Simulating Autonomous Mobile Programs on Networks

Natalia Chechina, Peter King, Rob Pooley and
Phil Trinder
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, EH14 4AS, UK
Email: {nc75, P.J.B.King, R.J.Pooley, P.W.Trinder}@hw.ac.uk

Abstract—**Autonomous mobile programs (AMPs) have been proposed for load management in dynamic networks. An AMP is aware of its resource needs and periodically seeks a better location in the network to reduce execution time. AMPs have previously been measured using mobile Java Voyager on local area networks (LANs).**

We have constructed a simulation model of AMPs and reproduced 4 sets of experiments on homogeneous networks, i.e. networks where all locations have the same processor speed, and 2 sets of experiments on heterogeneous networks with collection of large and small AMPs. The results show that simulated collections of AMPs obtain similar balanced states to those reached in the real experiments, and have only minor differences from real experimental results. The simulation model gives an opportunity to explore the greedy effect that can be observed in the real experiments. This gives us confidence to apply the simulation model for further investigation of AMP behaviour, including behaviours on wide area networks.

I. Introduction

The continuing decline in price and increase in network size and interconnection speed allows developers to create and apply mobile computations [1]. Usually distributed systems are developed to be driven by concerns relating to the collective goals independent of the actual environment they are running. So, such systems tend not to communicate with environment they are executed in, which impacts on their effectiveness in achieving the collective goal. To overcome this drawback autonomous mobile programs were developed.

Autonomous mobile programs (AMPs) are mobile agents, which aim to exploit advantages of mobile computations, agents and autonomic techniques [2]. They are aware of their resource needs, sensitive to the environment in which they execute and periodically seek better location for execution.

Results of real experiments on local area networks showed that AMPs are able to dynamically relocate themselves and quickly obtain optimal or near optimal balance for minimizing execution time.

The aim of the current research is to obtain a detailed understanding of autonomous mobile program behaviour on local area networks. For this purpose real experiments for homogeneous and heterogeneous local area networks are reproduced [2]. Other goals are the estimation of AMP capabilities and exploring the ways maximum utilization of AMP advantages.

The organisation of the paper is as follows. Section II gives related works, Section III describes simulation parameters. In Sections IV and V we present results of simulation experiments for networks with homogenous and heterogeneous locations respectively. Finally, in Section VI we summarize our work and discuss future directions for research.

II. Related Work

A. Load Management

Autonomous Mobile Programs (AMPs) were developed to manage computing load in a network [2]. The aim of AMPs is to minimise their time to completion by balancing the load on computers in the network.

The problem of load balancing in distributed computer systems has been widely studied, e.g. [3], [4], etc. The main difficulties which load balancing approaches face are minimizing execution time and maximizing resource utilization [5].

According to [6] AMPs are global dynamic load balancers with a hybrid scheme, which make decisions on the basis of a simple sender initiated policy [7].

An AMP is a mobile agent. In the current research, mobile agent means mobile computation, which decides whether it should migrate or not.

According to *which* entity decides what must be moved, AMPs are implemented on the basis of implicit mobile computations and make decisions about movement themselves [8].

The current simulation model is designed to reproduce real experiments with AMPs, which were programmed using Java Voyager [9].

B. Autonomous Mobile Programs

Autonomous mobile programs (AMPs) are developed for load management on large and dynamic networks. AMPs migrate within the network to raise program execution efficiency and exploit network resources.

Most autonomous mobile agent systems adapt their computations, however AMPs adapt their coordination. Hence, the feature of AMPs is *where* the program executes and not what it does.

One of the main aims for AMP development is minimizing execution time by the seeking the fastest location. AMPs are aware of their execution needs and of the execution parameters of locations on the network.

To estimate the program execution time T_{total}, the following cost model is used:

$$T_{total} = T_{Comp} + T_{Comm} + T_{Coord}, \quad (1)$$

where T_{Comp} is computation time, T_{Comm} is total communication time and T_{Coord} is total coordination time. The cost model is parameterised on system architecture, which includes location and interconnect speeds, data size, number of locations, cost of data processing and communicating.

The main rule that AMPs follow in making decision to migrate to another location or not is whether execution time on the current location T_h exceeds time for single communication T_{comm} and execution time on the new location T_n:

$$T_h > T_{comm} + T_n. \quad (2)$$

On the basis of this comparison, AMPs decide whether to stay on the current location or to move.

