Mobile Agents in Telecommunications Networks – A Simulative Approach to Load Balancing

Steffen Lipperts
Department of Computer Science (i4), University of Technology Aachen
Aachen, 52056, Germany

And

Birgit Kreller
Corporate Technology Department, Siemens AG
Munich, 81730, Germany

ABSTRACT

Networks today are growing continuously complex, with new kinds of services being included and heterogeneous networks interworking as a whole. Telecommunications networks in particular have become truly global networks, consisting of a variety of national and regional networks, both wired and wireless. Consequently, the management of telecommunications networks is becoming an increasingly complex task, as size and complexity constitute critical requirements that have to be met. Decentralized approaches to network management are currently being discussed, as is has become evident that central solutions cannot cope with scalability issues. Mobile agent technology in particular is being examined as a new distributed system and network paradigm.

One vital issue in telecommunications networks management is load balancing, as it allows to efficiently use the network to capacity and avoid overload situations. In this paper, we will examine swarming intelligence of mobile agents as a basis for the development of a decentralized load balancing mechanism in telecommunications networks. Various strategies for swarming intelligence will be evaluated and compared to conventional approaches with a simulative approach.

Keywords: Mobile Agents, Simulation, Load Balancing, Telecommunications Networks, Swarming Intelligence.

1. INTRODUCTION

Telecommunications networks today are volatile communication networks which consist of heterogeneous, very often incompatible, multi-vendor environments. Circumstances such as these cause the management of telecommunications networks to be complex and to contain operator-intensive tasks that need considerable human involvement. Legacy network management systems [1,2], however, follow a centralized approach which causes a number of problems. The management information to be processed threatens to excess the capabilities of the human managers. Moreover, the management solutions are insensitive to the rapidly changing network conditions and in addition cannot efficiently cope with the growing scale of the networks. In order to avoid the information overload, decentralized approaches to network management are currently being examined [3,4], with mobile agent technology [5,6] playing a crucial role in many of these approaches [7,8].

This paper focuses on a mobile agent based approach to load balancing in telecommunications networks. Load balancing aims at evenly distributing the load over a network, thus leaving no routers idle and preventing overloads for others. If no load balancing mechanisms are applied, network congestions can occur even if many of the network's nodes are not used to capacity at all. Centralized approaches to load balancing, which will be discussed in more detail in section 4, suffer from the same problems as other centralized management solutions. Therefore, in this paper we present a distributed approach which is motivated by biological phenomena.

In nature, several examples can be observed where an intelligent and efficient behavior of an overall system results from the interworking of autonomous individuals obeying simple rules, e.g. insect swarms, fish schools, and bird flocks. This behavior is being examined by Artificial Life [9], a research area which forms part of the Artificial Intelligence (AI) work. Artificial Life tries to determine in which way the simple rules and behavioral patterns result in an overall system execution. For this, the autonomous individual instances and the rules directing their local interaction and goal-directed behavior are being examined. It has been found out that there is no behavioral control on a global level, which is called locality of the system. It is rather a property called emergence which causes the complex, dynamic, and structured behavior on system level. Emergence is the crucial property in Artificial Life and denotes the occurrence of a system property drawing from the interaction of individual components, without having been specified explicitly or being directly deductible [10]. The term self-organization in this context describes the emergence of an improved system structure, e.g. with regard to stability or fault-tolerance. Apart from the autonomy of the components, their locality, and emergence, there are a number of additional characteristics of such systems. Since all of the components act autonomously and locally, they have a high degree of parallelism. This parallel dynamism differs from the predominantly sequential mode of execution which is given in traditional computer and network architectures. Transferring results of Artificial Life to computers and networks therefore introduces a new problem solving approach which will be discussed in detail in the following.

Another property of Artificial Life models is called temporal invariance of the components, i.e. these components have a predefined lifecycle. They are created and then remain unchanged, while able to multiply, and eventually are terminated. The latter can be triggered by the instance itself or by another component. This lifecycle is often extended by mobility, i.e. the components are able to move through the system autonomously. Together with the given lifecycle, especially the ability to multiply, the population of the system thus can dynamically adapt to a changing environment.

