Ants for load balancing in telecommunications networks

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

[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@csd.uwe.ac.uk jlb@hplb.hpl.hp.com roth@kgs.twi.tudelft.nl

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 be lost. 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 ants move across the network between arbitrary pairs of nodes, selecting their path at each intermediate node according 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 are lost. The results of using the ant-based control (ABC) are compared with those achieved by using fixed shortest-path routes, and also by using an alternative algorithmically-based type of mobile agent previously proposed for use in network management. The ABC system is shown to drop fewer calls than the other methods, while exhibiting many attractive features of distributed control.

1 Introduction

The notion of complex collective behaviour emerging from the behaviour of many relatively simple units, and the interactions between them, is fundamental to the field of artificial life. The growing understanding of such systems offers the prospect of creating artificial systems which are controlled by such emergent collective behaviour; in particular, we believe that the exploitation of this concept might lead to completely new approaches for the management of distributed systems, such as load balancing in telecommunications networks.

What is load balancing? For economic and commercial reasons, such 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 might 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; in a large network, there are a great many possible routes for each such call. It is thus possible to relieve actual or potential local congestion by routing calls via parts of the network which have spare capacity. Load balancing is essentially the construction of call-routing schemes which successfully distribute the changing load over the system and minimise lost calls.

Controlling distributed systems like these by 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. Further, the nature of distributed systems like these is highly dynamic, complex and stochastic, and their behaviour can neither be predicted nor explained by reducing it to a single central controllable factor.

A good decentralized control mechanism will not have the problems mentioned above. The field of artificial life has given us inspiration for such a mechanism that will be completely distributed, and highly adaptive to changes in the network and traffic patterns. This solution makes use of the parallel processing capability already inherently present in the network in the form of network nodes. The distributed nature of such an approach may make the system very robust against failures of individual controlling entities.

Our approach is inspired by the work of biologists studying social insects, who have uncovered the mechanisms controlling the foraging behaviours of ants [Beckers 92], [Deneubourg 89], [Goss 90], [Franks 89]. The most important method is the laying and sensing of trails of pheromones - specialised chemical substances which are laid in amounts determined by local circumstances, and which by their local concentration subsequently directly influence an ant's choice of route.

A network simulation model is presented where the network was populated by artificial ants that make use of the same principle; at each node they encounter in their journeys, ants leave an amount of simulated pheromone which is a function of the congestion of the node, and of the distance the ant has travelled from its source node; and ants select the next node in their journey on the basis of the local pheromone distribution. The routing of calls is then based on these pheromone distributions.

We compare this method with another decentralised network control mechanism, which is based on previous work carried out by British Telecom [Appleby 94]. It makes use of mobile agents, computational processes moving from node to node to gather information and make decisions for rerouting.

Both approaches are compared to an approach which uses fixed, non-adaptive routing tables algorithmically optimised to yield the shortest paths.

2 A Network Simulation to investigate distributed control mechanisms

An application has been written to simulate traffic patterns on a model of a switch-based telecommunications network.

A telecommunications network is most naturally represented by an undirected graph. Each node in the graph corresponds to a switching station; the links between nodes correspond to communication channels. A given node will usually only be linked to a subset of other nodes; links are intrinsically bidirectional. The network model used is a graph of *n* nodes, each of which has several attributes:

- A node identifier.
- A routing table with *n* entries, one for each node in the network. Each entry tells us which node is the next node on the route to the destination node concerned.
- A capacity. This is the number of simultaneous calls that the node can handle.
- A probability of being the end node (either source or destination) of a call.
- A spare capacity. This is the percentage of the capacity that is still available on the node.

The links between the nodes are assumed to have infinite capacity, so that the node capacities will be the only bottlenecks in the network.

When fixed routing tables or BT agents (Section 5) are used, the routing tables are initialised so that the length of every route, i.e. the number of nodes on the route, is minimal. In this way the average utilisation of the nodes will be minimised, but this is not necessarily the ideal way to avoid congestion, as congestion has to do with how the traffic on the network is distributed.