III. Simulation Parameters

A simulation model of AMPs on a local area network is constructed using OMNeT++ network simulator [10]. The simulation is validated by reproducing the results reported by [2]. The simulation is then used to investigate the behaviour of AMPs on heterogeneous networks.

The network is a fully connected graph of locations (computers). A single location in the simulation network comprises three main parts (devices): *Generator*, *Queue* and *Switch* (see Fig. 1). The *Generator* is used for generation of new messages. The *Queue* is used as a cell for message storage during execution. The *Switch* is a main device in the location, which combines all components, and makes decision about place and speed of message processing.

The real experiments were made with AMPs which implemented matrix multiplication. The cost model parameters for a LAN were measured and as follows [2]:

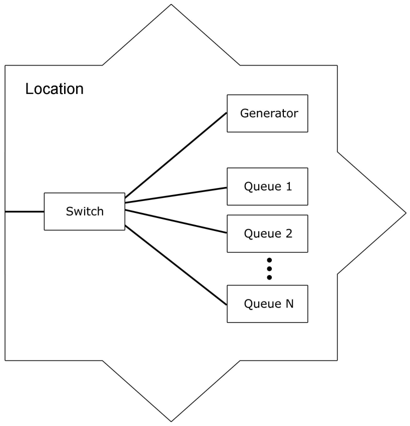

Fig. 1. A single location in the simulation network

- The total work W that must be done for square matrix multiplication of dimension d is [2, p. 57]:

  $$W = d^3. \quad (3)$$

  For current experiments 1000×1000 matrices are used.
- AMP migration takes place if condition (2) obeys. Time for communication T_{comm}, which is measured for d=1000, is 5.099 seconds [2, p. 79]. In the real experiments communication time is not a constant value, but depends on different factors, such as network workload, hardware, etc. To approximate this variability in the communication time, the simulation uses an exponentially distributed random variable for the communication time, with mean given by T_{comm}.
- The fragment of work $gran$ that must be executed between searches for a better location is [2, pp. 70, 79]:

  $$gran = \frac{T_{coord} \cdot S_h}{O}, \quad (4)$$

  where T_{coord} is a time for coordination in the load server architecture. Load server maintains information about location load, which allows to reduce coordination time for AMPs and time to discover load information. Results of real experiments showed that T_{coord} is 0.011 second [2, Subsection 5.2.3]. S_h is a speed of current location. For the experiments the value of overhead O is equal to 5%.
- Initially all AMPs commence execution on the first location.
- The real experiments showed that the first location only had 50% of its capacity available for serving the AMP workload [2, p. 116]. This feature is attributed to the overhead of communication with remote processes that were sent to other locations. Hence, we assign the first location a speed of 50% of its clock rate.
- Work-time ratio is estimated on the basis of experimental result analysis, presented in table 4.11 of [2]. It shows that a CPU speed of 3139 MHz corresponds to 21365101 work/second. Others simulation speeds are found using the same ratio.

IV. Homogeneous Network

This section reports comparable results for the simulation and real AMPs on *homogeneous* networks, i.e. where all locations have identical processor speed.

Four types of experiments are implemented for homogeneous networks: optimal balance, near-optimal balance, adding and removing AMPs [2, Subsection 5.3.1]. The number of processors is ranged from three to five, and between five and thirteen AMPs are started at the first location. Speed of all processors is 3139 MHz.

A. Optimal Balance

The first type of experiment tests the distribution of AMPs. AMPs start execution on the first location and after some time are distributed over all locations. The number of AMPs and the number of locations was chosen to allow equal numbers

TABLE I
OPTIMAL BALANCED DISTRIBUTION IN THE REAL AND SIMULATION EXPERIMENTS

	5 AMPs	7 AMPs	9 AMPs	10 AMPs	13 AMPs
3 Locs					
real	1/2/2	1/3/3	1/4/4	-	-
simulation	1/2/2	1/3/3	**2/3/4**	-	-
4 Locs					
eral	-	1/2/2/2	-	1/3/3/3	1/4/4/4
simulation	-	1/2/2/2	-	1/3/3/3	**2/4/4/3**
5 Locs					
real	-	-	1/2/2/2/2	-	-
simulation	-	-	1/2/2/2/2	-	-

TABLE II
NEAR-OPTIMAL BALANCE IN THE REAL AND SIMULATION EXPERIMENTS

	5 AMPs	6 AMPs
2 Locs		
real	2/3	-
simulation	2/3	-
3 Locs		
real	-	1/2/3
simulation	-	1/2/3

of AMPs on all locations except the first. The results of the real [2, Table 5.25] and the simulation experiments are shown in table I. For each number of locations in table I, the first row is distribution in the real experiments, and the second row is distribution in the simulation experiments. Two simulation results which differ from the real experiments are highlighted in bold.

