Agent-based banking transactions & information retrieval – What about performance issues?

Jens Hartmann, Rune Evensen
Ericsson Eurolab Deutschland GmbH,
Ericsson Allee 1, D-52134 Herzogenrath
{Jens.Hartmann, Rune.Evensen}@eed.ericsson.se

Carmelita Görg, Peyman Farjami, Hai Long
Communication Networks
Aachen University of Technology
{cg, pemi, long}@comnets.rwth-aachen.de

Abstract – **For almost all applications within the telecommunication area speed and quality of data transfer is a very important factor. A main reason for using mobile agents in the telecommunication world is that there is a potential for better performance. In particular for mobile telecommunication networks like GSM, mobile agent technology can have considerable advantages. This is based on the fact that a GSM user has a relatively narrow bandwidth at his disposal, and that the connections are not always reliable. This is the case when the application does not need a permanent connection and the amount of data is relatively small. In order to verify these statements, an analytical performance model for mobile agents was derived. In order to test the performance model on a real application, measurements on an HBCI-based homebanking service for mobile agents [1] were carried out. In addition an optimisation strategy was defined and evaluated for an information retrieval service [2].**

I. INTRODUCTION

From the scientific disciplines artificial intelligence, distributed systems and object-orientation a new scientific working field called agent technology has been established. Currently agents are seen as a key to the issue of service provisioning within future telecommunication environments.

Fixed telecommunication networks are now providing not only voice services but also more and more data based services, whereas mobile networks are still developing into this direction. Compared with the traditional voice based networks, future networks need more flexibility to meet the new requirements. Agents, especially mobile agents, seem to be best suited because of their characteristics of autonomy, intelligence, mobility, co-ordination and co-operation.

The focus in this work is on mobile agents, in which the mobility of code, data and state is the most fundamental attribute. This allows these software entities to roam autonomously through a network and to perform dedicated tasks at specific network nodes, thereby taking advantage of locality [3].

The combination of mobile agents and mobile telecommunication systems is promising, because mobile agents can overcome the typical restrictions of mobile networks, such as:

- limited bandwidth
- high bit error rate on the air channel
- bounded coverage
- low processing power of the end-systems
- simple user interface

Moreover, the application of mobile agents for service provisioning brings the following advantages:

- asynchronous communication is possible
- agents could work without a permanent network connection
- reduction of network traffic
- very processing power consuming activities could be performed locally
- the reality could be better modelled with agents

Considering provisioning of new, sophisticated services by a network operator or service provider, the services should be provided in a more direct and flexible way. A user should have the opportunity to access these services anytime and anywhere, independent of the terminal and network technology.

Using agents, services can be easily subscribed, downloaded or migrate to the user's end device. The user needs only to subscribe a certain type or version of agent, who realises the user's preferred look and feel. Thus, this new service-provisioning paradigm can be identified not only as 'service on demand', but also as 'look and feel on demand'. Using an agent-enabled system, agents can represent almost any party of the system. Some old application processing scenarios can be executed in a more effective and flexible way. In particular, this is valid for new applications in the very promising area of electronic commerce and information retrieval. With the help of agents, it is possible to build these applications easily in an asynchronous way.

II. THE PERFORMANCE MODEL

The purpose of this paper is to derive a performance model that shows the efficiency of mobile agents in several scenarios. In a comparison with the conventional interaction model, Remote Procedure Calls (RPC), the efficiency of mobile agents is discussed in general. To evaluate the performance aspects of using agent technology to implement advanced telecommunication services, e.g. information retrieval services, an investigation comparing Agent Migration, RPC and Messaging has been performed. For a quantitative comparison the selected agent platform and respective protocols as well as specific services have to be considered. As a result it is proposed to use an optimisation strategy that decides for each step whether the agent should migrate. Agent platforms support this kind of dynamic behaviour, where the location of code execution can be chosen dynamically. The decision is based on the amount of data to be transferred and the overall time needed to perform a given task. The optimisation can be based on a cost function, which combines these two basic parts, depending on the priorities for a specific scenario.

Two scenarios are considered in this paper. The first is a homebanking service where only two locations are involved: the mobile user and the banking server. In the second scenario, the information retrieval scenario, a variable number of locations is involved, so that the dynamic optimisation is evaluated for this scenario.

III. HOMEBANKING SCENARIO

A test scenario is derived, with different numbers of bank transactions and given network characteristics. The calculations are made with billing unit durations of 60 seconds, 10 seconds and 1 second. Whilst varying the number of remittances per interaction, the number of telephone units spent are calculated. The performance study is based on the agent migration protocol of the Voyager agent platform [4] and the RPC used by the dialog-based HBCI homebanking protocol [5]. The amount of transmitted data is the basis for the calculation of the billable time. Figure 1 shows a measurement of the transmitted data amount by money remittances using a homebanking agent.