In the following section, we will describe swarming intelligence which deals with the emergence of group behavior and show how it integrates with Artificial Life. We will then point out how mobile agents can be regarded as a challenging technology for swarming intelligence strategies. Section three introduces a swarming intelligence architecture for load balancing in telecommunications networks which is based on mobile agents. Section four describes different strategies for the load balancing process which can be implemented with mobile agents. The development of a simulative tool for the evaluation of these strategies and the results are presented in section five. The final section concludes with a summary and an outlook on future work.

2. SWARMING INTELLIGENCE

In the first section, Artificial Life has been introduced as a new problem solving paradigm for complex distributed systems and networks. Research in Artificial Life can be distinguished in two categories. First, work is being executed to analyze and imitate biological and social mechanisms. Second, the application and adaptation of models for lifelike behavior basing on biological systems to artificial systems is aimed at. In order to achieve the latter, in particular to apply Artificial Life models to computer and communication networks, mobile agents constitute a promising technology, as their characteristics can be mapped directly to the principles outlined above.

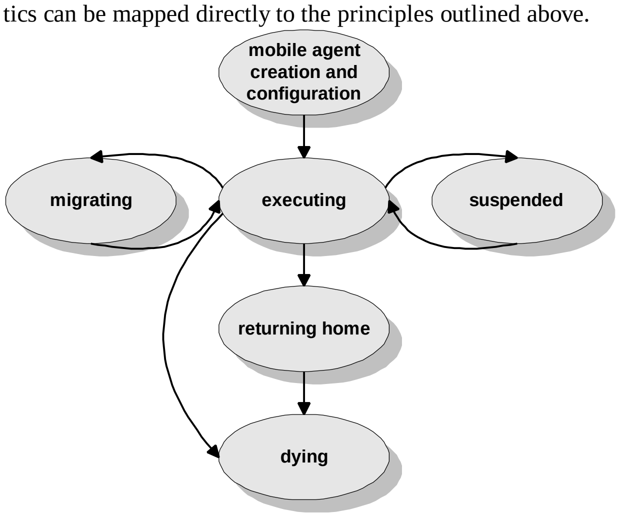


Figure 1: The Mobile Agent Lifecycle Model

As shown in figure 1, mobile agents adhere to a lifecycle which is identical to the property of temporal invariance. Equally importantly, mobile agents are able to migrate, i.e. to take both their code and status along and transfer themselves to another host and continue execution there, thus fulfilling the mobility property. The ability to migrate also enables the meeting concept where mobile agents first meet at a common location and then interact locally, thus addressing the locality property. Finally, as mobile agents act autonomously and in addition are parallel processes, they can be taken as a basis to transfer Artificial Life models to computer and communication networks.

In this paper, mobile agents technology is used for a decentralized load balancing of telecommunications networks. The problem solving technique applied is a swarming intelligence method, i.e. based on independent, autonomous agents with the overall system behavior drawing from the emergence of their interaction. Figure 2 shows how swarming intelligence can be classified as part of Artificial Life.

Swarming intelligence solutions offer a number of advantages if applied in distributed system and networking issues. First of all, in contrast to centralized approaches, there is no problem with scalability, as a swarm of independent agents constitutes an entirely decentralized solution which can adopt to the size of a system by reproduction and migration. In addition, swarming intelligence provides highly adaptive systems, as the agents' lifecycle and their ability to migrate allows to dynamically adopt to changing system requirements. Therefore, swarming intelligence is particularly suitable for large and highly dynamic systems. Moreover, with agents being autonomous, i.e. able to execute without relying on other agents, the overall system based on independent agents is robust and fault tolerant. While the crash of a central component in a centralized approach will cause the entire system to fail, the termination of a group of agents can cause a system to act less efficiently, but it will not cause the entire system to halt.