Results show that the simulation model reflects the real balancing of AMPs in the network, except for two cases: 3 locations with 9 AMPs and 4 locations with 13 AMPs. This mismatch is a result of using 50% of capacity at the first location. When the first location has 2 AMPs and the last has 4 AMPs the calculated speeds of the locations become equal. The AMP does not move because the communication time would be added to its total execution time and so it is better not to move. If the workload of 48% for the first location is used in the simulation model, then simulation results agree with the experiment results in table I.

B. Near-Optimal Balance

The second type of experiment investigates near-optimal balance. Near optimal balance is a state, when the total number of AMPs makes it impossible for equal numbers of AMPs to be at each location, but the discrepancy between locations should be at most one AMP [2].

Table II shows distribution of 6 AMPs between 3 locations and 5 AMPs between 2 locations. For each number of locations in table II, the first and the second rows represent distribution in the real [2, Figs. 5.56, 5.57] and simulation experiments respectively. As we can see, the results are identical.

C. Adding Autonomous Mobile Programs

Two experiments were conducted to analyse the distribution after adding more AMPs [2, p. 114].

The first experiment has 4 locations and 7 AMPs at the start. Then 3 more AMPs are added one by one. Fig. 2 and Fig. 3 show distribution and movements of AMPs between locations in the real and simulation experiments.

The states S1, S2, S3 and S4 depict the balanced states which the system obtains before (S1) and after (S2, S3, S4) adding AMPs. As locations in the experiments are homogeneous, and number of AMPs in the balanced states S1, S2, S3 and S4 for real experiments correspond with simulation experiments, the real and simulation results are identical.

The second experiment has 3 locations and 5 AMPs, which start execution on the first location. 4 more AMPs are added sequentially. Fig. 4 and Fig. 5 depict AMPs movement between locations in real and simulation experiments respectively. Identical balanced states are marked with states S1, S2 and S3.

However, after adding the third AMP, the movements of the simulated AMPs are not the same as the real AMPs. The mismatched states are marked with S4, S5 and K4, K5 in the real and simulation experiments respectively. The cause of the mismatch is the 50% first location workload, as discussed in Subsection IV-A.