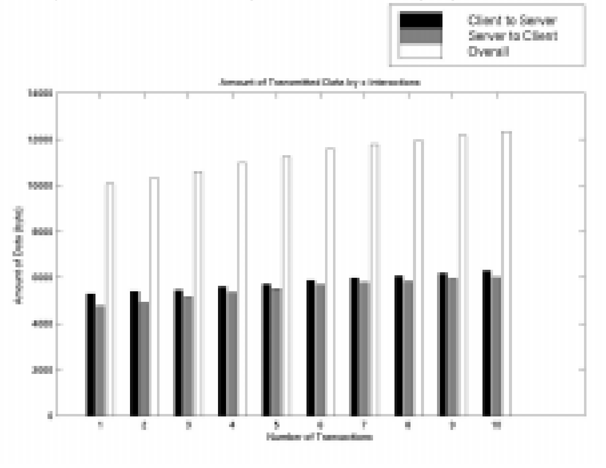


Figure 1 – Measurement: Transmitted Data using Mobile Agents

The transmitted data amount increases slightly with the number of transactions. However, the user has to pay not only for the time it takes to transmit the data.

The establishment of a TCP/IP connection between client and server takes some time as well. This time will also be charged, as the user pays for the GSM connection and not for the TCP/IP connection. Figure 2 shows the billable time for the homebanking agent by money remittances. In figures 1 and 2 you can see that the data amount and the billable time does not increase significantly if the number of transactions is increasing.

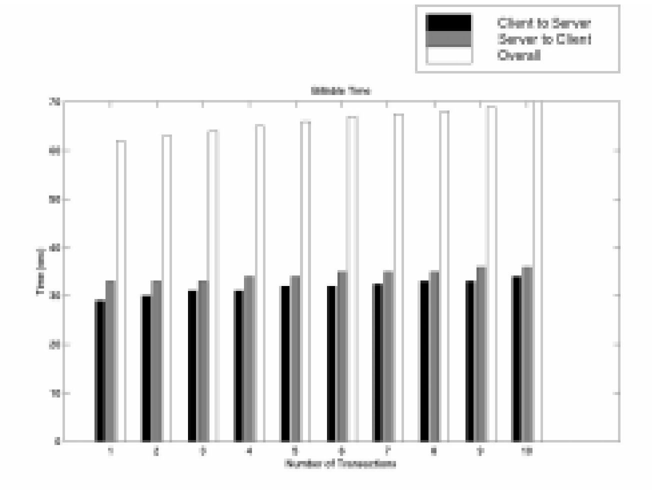


Figure 2 – Measurement: Billable Time of Homebanking Agent

In figure 3 the transmitted data for the mobile agent based application is compared to the corresponding data amount using RPC:

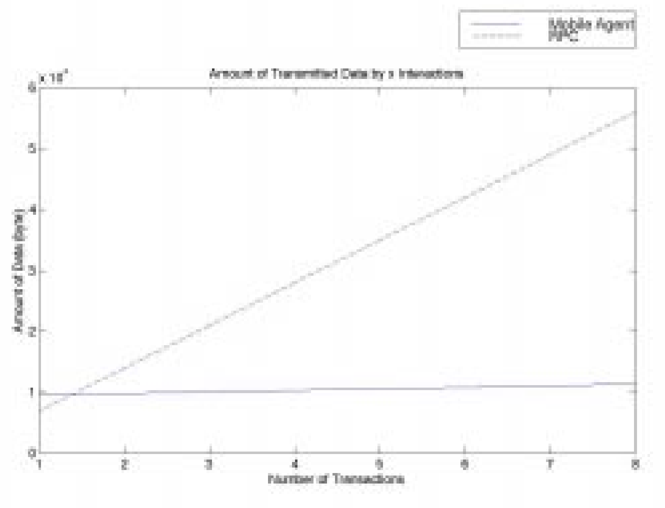


Figure 3 – Calculation: Transmitted Data

The data amount by using RPC increases considerably compared to the respective homebanking agent data amount. This will effect the billable time accordingly. The homebanking agent connects to the server twice, once before the processing and once after the processing. The processing time will therefore not be charged using the homebanking agent. Using RPC the customer will stay online during the processing of the transactions, and this influences the billable time substantially.

In figure 4 a comparison between the homebanking agent and RPC is made based on billable time. Using RPC the billable time increases linearly, whilst the billing time for the homebanking agent stays close to constant. Please note that the billable time results from two calls using the mobile agent, but only from one call using RPC. This is because the connection between the client and the server is interrupted after transferring the agent. For transmitting the result a new connection has to be established. This effect must be considered when calculating the costs.