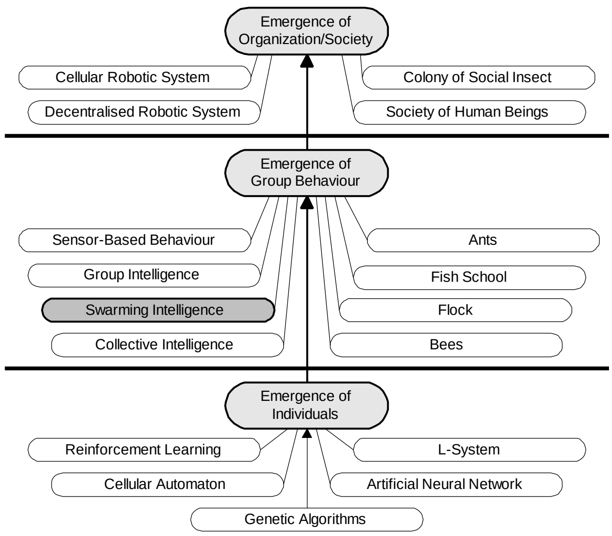


Figure 2: Swarming Intelligence in Artificial Life

Last but not least, conceiving a problem solution based on individual and self-contained components results in highly modular and clearly structured systems, thus improving maintenance and updates. Bearing these potential benefits in mind, the next section explains how mobile agent technology can be deployed for a swarming intelligence in the load balancing process of a telecommunications network.

3. A MOBILE AGENT BASED ARCHITECTURE FOR LOAD BALANCING

A vital issue in telecommunications networks is the availability of lines and services. First of all, it must be ensured that the number of calls which are blocked or lost are kept to a minimum. At first sight, it might therefore be a straightforward solution to generously provide capacities to avoid situations of high load. This, however, is neither efficient nor economically feasible. It must rather be aimed at utilizing the available capacities to a maximum degree. Load balancing addresses the topic of distributing load over the nodes of a network. As a consequence, a higher number of calls will then be allowed to go through and co-exist.

Recent research demonstrates the general applicability of mobile agent technology for network management and its potential benefits [7,8,11], especially if the mobile agents form a swarming intelligence [12,13]. Examinations of swarming intelligence in specific areas of network management cover configuration management [14] and fault management [15]. In the following, we will focus on swarming intelligence with mobile agents for load balancing in telecommunications networks and present a decentralized architecture which suits this purpose. As discussed later in this paper, it will serve as an enabling technology for the load balancing strategies.

In accordance with the approach introduced in [16], two classes of mobile agents are defined by the architecture depicted in figure 3, load agents and strategy agents. Load agents operate on the lowest layer of the architecture. If a load agent is emitted into the network, it will determine the paths offering the largest free capacity from the current node to each of the other nodes in the network and then modify the routing tables accordingly, making use of its ability to migrate to the other nodes. The algorithm of Dijkstra [17] is applied, and the mobility of agents allows a straightforward realization of this algorithm. While the details of the algorithm are omitted here, it is important to note that the updates of the routing tables have to be made in reverse order, i.e. starting from the target node, in order to avoid loops.

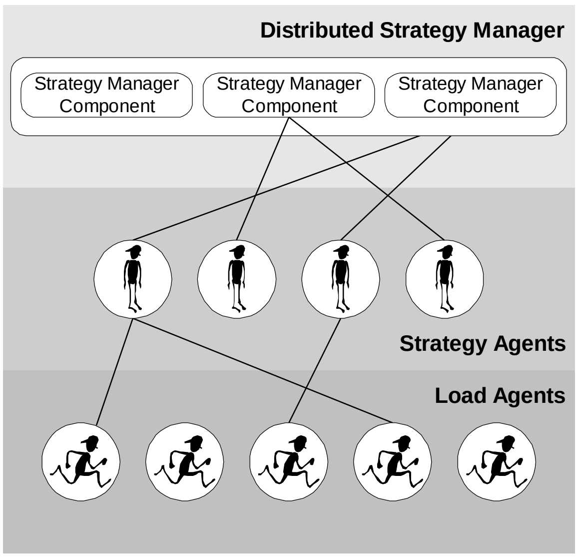


Figure 3: Outline of the Load Balancing Architecture

The remaining capacity of an entire path is set by the connection element with the minimum free capacity. Hence, the determination of the path offering the maximum free capacity requires an examination of all available paths and their elements and the selection of the path with the maximum value. In this process, both the links and the nodes of a network can be taken into account, because either of these can form a bottleneck for a connection. Hence, with given nodes and links of varying capacities, different paths can be selected according to the selection criteria. An example is given in figure 4. The numbers indicate the free capacity of the links and nodes. Taking only the values of the links into account will result in a selection of path A, whilst regarding the nodes or both links and nodes will result in selection of path B. The figure thus also expresses that depending on the selection criteria, the path which is selected need not be the shortest connection, as there is also a direct connection between the two nodes available. The selection of longer paths will result in an increased load of the overall network. Therefore, load agents can also be instructed to prioritize short paths.