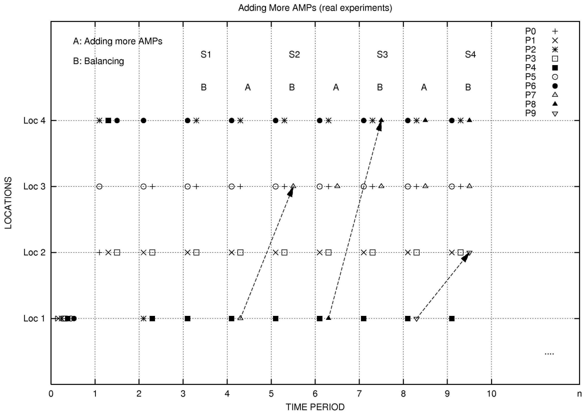

Fig. 2. 7+3 AMPs on 4 locations: real experiments

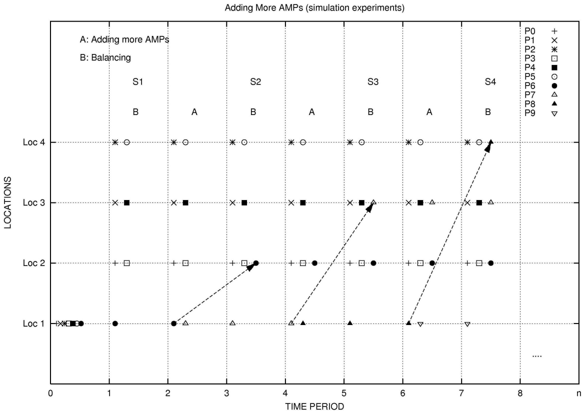

Fig. 3. 7+3 AMPs on 4 locations: simulation experiments

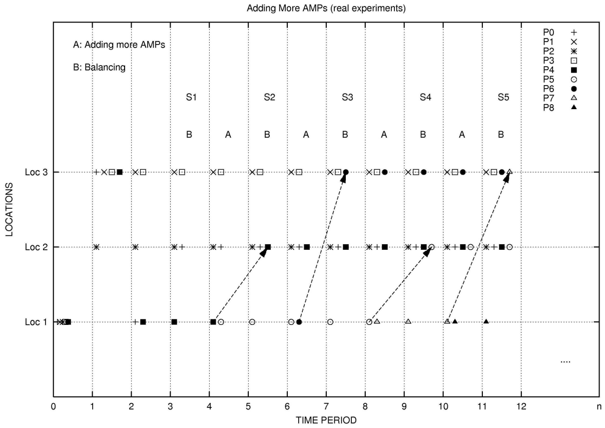

Fig. 4. 5+4 AMPs on 3 locations: real experiments

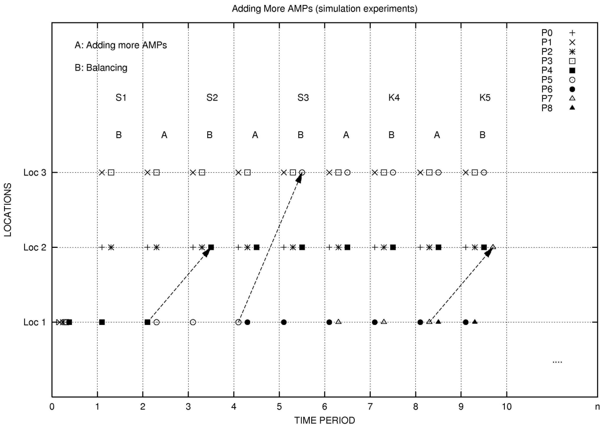

Fig. 5. 5+4 AMPs on 3 locations: simulation experiments

D. Removing Autonomous Mobile Programs

To examine AMP behaviour in the homogeneous network after removing of some AMPs, experiments with 5 large and 5 small AMPs are implemented. These reproduce experiments on the page 115 of [2]. Initially all AMPs are started on the first location. Large AMPs are matrix multiplications of size 1000×1000, and small AMPs are matrix multiplications of size 500×500.

Three locations are used in this experiment, and the initial distribution of small and large AMPs is random.

Fig. 6 and Fig. 7 depict the distribution of AMPs over locations in real and simulation experiments respectively. The sign '-' is used to show termination of execution of an AMP and its removal.

States S1, S2, S3 and S4 are balanced AMP distribution in the real experiments. All simulation experiments obtain states S1, S3 and S4, however, only 18% of experiments obtain state S2. This is due to the initial distribution of large and small AMPs, communication time, and, as a result, time of small AMP removing. Depending on which location an AMP terminates (and hence is removed) and on which other AMP detects the available processing power, state S2 (2/3/3) or K2 (1/3/4) may result. Both states are balanced. Table III shows the properties of experiments in which the corresponding states occurred. States K1, K2 and K3 are also balanced states, which the system may enter.

TABLE III
BALANCED STATES OF SIMULATION EXPERIMENTS

Balanced States	Per cent of experiments
S1 (2/4/4)	100%
K1 (2/3/4)	41%
S2 (2/3/3)	23%
K2 (1/3/4)	70%
K3 (1/3/3)	94%
S3 (1/2/3)	100%
S4 (1/2/2)	100%

Real experiments exhibit two additional rebalance states: R0 and R2 on Fig. 6, which are not typical for the current simulation model. These states come of receiving and sending locations take part in AMP transmitting in real experiments; as in the simulation experiments only receiver location contributes to the AMP transferring. However, implementation of sender and receiver participating scheme for AMP transferring in simulation model will reduce the model to the definite class of tasks. Because time and computer resources, which are