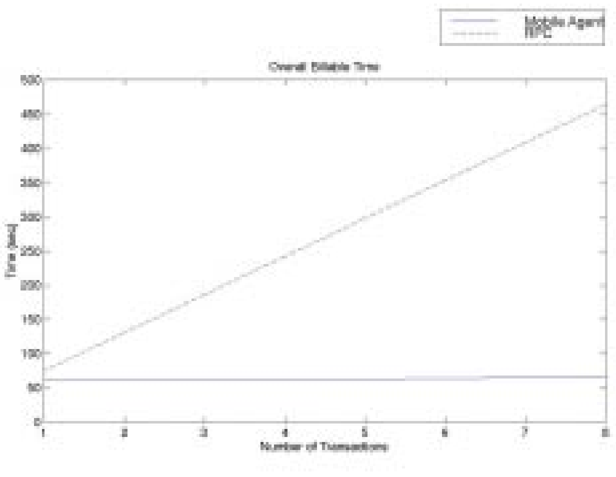


Figure 4 – Calculation: Billable Time Comparison

Based on the billable time the costs for the end user using the homebanking agent can be calculated. The billable time consists of the terms (1) and (2):

$$T_{mig} = T_c + \frac{B_{mig}}{\mu_{eff}} \tag{1}$$

$$T_{rep} = T_c + \frac{B_{rep}}{\mu_{eff}} \tag{2}$$

The costs using the mobile agent is then calculated in term (3):

$$C_A = \left\lceil \frac{T_{mig}}{T_{unit}} \right\rceil U_{price} + \left\lceil \frac{T_{rep}}{T_{unit}} \right\rceil U_{price} \tag{3}$$

By combining (1), (2) and (3) the costs for using the homebanking agent is given in (4):

$$C_A = \left\lceil \frac{\mu_{eff} * T_c + B_{mig}}{T_{unit} * \mu_{eff}} \right\rceil U_{price} + \left\lceil \frac{\mu_{eff} * T_c + B_{rep}}{T_{unit} * \mu_{eff}} \right\rceil U_{price} \tag{4}$$

T_c	the time needed for setting up a TCP/IP connection
B_{mig}	transmitted data amount from client to server
B_{rep}	transmitted data amount from server to client
μ_{eff}	processing rate
T_{unit}	billing unit duration

It is worth noticing that, generally, the execution time in a server does not affect the costs by using a mobile agent. If the execution time for some reason should be long, the cost factor in any system using public telephone networks could be reduced significantly by using a mobile agent.

In figure 5, a comparison regarding the costs is made based on a unit duration of 60 seconds. For these units the costs of using the homebanking agent were constant by two billing units, one unit for each initiated data transfer. The costs of using RPC rise linearly with the number of remittances. The RPC costs were at no point lower than the costs for the usage of a mobile agent.

For a unit duration of 10 seconds the price difference gets slightly bigger. The RPC costs increased linearly also here, while the costs for the mobile agent were close to constant on a low level. For one second units the billable time by using the mobile agent was reduced by about 58% compared to one minute units, which reduces the costs accordingly. The billable time by using RPC was reduced by 14%-25%.

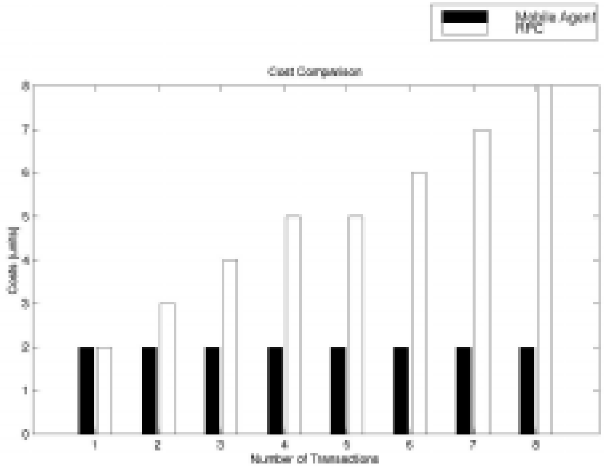


Figure 5 – Calculation: Cost Comparison. Billing unit duration: 60 seconds

An application based on mobile agents offers a great deal of flexibility, and the homebanking scenario is a good example for that. Sometimes, for instance during busy office hours, the customer wants to spend as little time as possible interacting with the bank. Then the possibility of using a mobile agent is suitable. The bank transactions are given to the homebanking agent, the agent is sent off to the bank, and the customer can regard the transactions as done. In this case this is convenient for the customer because he only needs to prepare the transactions and give them to the agent.

After the homebanking agent has been transmitted, the user does not have to wait during the queuing and execution in the bank. To be sure that everything has been accomplished to his satisfaction, he will have to connect to the bank again to get a confirmation. The customer can do this whenever he wants to, for example after office hours when he has time enough and the billing units are cheaper. In this case, however, it is still possible that something has gone wrong and that the answer from the homebanking agent consists of an error message.