Deploying mobile agents to execute the updates of the routing tables allows to easily modify the selection criteria. It is done simply by emitting the corresponding agents which will do the required modifications of the routing tables while considering a given subset of the criteria. Hence, a change of the criteria can even be done at runtime by emitting agents with a modified set of criteria. This is an indication of the flexibility of the mobile agent approach, in addition the modularity of the architecture given by the fact that the modifications of the load agents are entirely transparent to strategy agents which operate on top of them. Analogously, the operation and modification of strategy agents is independent of the underlying load agents.

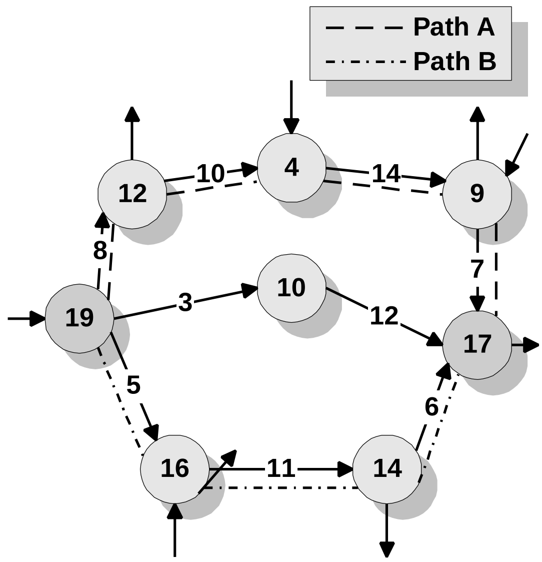


Figure 4: Selection of the Optimal Path

Strategy agents are responsible for the population of load agents, i.e. their creation and termination, and for delegating tasks to them. Since no central instance is given in the mobile agent approach for load balancing, strategy agents will move around the network randomly and gather information about the links and nodes, such as the current load and the number of calls originating from a node. Comparing the current values of the load to the mean values of former visits, the strategy agents can detect changes of the traffic and the network itself and then decide on emitting load agents accordingly. What number of load agents to be created and which node to select for their emission, these factors are determined by the strategy applied by the strategy agent. These strategies are presented in detail in chapter 4.

According to the responsibilities of the strategy agents, they operate in two different modes. The first mode is a purely observing one, where information about the environment is being collected. If an irregularity or overload situation has been detected, the strategy agent will switch to the second mode where it will emit load agents and thus start fixing the situation, if no other strategy agent has already started this process. According to these two modes, strategy agents will mutually influence their route. A strategy agent will move to another node, if the current node is already being observed by a predefined number of agents, if another agent is already working on that node, or if there already is a sufficient number of load agents updating the routing tables in the network.

This two-layered architecture offers the advantages of load balancing which were discussed in section 2, e.g. scalability and adaptability. Direct inter-agent communication, an important research issue for enabling mobile agent deployment in many application domains, is a critical factor for mobile agent solutions. It is currently being addressed by standardization [18,19] and research [20,21,22]. Based on swarming intelligence, the architecture presented here is entirely based in indirect communication of agents, i.e. both load and strategy agents leave marks for their peers and other agents to be found. This avoids problems of agent synchronization and also equips the architecture with additional robustness.

What is still needed, however, is an additional layer for the management of the strategy agents, which will allow operations such as monitoring, change of strategies etc. The third layer is a Distributed Strategy Manager, depicted in figure 3, which consists of distributed Strategy Management Components. The Distributed Strategy Manager thus adheres to the de-centralized approach and offers visualization and steering facilities to the human manager. For example, if a change of strategy is to be carried out, the Strategy Management Components will be told to terminate all strategy agents following the expired strategy and replace them with new ones. If a strategy agent has crashed, a new strategy agent will be created by a Strategy Management Component. Additional strategy agents will be created, if a network error is imminent and error detection and recovery is to be strengthened. Timestamps can be deployed to determine failure of agents and also the current agent population of the system. For details on timestamp usage, see [16].

An architecture based on reactive mobile agents as presented in this chapter allows different strategies for load balancing in telecommunications networks. In the following chapter, details of these strategies are discussed in order to explain their mode of operation, before their efficiency is analyzed in section 5.