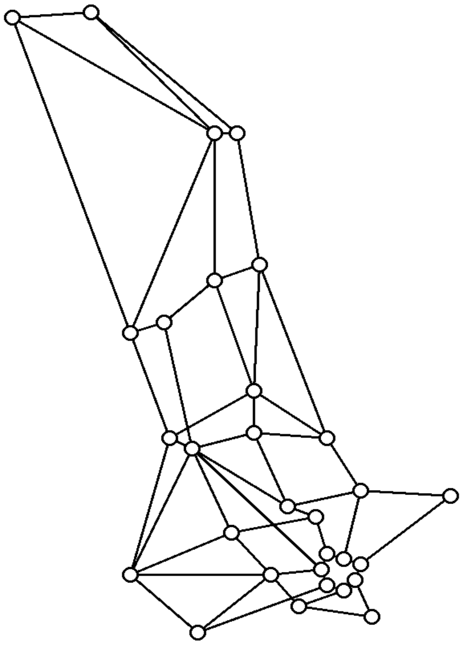
2.1 The simulation

Every time step of the simulation proceeds as follows. First, calls that have expired are removed, releasing capacity at the nodes. Then, one or more calls are generated by a traffic generator. These generated calls consist of a source node, a destination node and a duration, measured in time steps. When a call is generated, its route is determined by the current routing tables, and the call is placed on the network, reducing the spare capacity of each node on the route. The expected number of calls that is generated in one time step follows a Poisson distribution, and the duration of each call an exponential distribution. The source and destination nodes are randomly chosen according to their probability of being an end node; this is both convenient and reasonable.

2.2 Test details

The 30-node network topology of Figure 1 was chosen because this is a realistic interconnection structure of a possible switch-based network. It is also the same topology as was used in [Appleby 94], and is in fact the structure of a British Synchronous Digital Hierarchy (SDH) network. The interconnection structure is an irregular mesh that is interesting because it makes traffic management a complex and difficult task.

FIGURE 1. This network topology is the same as the interconnection structure of the SDH network of British Telecom and provides a realistic network topology.



The number of parameters that can vary in this network model is large. This implies that some arbitrary choices have to be made for testing:

- When a call is set up, each node in the model of Figure 1 has a certain probability of being an end point of the call. These probabilities are generated by selection from a suitable distribution at the start of every run, and lie between 0.01 and 0.07. After generation these probabilities are normalised to sum to 1.
- The capacity of each node is 40 calls, so that every call using the node increases the utilisation (or decreases the spare capacity) of the node by 2.5%.
- During every time step of the simulation, an average of 1 call is generated with an average duration of 170 time steps. This means that the average number of calls on the network will be 170, once the traffic pattern has built up.
- The average length of the shortest route between two nodes is 4.07. With fixed shortest-path routing tables, each of the 170 calls will use 2.5% of the capacity of on average 4.07 nodes. As there are 30 (number of nodes) times 40 = 1200 ‘capacity units’, the average utilisation of the nodes will be 170 x 4.07 / 1200 = 57.7%.

To facilitate seeing exactly what is going on in the network simulation, the traffic in the network is represented visually during run-time by changing the colour of the nodes to indicate their spare capacities. A large number of features to support visualisation and usability have been added to the program, such as displaying the changes with time of one particular route, running the simulation step-by-step, and inspecting every part of the network during the simulation. The software was written in Smalltalk, running in the VisualWorks™[1] environment on a Hewlett-Packard 9000 series 700 workstation.

3 Ants in nature

Individual ants are behaviourally 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. A few examples of collective behaviour that have been observed in several species of ants are [Hölldobler 94], [Franks 89]:

- regulating nest temperature within limits of 1ºC;
- forming bridges;
- massively raiding areas for food;
- building and protecting their nest;
- sorting brood and food items;
- cooperating in carrying large items;
- massive emigration of a colony;
- egg care;
- finding the shortest routes from the nest to a food source;
- preferentially exploiting the richest available food source.

These behaviours emerge from the interactions between large numbers of individual ants and their environment. In many cases, the principle of stigmergy 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. The environmental change may take either of two distinct forms. In the first, the physical characteristics may be changed as a result of carrying out some task-related action, such as digging a hole, or adding a ball of mud to a growing structure. The subsequent perception of the changed environment may cause the next ant to enlarge the hole, or deposit its ball of mud on top of the previous ball. In this type of stigmergy, the cumulative effects of these local task-related changes can guide the growth of a complex structure. This type of influence has been called sematectonic [Wilson 75]. In the second form, the environment is changed by depositing something which makes no direct contribution to the task, but is used solely to influence

1. VisualWorks is a trademark of ParcPlace Systems, Inc.

subsequent behaviour which is task related. This sign-based stigmergy has been highly developed by ants and other exclusively social insects, which use a variety of highly specific volatile hormones, or pheromones, to provide a sophisticated signalling system. Some of the above behaviours have been successfully simulated with computer models, using both sematectonic stigmergy [Theraulaz 94], [Theraulaz 95], and sign-based stigmergy [Stickland 92], and also on robots [Beckers 94], [Russell 95], [Deveza 94].

