for both ant systems). The results are also better than those discussed above, involving adaptation to the network topology alone. The size of the difference gives some indication of the influence of the call probabilities compared with the topology alone.

The graph of call failures after a change in call probabilities in Figure 4 is particularly revealing. Immediately after the change, failures are at a high level, but as the ant based system adapts to the new call probabilities, the failure rate declines until it is at the original adapted level. This shows the dynamics of adaptation to call probabilities very clearly.

8.3 Adaptation to temporary situations

The performance of ants and agents on adapting to temporary situations is indicated by the differences in performance under unchanging call probabilities of the conditions where load balancing is either continued or stopped. The situations where ants are launched after 7500 time steps perform better than those in which launching is stopped. Although the ant based system has adapted to call probabilities and to the topology, routes are still changed frequently in response to temporary situations. Because this results in improved performance, we can take this to indicate that ants are dynamically adapting the routes on the network to temporary situations as well. The same can be said about the agents, which seem in fact to be more sensitive to these temporary situations than the ants, as the difference in performance between the two agent experiments is relatively larger. Close observation of the network while running the simulation also confirmed our impression of useful reaction to temporary situations for both the ants and the agents.

9 Conclusions and Further Work

This work shows that ant based load balancing is a promising technique. Further work is now being undertaken to establish whether such systems cope well with the characteristics of real telecommunications networks (continuous growth, sudden failures in links and nodes), whether they can deal with some of the known technical problems in network regulation such as Braess' paradox, whether they are in fact robust with respect to agent failure, and whether they are better than some of the new techniques of dynamic routing recently adopted for use in real networks.

10 References

Appleby, S., and Steward, S. 1994. Mobile software agents for control in telecommunications networks. *BT Technology Journal*, Vol. 12, No.2.

Beckers, R.; Deneubourg, J.L.; and Goss, S. 1992. Trails and U-turns in the Selection of a Path by the Ant Lasius Niger. *J. theor. Biol.* 159, 397-415.

Beckers, R.; Deneubourg, J.L.; and Goss, S. 1993. Modulation of trail laying in the ant Lasius Niger and its role in the collective selection of a food source. *J. Ins. Behav.* (in press)

Beckers, R.; Holland, O.E.; and Deneubourg, J.L. 1994. From local actions to global tasks: Stigmergy and Collective Robotics. In R.A. Brooks, & P. Maes (Eds.) Artificial Life IV, Cambridge, MIT Press.

Deneubourg, J.L.; and Goss, S. 1989. Collective patterns and decision-making. *Ethology, Ecology & Evolution* 1, 295-311.

Deneubourg, J.L.; Goss, S.; Franks, N.; Sendova-Franks, A., Detrain C.; and Chrétien, L. 1990. The dynamics of collective sorting robot-like ants and ant-like robots. In J.-A. Meyer & S. Wilson (Eds.), From Animals to Animats: Proceedings of the first international conference on simulation of adaptive behavior. Cambridge, MIT Press.

Dijkstra, E.W. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik* vol. 1.

Dorigo, M.; Maniezzo, V.; and Colorni, A. 1996. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics* Part B, Vol. 26, No. 1, 1-13.

Franks, N.R. 1989. Army Ants: A Collective Intelligence. *American Scientist*, Volume 77, March-April.

Goss. S.; Beckers, R.; Deneubourg, J.L.; Aron, S.; and Pasteels, J.M. 1990. How trail-laying and trail following can solve foraging problems for ant colonies. R.N. Hughes (Ed.) NATO AS1 Series, Vol. G20 Behavioral mechanisms of food selection, Springer Verlag.

Grassé, P.P. 1959. La reconstruction du nid et les coordinations inter-individuelles chez Bellicositermes natalensis et Cubitermes sp. La theorie de la stigmergie: Essai d'interpretation des termites constructeurs. *Ins. Soc.*, 6, 41-83.

Gambardella, L.M.; and Dorigo, M. 1995. Ant-Q: A Reinforcement Learning approach to the traveling salesman problem. Proc. of ML-95, 12th Int. Conf. on Machine Learning, Morgan Kaufmann, 252-260.

Hölldobler, B.; and Wilson, E.O. 1994. Journey to the Ants. Belknap Press / Harvard University Press.

Schoonderwoerd, R. 1996. Collective Intelligence for Network Control. Ir.-thesis, Delft University of Technology, Faculty of Technical Informatics.

Sutton, R.S. 1990. Reinforcement Learning Architectures for Animats. In J.-A. Meyer & S. Wilson (Eds.), From Animals to Animats: Proceedings of the first international conference on simulation of adaptive behavior. Cambridge, MIT Press.

Wilson, E.O. 1975. Sociobiology, Belknap Press / Harvard University Press.