4. LOAD BALANCING STRATEGIES

The endeavor to optimize the usage of networks and in particular telecommunications networks has lead to intense investigations of distributing load over the available nodes. According to the effort made to balance the load of a network, three main groups of routing strategies can be distinguished: static strategies, dynamic strategies, and swarming intelligence strategies with mobile agents. In the beginning, only static routing was applied. In these approaches, specific routing tables were generated before a network was taken into operation. These routing tables were independent of time and load situations. Methods belonging to this category are for instance FIX (Fixed Routing) and FAR (Fixed Alternate Routing). In the FIX strategy, all routing tables are set up to contain only the shortest path to the destination nodes. Consequently, no load balancing is done, since all connections between nodes are predetermined. With the network's load growing, however, it is obvious that these paths soon will overload and calls will be lost, as there is no adaptation to this situation. Therefore, in the FARx strategy, x alternative paths to a destination node are memorized in an ordered list. If the capacity of the optimal route is low or insufficient, the follow-up route will be chosen from this ordered list of paths. Although this approach allows to avoid overload situations through a first, yet very rigid load balancing, it still does not adapt to the actual load of the network.

In recent years, dynamic strategies have come up which allow an adaptation of the routing tables at runtime, depending on the given network load. The best known strategies of this category are DAR (Dynamic Alternate Routing) from British Telecom [23], ADR (Adaptive Dynamic Routing) from Northern Telecom [24,25], and DNHR (Dynamic Non-Hierarchical Routing) from AT&T [26]. All of the dynamic strategies determine a number of alternative paths from source nodes to destination nodes, but in contrast to static strategies, the current load of the nodes and links are taken into consideration when this list of paths is frequently updated. This guarantees that in situations of high load on a given path, the alternative path which meets the current situation of the network best will be selected.

A new category of strategies is established when evaluating the applicability and efficiency of mobile agent based swarming solutions for load balancing, which would provide the benefits described in section 2. In the remainder of this paper, we will present five strategies of this kind [27] which are realized with the architecture described in section 3. Their performance will be compared to five strategies representing the first two groups, namely static, multi-path and alternative routing, with a specifically developed simulation tool.

Two static strategies (named strategy 0 and 1) have been examined. They differ in the number of alternative paths available for connecting source and destination nodes. In strategy 0, the optimal path according to the algorithm of Dijkstra for connecting each source node to the destination nodes is calculated in advance and written to the routing tables, and no load balancing is done at runtime. Strategy 1, however, selects a predefined number of low-cost paths from the set of all possible paths. Since there might be a huge overall number of paths, a restriction of the length of the paths to be selected can be introduced via a hop count. In the simulation presented in section 5, the load in strategy 1 will be evenly distributed over all of these low-cost paths.

Three strategies (named strategy 2, 3, and 4) representing the class of dynamic strategies have been simulated. Strategy 2 is a decentralized alternative routing, where all alternative paths (possibly restricted by a hop count) are determined in advance. On network operation, each of the nodes will examine its current load in fixed intervals and will select the most suitable path from the predefined set, depending on this local information. Strategy 3 is also decentralized, but does not operate on a given set of alternative paths. It rather calculates the optimal path to destination nodes on the fly, using the Dijkstra algorithm. This calculation process is triggered in parallel on all nodes in given intervals. A variation of this approach is given with strategy 4. In order to avoid the massive computation required in strategy 3, the process of applying the Dijkstra algorithm is here started on the nodes of a network in a sequential order, also in given intervals.

The strategies based on the swarming intelligence architecture presented above are named strategy 5 to 9. They operate on routing tables which are initially set to the optimal paths between nodes (only strategy 9 holds a list of alternative paths) and dynamically and adaptively modify these routing tables during network execution through the load agents. The strategies differ from each other in the applied method to determine the location for emitting the load agents into the network. In strategy 5, the strategy agents select the node for launching a load agent from of a list containing the ten most recently visited nodes. From this list, the node is selected where most calls originate from, i.e. the one with the highest source rate. Load agents in strategy 6, however, will be started on a node next to the one which currently holds the largest source rate. This aims at freeing capacity at the overloaded node without adding the additional computation overhead for the route calculation to the node itself. Similarly, strategy 7 starts load agents on all neighboring nodes of the overloaded node, in order to maximize the amount of traffic taken away from it. Strategy 8 is similar to strategy 5, but here threshold values are introduced to avoid instabilities. These instabilities have emerged with strategy 5 and were caused by a continuous re-routing of the load. Finally, strategy 9 is a modification of the alternative routing given in strategy 1, as all possible paths from one node to another are computed in advance. Load agents here are emitted in analogy to strategy 5, but they operate only of the predefined set of alternative paths, i.e. if a strategy agent detects an overloaded node, a load agent will be emitted to the node with the maximum source rate. It will then determine the path offering the largest free capacity from the set of known paths. This holds the advantage that the computation effort is reduced, but it also neglects even better paths which are not contained in the set.