A type of sign-based stigmergy is used in our network model. It is based on the way ants find short routes from their nest to a food source, and also on the way they select between food sources of different value. The way ants organise these routes has inspired us to investigate a totally different approach for congestion avoidance in telecommunications networks.

3.1 Basic principle of bidirectional trail laying

Ants drop pheromones as they walk by stopping briefly and touching their gaster, which carries the pheromone secreting gland, on the ground. The strength of the trail they lay is a function of the rate at which they make deposits, and the amount per deposit. 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. The principles applied by ants in their search for food are best explained by an example as given in [Beckers 92]:

FIGURE 2. Ants have a decision to make

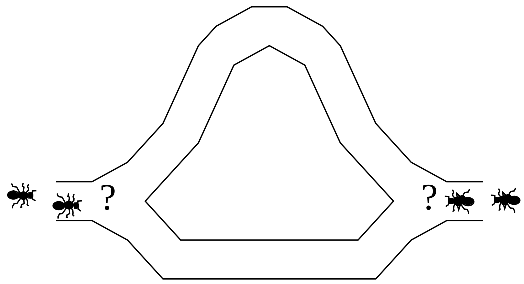
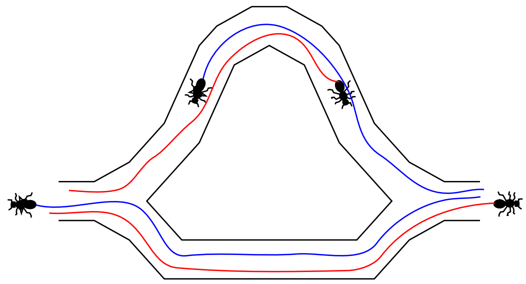


Figure 2 illustrates two possible routes from nest to food-source. Initially, an ant arriving at the T-crossing, makes a random choice. Approximately 50% of the ants will decide to go right and another 50% will choose to go left. Now suppose there are two ants leaving their nest, looking for food, and two ants are returning from the food source to their nest. Let the ants be of a type such as *Lasius Niger* which deposits pheromones when travelling both to and from the nest. After a while a situation occurs like that in Figure 3. The ants that chose the shorter branch have arrived at their destination, while the ones that chose the longer branch are still on their way. Ants initially select their way with a 0.5 probability for both branches, as there is no pheromone on the paths yet. If there is pheromone present, there is a high probability of an ant choosing the path with the higher pheromone concentration, i.e. the path where more ants have travelled recently. If at the moment of the situation in Figure 3 other ants arrive and have to choose between the two paths, they are more likely to choose the shorter path, because that is where the concentration of pheromone is higher. This means that the amount of pheromone on the shorter path is more likely to be reinforced again. In this way, a strong phenomone trail will arise on the shorter path, and so the path will be selected by an increasing proportion of ants As fewer ants choose the longer path, and the existing pheromone slowly evaporates, the trail on the longer path will weaken and eventually disappear.

FIGURE 3. Situation several moments later



Although this ‘learning technique’ is self-organizing, ants have to cope with a phenomenon that looks very much like overtraining in reinforcement learning techniques. There are two main issues: the blocking problem and the shortcut problem [Sutton 90]. The blocking problem occurs when a route previously found by the ants is no longer available. It can then take a relatively long time for the ants to find a new route. The shortcut problem occurs when a new, shorter route suddenly becomes available. In this case the new route will not easily be found, because the old trails are so strong that almost all the ants choose them.

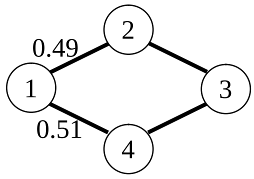
4 Ant-Based Control (ABC) for network management

How could this trail laying and following behaviour be applied to something like a telecommunications network? And can we overcome the blocking problem and the shortcut problem? This section describes how we implemented an artificial ant population on the network model. Further details may be found in [Schoonderwoerd 96].

4.1 Pheromone tables

We replaced the routing tables by tables of probabilities, which we will call ‘pheromone tables’, as the pheromones are represented by these probabilities. Every node has a pheromone table for every possible destination in the network, and each table has an entry for every neighbour. For example, a node with four neighbours in a 30-node network has 29 pheromone tables with four entries each. One could say that an *n*-node network has *n* different kinds of pheromones. The entries of the tables are probabilities which influence the way ants choose to go to their destination node. Figure 4 shows us a possible network configuration and a pheromone table. For example, ants that go from node 1 to node 3 have a 0.49 probability of choosing node 2 as their next node, and 0.51 of choosing node 4. 'Pheromone laying' is represented by 'updating probabilities'.