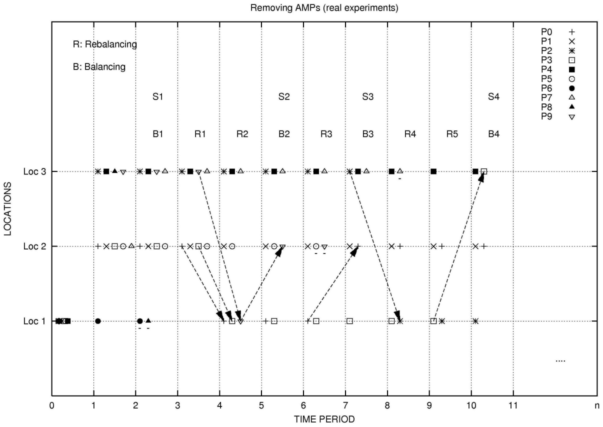

Fig. 6. AMP removing: real experiments

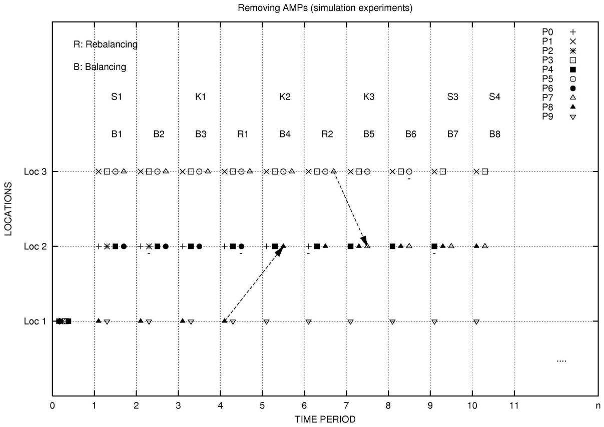

Fig. 7. AMP removing: simulation experiments

needed for program transferring, greatly depend on program size and complexity.

Thus, comparative analysis shows that simulation AMPs have similar behaviour to real AMPs in homogeneous networks. The differences in behaviour are caused by the use of a constant value for the workload at the first location and by the random initial distribution of large and small AMPs.

V. Heterogeneous Network

In this section, we present results of real and simulation experiments on the *heterogeneous* networks, i.e. networks where locations have different CPU speeds.

Two type of experiments are implemented for analysis of distribution in heterogeneous networks [2, Subsection 5.3.2]. AMPs of different sizes are used in the experiments. By the *different sizes of AMPs* we mean different number of computations.

The first type of experiments is implemented for 25 AMPs and 15 locations with CPU speeds 3193 MHz ($Loc1 - Loc5$), 2168 MHz ($Loc6 - Loc10$) and 1793 MHz ($Loc11 - Loc15$). According to the CPU speed, locations can be divided into slow ($Loc1 - Loc5$), middle ($Loc6 - Loc10$) and fast ($Loc11 - Loc15$). 13 large and 12 small AMPs are started at initial time.

Fig. 8 depicts AMP distribution in the real experiments and Fig. 9 depicts distribution in the simulation experiments. The difference in the state B3 of real and simulation experiments is due to the type of locations, where an AMP first discovers the opportunity to move to a less loaded location. In the real and simulation experiments AMPs discover better location first from middle and fast locations respectively. Here, both states, which real and simulation systems enter, are balanced, and 18% of simulation experiments have the same distribution in the balanced states as real experiments, according to the groups of locations with the same CPU speeds.

Table IV represents distribution of balanced states B2 and B3 on Fig. 8 and Fig. 9, which the system enters after the first and second AMP removals. The '1/3...' represents the states with 3 AMPs at one location. '1/2...' represents the states where no location has 3 AMPs present. After AMP removals the states reached are balanced, irrespective of whether there are 3 or 2 AMPs at the first location.

If, after an AMP removal, a single movement of an AMP does not result in a balanced state, a second movement may occur. It depends on the type of location from which an AMP discovered better location first, and can be observed on Fig. 10. The solid lines show the optimal movement of AMPs, and the dotted lines show the movements that actually occurred. This is the so called *greedy effect*. The greedy effect is a result of the decentralised implementation for load management in the network, and AMP rule for self-optimization – making locally optimal choice.