The following section presents the analytical conditions and the results of the simulations for the individual strategies and thus indicates, in which way mobile agent based swarming intelligence can contribute to load balancing of telecommunications networks.

5. SIMULATION RESULTS

The aim of load balancing mechanisms is to evenly distribute load over a network, thus leaving no nodes idle and preventing overload of others. In other words, the variance of the load over all nodes within a network is to be minimized. In order to formally specify this property, the load of a network has to be examined in more detail.

Let $s_i(t)$ be the overall load caused at node *i* by outgoing calls which terminate at node *k*, i.e.

$$s_i(t) = \sum_{k=1}^{n} A_{ik} \qquad \text{(I)}$$

Analogously, let $r_i(t)$ be the overall load of incoming calls at node i, namely

$$r_i(t) = \sum_{k=1}^{n} A_{ki} \qquad \text{(II)}$$

Additional load at node *i* results from paths from node *j* to node *k* that use this node. This load is given by

$$u_i(t) = \sum_{\substack{j,k=1 \\ j \neq k}}^{n} A^{i}{}_{jk} \qquad \text{(III)}$$

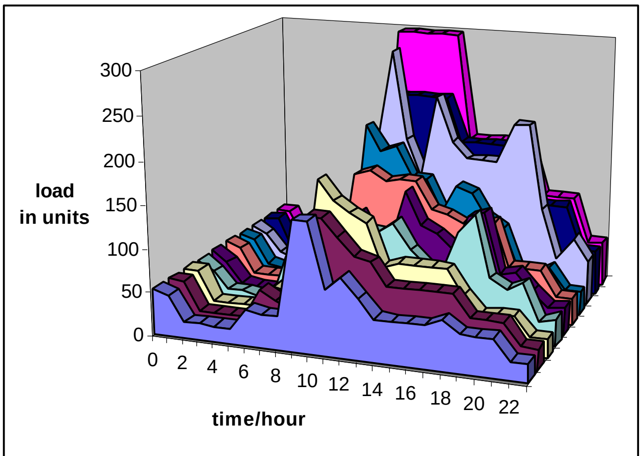
With (I-III), the entire load $L_i(t)$ currently given at a node i can therefore be determined as

$$L_i(t) = s_i(t) + u_i(t) + r_i(t), \text{ where } s_i, u_i, r_i \geq 0 \qquad \text{(IV)}$$

Given the load of the individual nodes of the network, the minimum variance of the loads in the network can be expressed with the following optimization function:

$$MIN = \int_{t_1}^{t_2} \left(\sum_{i=1}^{n} \left| \frac{\sum_{k=1}^{n} L_k(t)}{n} \right| - L_i(t) \right) dt = \int_{t_1}^{t_2} \left(\sum_{i=1}^{n} \sum_{\substack{k=1 \\ i \neq k}}^{n} \left| L_i(t) - L_k(t) \right| \right) dt \qquad \text{(V)}$$

Basically, what is described here is a summation of the load of all nodes which is then divided by the number of nodes in the network. This will result in the mean load in the network at time t. Given this mean value, the absolute value of each node's load varying from the average load is determined. The summation of these values and their integration gives the variance of the load in the network.



Front to back:

Strategy 5 (swarming)	Strategy 6 (swarming)
Strategy 8 (swarming)	Strategy 7 (swarming)
Strategy 9 (swarming)	Strategy 2 (dynamic)
Strategy 4 (dynamic)	Strategy 0 (static)
Strategy 3 (dynamic)	Strategy 1 (static)