FIGURE 4. Using ants for network management



Example pheromone tables
for node 1:

goto:	2	4
2	0.95	0.05
3	0.49	0.51
4	0.05	0.95

Every time step during the simulation, on every node of the network ants can be launched with a random destination node. They move from node to node, selecting the next node to move to according to the probabilities in the pheromone tables for their destination node. Arriving at a node, they update the probabilities of that node's pheromone table for their source node i.e. ants lay the kind of pheromone of the node they were launched from. They alter the probabilities in favour of their previous node. When ants have reached their destination, they die.

In the situation in Figure 4, suppose an ant launched at node 1 has just travelled from node 2 to node 3. The pheromone table at node 3 for node 1 might be 0.5 to go to node 2 and 0.5 for node 4. The ant updates the relevant cell entries by the method described below; the probabilities may now be equal to 0.545 for a subsequent ant to go from node 3 to node 2 if node 1 is its destination, and 0.455 to go via node 4. In this way, ants moving away from their source node can only directly affect those ants for which it is the destination node. This is unlike the trails of bidirectional trail laying ants, in which a trail laid in one direction can directly affect ants travelling in either direction. However, the ants which can be directly influenced by an ant travelling from a source node S to a destination node D will include those travelling from D to S; these are the very ants which could be expected to have most influence on ants travelling from S to D, and so ants travelling from S to D may have a strong influence on ants subsequently travelling that route via their effect on the ants travelling on the opposite route. The system may thus achieve similar effects to the biological bidirectional trail layers, but through an indirect form of interaction.

This way of directly updating probabilities differs from the way ants lay pheromones, but is probably functionally equivalent. We feel that using probabilities instead of absolute pheromone quantities helps us to understand the behaviour of the ants better. The tables we use give the probabilities of alternative choices between paths directly, whereas the pheromones of real ants are basically a code that is effectively converted into probabilities by the ant's nervous system.

The method used to update the probabilities is quite simple: when an ant arrives at a node, the entry in the pheromone table corresponding to the node from which the ant has just come is increased according to the formula:

$$p = \frac{p_{old} + \Delta p}{1 + \Delta p}$$

Here p is the new probability and Δp is the probability (or pheromone) increase. The other entries in the table are decreased according to:

$$p = \frac{p_{old}}{1 + \Delta p}$$

Since the new values sum to 1, they can again be interpreted as probabilities. Note that a probability can be reduced only by the operation of the normalisation following the increase in another cell in the table; since the reduction is achieved by multiplying by a factor less than one, the probability can approach zero if the other cell or cells are increased many times, but will never reach it. For a given value of Δp the absolute and relative increase in probability is much greater for initially small probabilities than for those which are larger. This has the effect of weighting information from ants coming from nodes which are not on the currently preferred route, a feature which may assist in the rapid solution of the shortcut problem.

4.2 Ageing and delaying ants

A primary requirement of this work was to find some simple methods of encouraging the ants to find routes which are relatively short, yet which avoid nodes which are heavily congested. Two methods are used. The first is to make Δp, the value used to change the pheromone tables, reduce progressively with the age of the ant. When the ant moves at one node per time step, the age of the ant corresponds to the path length it has traced; this biases the system to respond more strongly to those ants which have moved along shorter trails. The second method, which depends on the first, is to delay ants at congested nodes to a degree which increases with the degree of congestion. This delay has two complementary effects:

- it temporarily reduces the flow rate of ants from the congested node to its neighbours, thereby preventing those ants from affecting the pheromone tables which are routing ants to the congested node, and allowing the probabilities for alternative choices to increase rapidly.
- since the ants are older than they otherwise would have been when they finally reach the neighbouring nodes, they have less effect on the pheromone tables.

It is of course possible to achieve the second effect alone by increasing the parameter representing the age of the ant without actually delaying the ant; this essentially reduces the effect on pheromone tables of an ant which has passed through a congested node. This has a biological parallel: in some species of ants, those returning from a richer food source tend to drop more pheromone than those from a poor source [Beckers 93]. In the case of our network simulation, however, the combination of delay and age-related penalty seems to be particularly effective. An added advantage of this formulation is that the manipulation of the parameter relating delay to the degree of congestion can be used to control the relative weighting which the system gives to preferring the shortest route (which maximises spare capacity), as against preferring the least congested route.