TABLE IV
Type of stages after AMP removing

Type of distribution	After 1^{st} rem.	After 2^{nd} rem.
1/3...	88% of exp	41% of exp
1/2...	12% of exp	59% of exp

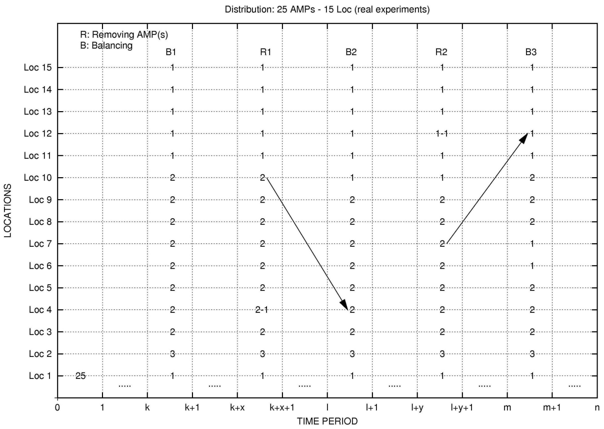

Fig. 8. Real AMP distribution in a heterogeneous network

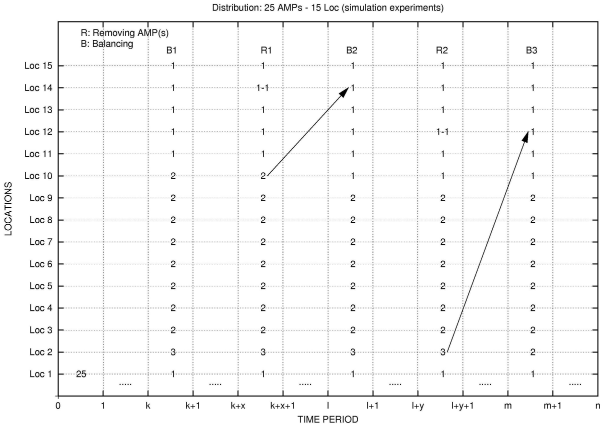

Fig. 9. Simulation AMP distribution in a heterogeneous network

As an example of this greedy effect, consider the set of state changes (states R1, B2) shown in Fig. 10, which were observed. An AMP terminates at $Loc14$ leaving it with excess capacity. All middle speed locations, $Loc6$–$Loc10$, have 2 AMPs and the fast locations $Loc2$–$Loc5$ all have 2 AMPs except $Loc2$ with 3 AMPs. The optimal movement pattern would involve an AMP moving from a middle speed location to $Loc14$. However, one of the fast locations with 2 AMPs ($Loc3$–$Loc5$) detects the availability of $Loc14$ first. Thus, an AMP moves from that location, leaving only 1 AMP. The second movement occurs when the fast location with only 1 AMP is detected by either a middle speed location or $Loc2$ which has 3 AMPs.

7% of simulation experiments with 25 AMPs and 15 locations show the worst case, when two AMPs move after every removal. This effect is also observed in the real experiment results [2]. Table V shows the number of movements after every removing in the simulation experiments.

As AMPs are not homogeneous, the type of locations, where AMPs finish their execution, is indifferent. However, in the 6% of simulation experiments the types of locations where

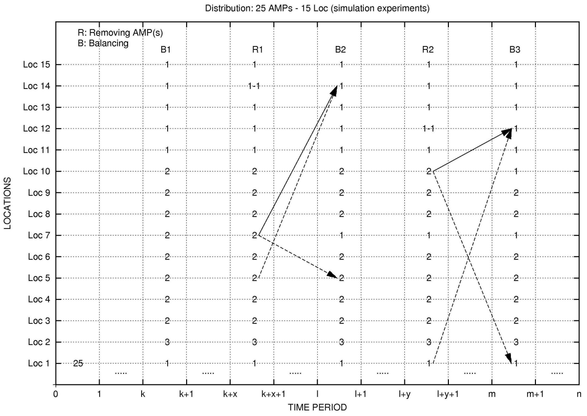

Fig. 10. AMP distribution in the heterogeneous network: greedy effect state changing in the simulation experiments

AMPs finish, agree with results in the states R1 and R2 of real experiments.

In the second type of experiments 20 AMPs on 10 locations are examined, with CPU speeds 3193 MHz ($Loc1 - Loc5$), 2167 MHz ($Loc6$) and 1793 MHz ($Loc7 - Loc10$). 10 small and 10 large AMPs are started on the first location. Results are similar to the results of experiments with 25 AMPs on 15 location [11].

The above analysis of AMP behaviour in heterogeneous networks shows that simulated AMPs reflect real AMP behaviour. The minor differences are result of random initial allocation of large and small AMPs and random location from which detects the location with extra capacity first.