Figure 5: Comparison of the Mean Variance of all Strategies

This formula has been taken as a basis for realizing a simulative tool, which helps to compare the load balancing strategies presented above. For a given network, the tool first initializes the routing tables of the nodes according to the requirements of the current strategy and assigns an initial load to the nodes. During the simulation of the network operation, the load of the individual nodes is frequently updated, i.e. for each new connection and for each terminated connection, the load of the nodes involved is modified accordingly. The number of calls to be initiated and terminated is taken from probability curves denoting the call frequency at each time of the day. These curves can be adapted to suit the distribution of calls over a period of time, e.g. to reflect that during business hours, a higher number of calls is established than at night.
In parallel to the load distribution, the load balancing processes are simulated, with the mobility of mobile agents taken into consideration in case of swarming intelligence strategies. Simulations have been made for each of the ten strategies. The mean variance of the strategies, which allows to compare their overall effectiveness, is shown in figure 5. It can be seen that the strategy displaying the lowest variance and thus the highest effectiveness is strategy 5, the swarming intelligence solution using the list of most recently visited nodes to determine the location to start the load agents. The high adaptability of this strategy is depicted in more detail in figure 6. The progress of the load distribution over the selected nodes very quickly adapts to an increased and decreased load of the network. This is an indication of the load agents taking specific actions, with the overall effect being not only an evenly distributed load, but also a very stable operation of the network, as the immediate adaptation minimises temporary imbalances.

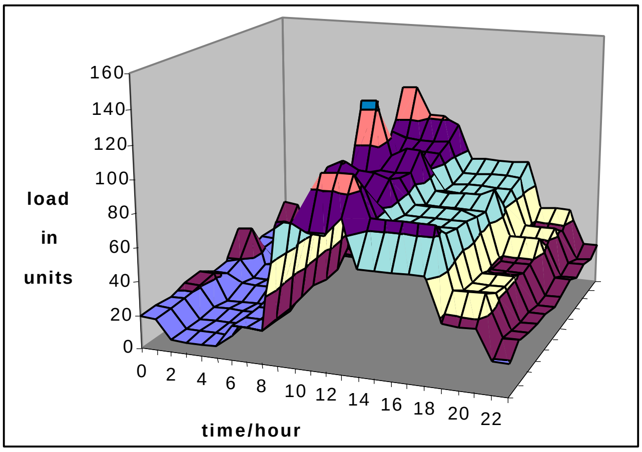


Figure 6: Load Distribution over the Nodes for Strategy 5

Comparing the swarming intelligence strategies to the static and dynamic ones, figure 5 also shows that swarming intelligence offers very efficient solutions. The three most efficient strategies are all swarming intelligence based. Merely strategies 6 and 7 perform worse than some dynamic strategies. This indicates that starting load agents on neighboring nodes to avoid additional usage of highly loaded nodes does not have a sufficient effect on the traffic of the target node. Finally, it can be seen, that static strategies are much less successful in providing network robustness and use to capacity.

6. CONCLUSION AND OUTLOOK

In this paper, we have evaluated the deployment of mobile agents in form of swarming intelligence for load balancing in telecommunications networks. Having introduced swarming intelligence characteristics and benefits, we have presented an architecture deploying layers of mobile agents which serves a robust, flexible, scalable, and adaptive load balancing. The efficiency of strategies based on this architecture has been shown in a simulative approach, using a specifically developed tool. For this tool, we have made an analytical model of the load distribution in a network, which resulted in the definition of a formula denoting the variance of the load in telecommunications networks.
As a main result, it can be stated that deployment of autonomous mobile agents, which is being discussed for different application areas, especially e-commerce, user management, and network management, has proven to offer benefits for network management, specifically load balancing. Future work concerning load balancing will focus on equipping the agents with one additional characteristic, namely the ability to learn. This will allow the definition of active agents, rather than the purely reactive ones deployed here. Agents will then be capable to understand recurring traffic patterns and to take precautionary actions. For example, they can be able to foresee situations of high load and act e.g. by adjusting the number of load agents in advance.

7. ACKNOWLEDGMENTS

This work was partially funded by the German Research Council (DFG) under SFB 476-99 IMPROVE and GRK 185/3-99.