4.3 How calls are routed

Having explained how ants ‘choose’ their routes through the network, let us consider the calls. Calls operate independently of the ants. To determine the route of a call, the largest probability in the pheromone table for the desired destination is looked up. The neighbour node corresponding to this probability will be the next node on the route to the destination.

In this way, calls and ants dynamically interact with each other. Calls influence the load on nodes, by which the ants will be influenced. Ants influence the routes, which in their turn influence the calls.

4.4 Initialisation

A network initialised with random or uniform entries in the pheromone tables will not initially contain any useful information about complete and consistent (i.e. non-circular) call routes, let alone good routes. It therefore makes little sense to examine network performance during this phase. However, even in the absence of calls on the network, the ants should bias towards shortest paths. When this process has stabilised, calls can safely be put on the network; subsequent adaptation will then be influenced by any congestion caused by calls. All ant networks were therefore initialised with equal probabilities for neighbour nodes in each pheromone table, and allowed to run for a fixed period before calls were applied.

4.5 Noise

So far, we have not considered the blocking problem and the shortcut problem. We need to avoid ‘freezing’ of the routes in situations that remain static for a long time and then suddenly change. One way of doing this is by adding an exploration probability, or noise, to the random walk of the ants; this will ensure that even apparently useless routes are used occasionally, so that at least some level of knowledge about them is present in the system to give a head start when a route is blocked (e.g. by extreme node congestion or failure). A convenient implementation is to arrange that a noise factor of f means that for 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 beneficial effects of the addition of noise to ant-based algorithms were noted in [Deneubourg 90]: ‘Rather than simply tolerating a certain degree of error, it can even be desirable to deliberately add error where none or little exist.’

4.6 General framework for ant-based control systems

The basic principles for ant-based control (ABC) systems can be characterised as follows:

- Ants are regularly launched with random destinations on every part of the system.
- Ants walk randomly according to probabilities in pheromone tables for their particular destination.
- Ants update the probabilities in the pheromone tables for the parts of the system they were launched from. They increase this probability in favour of the part they just came from.
- The increase of these probabilities is a decreasing function of the age of the ant and of the probability already there. It could also be a function of penalties the ant has gathered on its way.
- The ants get delayed on parts of the system that are congested.
- To avoid overtraining through freezing of pheromone trails, some noise can be added to the behaviour of the ants.

4.7 Parameters

There is a large number of parameters to tune for this system - choices are based on experience with a variety of previous simulations:

- The speed of the ants is one node per time step (unless they are delayed on a particular node).
- We chose to let every node launch an ant with a random destination on every time step of the simulation.

- The probabilities are updated as explained in Section 4.1, and according to the following formula, where *age* stands for the number of time steps that passed since the launch of the ant:

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

- The *delay* in time steps that is given to the ant is dependent on the spare capacity *s* of the node:

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

- The initialisation period, that is the period during which the ants initialise the routes on the network without traffic, is between 250 (no noise) and 500 (4% noise) time steps.

Before we give some results of our simulations, we will describe briefly another distributed network management approach based on work by researchers from British Telecom. We compared the ABC method with this method and with a fixed routing scheme.

5 Mobile Software Agents

A different approach for a distributed control mechanism is provided by the mobile agents developed by researchers from British Telecom in [Appleby 94]. We implemented a modified version of their scheme on our network simulation model. Further details may be found in [Schoonderwoerd 96].

In this mobile agents approach, there are two ‘species’ of agents: load management agents and parent agents. The lowest level of control is provided by the load management agent. Each such agent is launched from a particular node, and then searches algorithmically for better routes from nodes in the network to the node where it was launched. Parent agents provide the second level of control. According to heuristics and information gathered on the network, a parent agent can decide that network management is needed to relieve certain locations, and therefore launches load agents at those locations.

5.1 The Load management agent

A load agent is launched on a particular node and optimises the routes from all other nodes to that source node. It does this by visiting every node in the network, recording the current spare capacity, and amending the routing tables. Load agents find the route with a minimum bottleneck i.e. they maximise the minimum spare capacity on the route.

The algorithm that is used to find these routes is a version of Dijkstra's shortest path algorithm [Dijkstra 59]. As a criterion for ‘shortest path’ the minimum spare capacity on the route is used.

5.1.1 Differences from the BT approach

In our implementation, there is one major difference to the approach in the BT work: We changed the direction in which routing tables were updated. In [Appleby 94] load agents did not update the routing tables in the direction of their source node, but in the direction of the newly visited nodes, and from all nodes on the route between this node and the source node. In this way two load agents may at the same time do updates of routes to a certain node. As these agents might have different data, because of constant network changes, we suspected that circular routes might occur in the network. In early simulations we observed such circular routes [Schoonderwoerd 96]. By changing the direction in which updating occurs we avoided this problem.