VI. Discussion

To examine AMP behaviour in homogeneous and heterogeneous networks we constructed simulation model and implemented set of experiments. Comparative analysis of real and simulation experiments for optimal, near-optimal balancing, adding and removing AMPs in homogeneous networks shows the following:

Optimal balance. All distributions in simulation experiments are matched with the distributions in the real experiments, except two cases, when communication workload in the real experiments varies between 48% and 51%, as in the simulation experiments the workload is constant 50%.

Near-optimal balance. Real and simulation experiments enter identical states.

Adding AMPs. Simulation and real experiments obtain the same distribution. The cause of the only difference is the same as in the optimal balanced experiments.

Removing AMPs. All simulation experiments obtain 3 of 4 balanced states of real experiments, i.e. S1, S3, S4. 18% of simulation AMPs obtain all states of real experiments. 23% of simulation experiments have state S2, 70% have state K2, which is also balanced state. For generalisation and abstracting from particular type of programs, the AMP transmitting scheme that is used in the simulation model differs from the real one.

TABLE V
Number of movements

Number of mov.	After 1^{st} rem.	After 2^{nd} rem.
1	88% of exp	71% of exp
2	12% of exp	29% of exp

For experiments in heterogeneous networks with 25 AMPS on 15 locations and 20 AMPs on 10 locations the results are as follows:

- in the 41% and 58% of simulation experiments states B1, B2 and B3 coincide with the same states of real experiment in the first and second experiments respectively; here, other states, which simulation experiments obtain, are also balanced states;
- in the 6% of the first type simulation experiments AMPs remove from the same type of locations as in the real experiment;
- the greedy effect that can be observed in the real experiments, can also be seen in the simulation experiments.

We conclude that, other than a small number of explainable deviations *our current simulation is an excellent model of AMPs on LANs*. Hence, we are confident about using the model as the basis for further experiments, e.g. on simulated wide area networks.

Two directions of further work are being undertaken. First, we are analysing the greedy effect revealed by the extra movements to achieve balanced states in a heterogeneous network. Secondly, larger networks with different costs of reacting remote locations will be investigated to find AMP properties on such wide area networks.

References

[1] A. Fuggetta, G. P. Picco, and G. Vigna, "Understanding code mobility," *IEEE Trans. Softw. Eng.*, vol. 24, no. 5, pp. 342–361, May 1998.

[2] X. Y. Deng, "Cost driven autonomous mobility," Ph.D. dissertation, School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, UK, June 2007.

[3] R. van Renesse, H. van Staveren, and A. D. Tanenbaum, "The performance of the Amoeba distributed operating system," *Softw. Pract. Exper.*, vol. 19, no. 3, pp. 223–234, March 1989.

[4] N. G. Shivaratri, P. Krueger, and M. Singhal, "Load distributing for locally distributed systems," *IEEE Computer*, vol. 25, no. 12, pp. 33–44, December 1992.

[5] B. A. Shirazi, K. M. Kavi, and A. R. Hurson, *Scheduling and Load Balancing in Parallel and Distributed Systems*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1995.

[6] T. L. Casavant and J. G. Kuhl, "A taxonomy of scheduling in general-purpose distributed computing systems," *IEEE Trans. Softw. Eng.*, vol. 14, no. 2, pp. 141–154, 1988.

[7] H. G. Rotithor, "Taxonomy of dynamic task scheduling schemes in distributed computing systems," *IEE Proceedings Computers & Digital Techniques*, vol. 141, no. 1, pp. 1–10, January 1994.

[8] A. R. D. Bois, "Mobile computation in purely functional language," Ph.D. dissertation, School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, UK, April 2006.

[9] *Voyager Technical Documentation*, Recursion Software, Inc., 2591 North Dallas Parkway, Suite 200, Frisco, TX 75034, 2008. [Online]. Available: http://www.recursionsw.com/Products/voyager.html

[10] A. Varga, *OMNeT++. User Manual. OMNeT++ version 3.2*, October 2005. [Online]. Available: http://www.omnetpp.org/doc/manualusman.html

[11] N. Chechina, P. King, R. Pooley, and P. Trinder, "Simulation modelling of autonomous mobile programs on a LAN," Heriot-Watt University, Edinburgh, UK, Tech. Rep. 0065, April 2009.