7. REFERENCES

[1] Raman, L.: OSI System and Network Management. In: IEEE Communications: Management of Heterogeneous Networks, vol.36, no.3, March 1998

[2] Stallings, W.: SNMP and SNMPv2: The Infrastructure for Network Management. In: IEEE Communications: Management of Heterogeneous Networks, vol.36, no.3, March 1998

[3] Goldszmidt, G.; Yemini, Y.: Delegated Agents for Network Management. In: IEEE Communications: Management of Heterogeneous Networks, Vol.36, No.3, March 1998

[4] Pavon, J.; Tomas, J.: CORBA for Network and Service Management in the TINA Framework. In: IEEE Communications: Management of Heterogeneous Networks, Vol.36, No.3, March 1998

[5] Harrison, C.; Chess, D.; Kershenbaum, A.: Mobile Agents: Are they a good idea? Technical Report RC 19887, IBM Res., March 1995

[6] Pham, A.; Karmouch, A.: Mobile Software Agents: An Overview. In: IEEE Communications Magazine, July 1998

[7] Baldi, M.; Gai, S.; Picco, G.: Exploiting Code Mobility in Decentralised and Flexible Network Management. In: Proceedings of First International Workshop Mobile Agents '97, Berlin, Germany, April 1997

[8] Bieszczad, A.; Pagurek, B.; White, T.: Mobile Agents for Network Management. In: IEEE Communications Surveys, September 1998

[9] Langton, C.: Artificial Life. Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Things, Los Alamos, New Mexico, 1987

[10] Forrest, St.: Emergent Computation: Self-Organising, Collective and Co-Operative Phenomena in Natural and Artificial Computing Networks. In: Emergent Computation. MIT Press, 1991

[11] Sahai, A.; Morin, C.: Enabling a Mobile Network Manager (MNM) Through Mobile Agents. In: Proceedings of Second International Workshop Mobile Agents '98, Stuttgart, Germany, 1998

[12] White, T.; Pagurek, B.: Towards Multi-Swarm Problem Solving in Networks. In: Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS '98), July, 1998

[13] White, T.; Pagurek, B; Oppacher, F.: ASGA: Improving the Ant System by Integration with Genetic Algorithms. In: Proceedings of the 3rd Conference on Genetic Programming (GP/SGA'98), July 1998

[14] Pagurek, B.; Li, Y.; Bieszczad, A. et al.: Network Configuration Management in Heterogeneous ATM Environments. In: Proceedings of the 3rd International Workshop on Agents in Telecommunications Applications IATA'98, Agent-World'98, Paris, France, July 1998

[15] White, T.; Bieszczad, A.; Pagurek, B.: Distributed Fault Location in Networks Using Mobile Agents. In: Proceedings of the 3rd International Workshop on Agents in Telecommunications Applications IATA'98, AgentWorld'98, Paris, France, July 1998

[16] Appleby, S.; Steward, S.: Mobile Software Agents for Control in Telecommunication Networks. In: BT Technology Journal, April 1994

[17] Dijkstra, E.W.: A Note on Two Problems in Connection with Graphs. In: Numeric Mathematics 1, p. 269 – 271, 1959

[18] Foundation for Intelligent Physical Agents: Agent Communication Language. Geneva, Switzerland, October 1997

[19] GMD Fokus: Mobile Agent System Interoperability Facilities Specification. Joint Submission, November 1997

[20] Baumann, J.; Hohl, F.; Radouniklis, N. et al.: Communication Concepts for Mobile Agent Systems. In: Proceedings of First International Workshop Mobile Agents '97, Berlin, Germany, April 1997

[21] Dömel, P.; Lingnau, A.; Drobnik, O.: Mobile Agent Interaction in Heterogeneous Environments. In: Proceedings of First International Workshop Mobile Agents '97, Berlin, Germany, April 1997

[22] Lipperts, S.: CORBA for Inter-Agent Communication of Management Information. In: 5th International Workshop on Mobile Multimedia Com-mu-ni-ca-tion, Berlin, Germany, October 1998

[23] Stacey, R.; Songhurst, D.: Dynamic Alternative Routing in the British Telecom Trnk Network. In: International Switching Symposium ISS87, Phoenix, USA, 1987

[24] Chemuil, P.; Gauthier, P.: Adaptive Traffic Routing in Telephone Networks. In: L'Echo de Recherches, English Issue, 1990

[25] Regnier, J.; Cameron, W.: State-Dependent Dynamic Traffic Management for Telephone Networks. In: IEEE Communications Magazine, October 1990

[26] Ash, G.; Cardwell, R.; Murray, R.: Design and Optimisation of Networks with Dynamic Routing. In: Bell System Technical Journal, vol.60, 1981

[27] Kreller, B.: Mobile Agents for the Efficient Load Balancing Management in Telecommunication Networks. Otto-v.-Guericke-Universität Magdeburg, 1996