5.2 The Parent Agent

The next level of control in the network is provided by parent agents. They travel around the network and launch load agents where network management is needed. The decision to launch a load agent is made on the basis of information gathered, and a set of heuristic rules.

The parent agent travels randomly around the network to gather information about the different nodes. At each node the agent visits, it records the following data fields:

- The traffic destination rate. This is the number of calls that have the node as their destination
- The utilisation of the node, which is the percentage of the node's capacity that is used.
- The destination-rate history, which is the average destination rate of the last *d* visits to the node.

Further it records the following global information, when stepping around the network:

- The utilisation history, which is the average of node utilisations of the last *m* visited nodes.
- The number of nodes it has visited so far.
- A destination rate ranking table. This table contains a ranking of nodes according to their destination rate history.

When the agent encounters a node with a higher utilisation value than the agent's utilisation history, it assumes that traffic management is needed. The ranking table tells the agent which node is most used as a destination node of calls. The traffic to this node is likely to be the cause of some network overload. The agent then goes to this node and launches a load agent, unless the node has already got a load agent working for it. After reaching this node and launching a load agent, the parent agent continues its cycle of gathering information, and detecting where new traffic management is needed. Note further that our parent agents look at the destination rate, whereas the original BT agents looked at the sourcing rate of nodes. This is due to the difference of direction in which our load agents update the routing tables.

In real networks, the parent agent could also be programmed to solve problems like crashed load agents. This possibility is not modelled in our simulation.

5.3 Parameters

As with the ants, the space of possible parameter settings is huge: The values used in the simulations reported here were those found to be best according to our experience with previous experiments:

- The speed of the agents is basically the same as the speed of the ants: Every time step of the simulation an agent performs its task on its current node and moves to the next node.
- The length of the parent agent's global utilisation history. The parent agent takes the last 4 visits to each node into account to calculate the destination rate history.
- The global utilisation history is 60; the last 60 visits count in the calculation of the average utilisation.
- The destination ranking table has size 15. This means that if this table is smaller than 15, the parent agent will gather more information around the network
- The number of parent agents to test with is 2.

6 Results of the simulations

Every simulation on the network model of Figure 1 we present here has been performed 10 times, every time with the same call probabilities for every node, but with different call patterns, during time steps 10000 to 16000 (i.e. after the system has definitely stabilised away from any effects of initial conditions). The experiments have been performed for: a fixed routing scheme without load balancing; the BT agents; BT load agents being launched on every node as soon as a previous load agent finishes (so no parent agents are present); Ants without noise; Ants with noise; and ants that were stopped from being launched after time step 10000. The mean percentages of calls that were dropped are (standard deviation between brackets):

- Without load balancing: 13.1% (0.5%)
- BT agents: 8.4% (0.7%)
- BT load agents: 8.8% (0.6%)
- Ants without noise: 1.8% (0.5%)
- Ants with 4% noise: 1.7% (0.5%)
- No ants after 10000: 1.5% (0.3%)

We ran another series of experiments with the same parameters, but now at time step 10500 all call probabilities suddenly change (according to the same distribution as they had before). The new call probabilities are the same for all experiments:

- Without load balancing: 11.0% (0.8%)
- BT agents: 9.3% (1.1%)
- BT load agents: 9.0% (0.6%)
- Ants without noise: 2.9% (0.9%)
- Ants with noise: 2.8% (0.5%)
- No ants after 10000: 4.4% (0.4%)

We performed the Student t-test (2-tailed) for independent samples on these data. The following differences were found at the 0.01 significance level.

For the unchanging call probabilities: The three experiments with ants gave significantly better results than the two experiments with agents. The ant experiments were not significantly different from each other, nor were the experiments with the BT agents. All these experiments were significantly better than the experiment without load balancing.

For the tests with changing call probabilities the same conclusions can be drawn, except that during the experiments where no ants were launched after time step 10000, significantly more calls were dropped than in the other ant conditions, but the results of these experiments remained better than all experiments with the BT agents.

6.1 Static solution or dynamic adaptation

One of the most interesting questions raised during the simulation on the 30-node network was whether the system is converging to a good static set of routing tables that was ‘learned’ from the statistics of the system, or was constantly adapting to changing situations. A good static set of routing tables would 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 those particular call statistics.

The first series of experiments, where the call probabilities of the nodes remained the same, shows no significant difference between simulations where there were still ants on the system and those where we stopped launching ants. Therefore, we conclude that the ABC system converges to a good static solution.

In the second series of experiments we changed the call statistics, but the topology remained the same. We see here that the ants actively adapt to this new situation. The situation where we stopped launching ants now performs worse than the ants, but still a lot better than the BT agents and fixed routing tables. In this case the ABC system had adapted to the network topology and the previous set of call statistics.

The ant system is thus capable of adapting the system to both the information implicit in the network topology, and the congestion statistics arising from the different call probabilities.

Close observation of the network while running shows that the ants are also actively changing routes according to temporary congestion situations. This however appears to be of no clear quantitave benefit, once the network topology and statistics are adapted to. We suspect most benefit from dynamic adaptation is obtained when the best static solution is not yet reached, for example just after the change of the call patterns. As one might assume that call statistics regularly change, the ability to dynamically adapt remains important.

The balance between dynamic adaptation to temporary situations and finding good static routes depends on the parameters used for the ants. An exponentially decreasing delay function that starts relatively high (for example 80 time steps in our simulations), apparently increases the abilities of the network to dynamically adapt to changes in call patterns and the consequent changes in the distribution of congestion.

7 Ants versus Mobile software agents

Ants have a number of advantages over mobile agents, as well as some disadvantages.

Bandwidth consumption. An ant hardly requires any bandwidth on the network: It takes with it its age, its source id and its destination id, together with the fact 'I am an ant'. This is all. The BT agents take with them a number of relatively large tables and therefore require much more bandwidth than ants. A property of the mobile agents from BT is that more load agents tend to get launched as the load on the network increases, to do re-routing. This might be just the moment when you do not want any more agents on the network, as it is already congested with calls. In contrast, congestion-dependent delay of the ants temporarily reduces ant traffic at congested locations. Finally, at least in the situations we have examined so far, the numbers of BT agents used is of the order of tens, whereas ants are used in hundreds.

Limits to number of agents. In principle, there is no limit to the number of ants that can be used in a network, because ants do not interfere with each other. The number of load agents on a network is much more limited. Once a load agent is launched on a particular node, another one can not be launched until the first agent finishes its job, because otherwise they would interfere with each other.

Crashing of agents. Malfunctioning in the system might cause a mobile process such as an ant or agent to crash. If an ant crashes, this will not have a significant effect on the performance of the algorithm at all. However, if a load agent or parent agent crashes, this has to be detected, and special measures have to be taken to restore the damage.

Computational issues. Ants are likely to require more computation on the nodes of the network than the mobile agents, due to the extensive use of random generators. Further, with ant-based control, nodes need to allocate more space for their pheromone tables than is needed when normal routing tables are used. However, these issues do not affect bandwidth or switching capacity, which is our main concern.

Bidirectional routes. During the simulations of the ant controlled system, the route from one node to another tends most of the time to be the same that in the opposite direction. This is probably due to our mechanism of trail laying, where ants from complementary source and destination nodes mutually reinforce one anothers' trails. The BT agents do not have this property. At first sight, this property might be disadvantageous for good load balancing, but we think it makes no significant difference in the 30-node network.

Circular routes. In principle, ABC systems have the potential to yield circular routes. However, this situation was not observed, except when the noise was extremely high, or the initialisation period much too short. The BT agents as implemented here are guaranteed not to result in circular routes.

8 Conclusions and future work

We implemented a completely decentralized adaptive control system for telecommunications which made use of the emergent collective behaviour of artificial ants.

The principles of the algorithm are simple and general. We believe that the general framework for ant-based solutions presented here is capable of solving load balancing problems in large networks both circuit switched and packet switched. We do not yet know exactly how the statistical and topological properties of the network influence the ideal parameter settings. But as shown here, even tuning parameters by hand can lead to a well balanced system.

The balance obtained is a good example of emergent organisation. The individual ants are remarkably simple, but the resulting structures enable very efficient use of a resource like a telecommunications network.

We believe that ABC systems can be used to solve a large variety of optimisation problems, eg:

- distributing loads of interconnected processors on parallel machines and managing inter-processor communication for complex programs;
- material flow in production environments;
- optimal routing on integrated circuit-boards;
- organising public transport schemes.

We further think that ant-based simulations might help us in understanding a large variety of phenomena. An example could be the microeconomic analysis of the buying behaviour of people. People follow 'trails' that are laid by other people (I want what you want), or commercials.

Much investigation of the basic principles of the ant algorithm remains to be done. So far, our experiments have not yet enabled us to make statements that are sufficiently supported by statistics about the influence of the number of ants used in the simulations. We also do not know exactly how variations in the ants' influence on the pheromones affect the system, or about the effects of the size of the delays imposed on the ants. Most choices in this work are based on a relatively small number of experiments. It would also be useful to investigate the performance of the algorithm on extremely large or very small networks; the large networks will tell us about scaling, and the small neworks might assist our understanding of the basic processes which make the algorithm work. Another intriguing possibility is to use 'probabilistic routing' of calls. In that case, routes of calls, or perhaps a proportion of calls, would not be chosen according to the largest probabilities in the pheromone tables, but randomly according to these probabilities. A mechanism that is assumed not to be used by natural ants, but could be useful here, is laying 'anti-pheromone'. One could let ants directly decrease probabilities in the pheromone tables in particular circumstances, rather than increase them.

The pheromone tables do not only represent the best routes, but also contain information about the relative merits of alternative possibilities if something goes wrong. Our simulations have so far been confined to examining the use of this information in dealing with node congestion; sudden node (or link) failure and restoration also needs to be simulated to examine the abilities of the ants to deal with these contingencies. The ability to cope with the insertion of new nodes and links during network extension is also a topic of interest.

Extending ant-like algorithms to situations like telecoms networks which are not found in nature will also increase our understanding of the abstract and general abilities of such algorithms over and above those applications found in nature. We hope that this increase of knowledge will in turn assist and inform biologists studying social insects.

References

[Appleby 94] S. Appleby, S. Steward. Mobile software agents for control in telecommunications Networks. *BT Technology Journal, Vol. 12, No.2*. 1994.

[Beckers 92] R. Beckers, J.L. Deneubourg & S. Goss. Trails and U-turns in the Selection of a Path by the Ant *Lasius Niger. J. theor. Biol. (1992) 159, 397-415*.

[Beckers 93] R. Beckers, J.L. Deneubourg & S. Goss) Modulation of trail laying in the ant Lasius Niger and its role in the collective selection of a food source. *J. Ins. Behav. 1993 (in press)*

[Beckers 94] R. Beckers, O.E. Holland, and J.L. Deneubourg. From local actions to global tasks: Stigmergy and Collective Robotics. *Artificial Life IV. Brooks, R.A. and Maes, P. (ed.), MIT Press 1994. p. 181-189.*

[Deneubourg 89] J.L. Deneubourg and S.Goss. Collective patterns and decision-making. *Ethology, Ecology & Evolution 1: 295-311. 1989.*

[Deneubourg 90] J.L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain and L. Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. F*rom Animals to Animats. p.362. MIT Press 1990.*

[Deveza 94] R. Deveza, D.Thiel, A.Russell and A. Mackay-Sim. Odor sensing for robot guidance. *Int. J. Robotics Research, vol 13, no 3, 232-239, 1994.*

[Dijkstra 59] E.W. Dijkstra, A Note on Two Problems in Connexion with Graphs *In Numerische Mathematik vol. 1. 1959.*

[Franks 89] N.R. Franks. Army Ants: A Collective Intelligence. *American Scientist, Volume 77, March-April. 1989*

[Goss 90] S. Goss, R. Beckers, J.L. Deneubourg, S.Aron & J.M. Pasteels. How trail-laying and trail following can solve foraging problems for ant colonies. *NATO AS1 Series, Vol. G20 Behavioral mechanisms of food selection, Ed. R.N. Hughes, Springer Verlag 1990.*

[Hölldobler 94] Bert Hölldobler & Edward O. Wilson. Journey to the Ants. *Bellknap Press / Harvard University Press 1994.*

[Russell 95] R.A. Russell. Laying and sensing odor markings as a strategy for assisting mobile robot navigation tasks. *IEEE Robotics and Automotion Magazine, September 1995, 3-9.*

[Schoonderwoerd 96] Ruud Schoonderwoerd. Collective Intelligence for Network control. *Unpublished Ir.-thesis, Delft University of Technolgy, Technical Informatics, June 1996*.

[Stickland 92] T.R. Stickland, C.M.N. Tofts, and N.R. Franks. A path choice algorithm for ants. *Naturwissenschaften 79, 567-572. 1992.*

[Sutton 90] Richard S. Sutton. Reinforcement Learning Architectures for Animats. *From Animals to Animats. p. 288 - 296. The MIT Press 1990.*

[Theraulaz 95] G. Theraulaz & E. Bonabeau. Coordination in distributed building. *Science, 269, 686-688. 1995*

[Wilson 75] E.O. Wilson. Sociobiology, *Belknap Press / Harvard University Press 1975.*