Home Banking Scenario in GPRS

Using a mobile agent-based application in the traditional GSM network, the user spends unnecessary long time waiting for a GSM connection and a TCP/IP connection. Even if the cost comparison scenario shows that the agent-based alternative is cost efficient compared to RPC, it is clear that an alternative without log-on time and connection times could be even more cost and time efficient. In contrast to the situation in a circuit switched system, the GPRS user does not pay for the online time but for the actual data transfer. With GPRS it will be possible to stay virtually online without sending or receiving data, as the only chargeable component is the amount of transmitted data.

Additionally, the connecting time from the standby state to the active state is typically less than one second, which leads to two clear performance advantages for GPRS. The time a user would have to wait before the data transmission can begin is reduced drastically and the user will not have to pay for the time it takes to set up a TCP/IP connection.

However, the data transfer time in GPRS can only be estimated, not predicted. In a packet-oriented communication system the network load and the quality of service (QoS) are important factors, which are both neither constant nor completely predictable.

In addition, the packet-oriented nature of the GPRS system makes it possible to differentiate the quality of service. The different packets can be treated differently, and the customers can be divided into different priority classes. The QoS profile of a data unit in the GPRS system is considered to define the expected quality of service in terms of the following attributes [9]: precedence, delay, reliability, peak throughput, mean throughput class and radio priority.

The GPRS user will have the possibility to choose from four different classes of priority. The priority class chosen by the user corresponds to the radio priority on the air interface. This is the only QoS parameter which is stored in the HLR. The chosen priority class decides the precision and speed of the data processing for uplink as well as for downlink.

However, the mapping of user application parameters to GPRS QoS parameters is an implementation issue and is not part of the GPRS specifications. In order to do a comparison between the mobile agent performance in a conventional GSM network and the expected performance in a GPRS network, a simplified model based on the quality of service criteria regarding throughput in GPRS is developed. The mean throughput specifies the average rate at which data is expected to be transferred across the GPRS network. The network may limit the subscriber to the negotiated mean data rate, even if additional transmission capacity is available. The mean throughput class is defined as shown in table 1 [9].

Mean Throughput Class	Mean Throughput in octets per hour
1	Best effort.
2	100 (~0.22 bit/s).
3	200 (~0.44 bit/s).
4	500 (~1.11 bit/s).
5	1 000 (~2.2 bit/s).
7	5 000 (~11.1 bit/s).
10	50 000 (~111 bit/s).
11 .. 19	100 000 .. 50 000 000

Table 1 - Mean Throughput Classes

The peak throughput is defined as shown in table 2 [9].

Peak Throughput Class	Peak Throughput in octets per second
1	Up to 1 000 (8 kbit/s).
3	Up to 4 000 (32 kbit/s).
5	Up to 16 000 (128 kbit/s).
7	Up to 64 000 (512 kbit/s).
9	Up to 256 000 (2 048 kbit/s).

Table 2 - Peak Throughput Classes

In order to allocate the throughput classes to a radio priority, the following settings are used in this model:

Radio Priority	1	2	3	4
Peak Throughput Class	7	5	3	1
Mean Throughput Class	10	7	5	2

Table 3 - Radio Priority Allocation

If a GPRS node is seen as a system with a Markovian interarrival rate, the delay can be seen as the mean sojourn time S(μ,ρ). The mean sojourn time can be calculated with the Pollaczek-Khintchine formula:

$$S(\mu,\rho)=\frac{1}{\mu}+\frac{\rho(1+C_b^2)}{2\mu(1-\rho)} \tag{5}$$

If the GPRS node is seen as an M/M/1 queuing model the coefficient of variance (C_b) equals 1. Term (5) is used as a basis for calculations regarding mean and peak throughput according to table 1 and 2.

To compare the circuit switched homebanking scenario with the GPRS homebanking scenario the following example is chosen:

- execution of two money remittances with HBCI
- the packets have to pass three nodes
- all GPRS nodes and the air interface have the same network load

The transmitted data for two money remittances must be split into packets (packet size: 1500 byte).

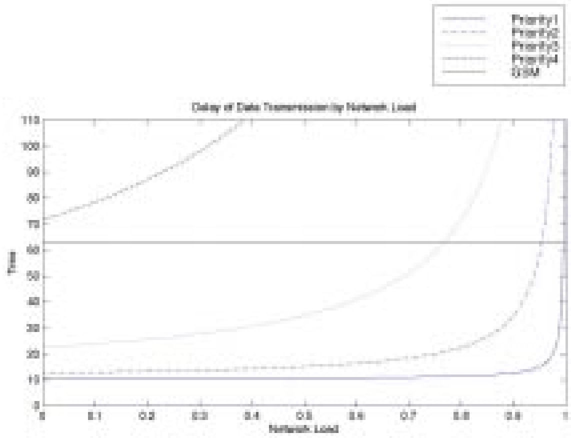
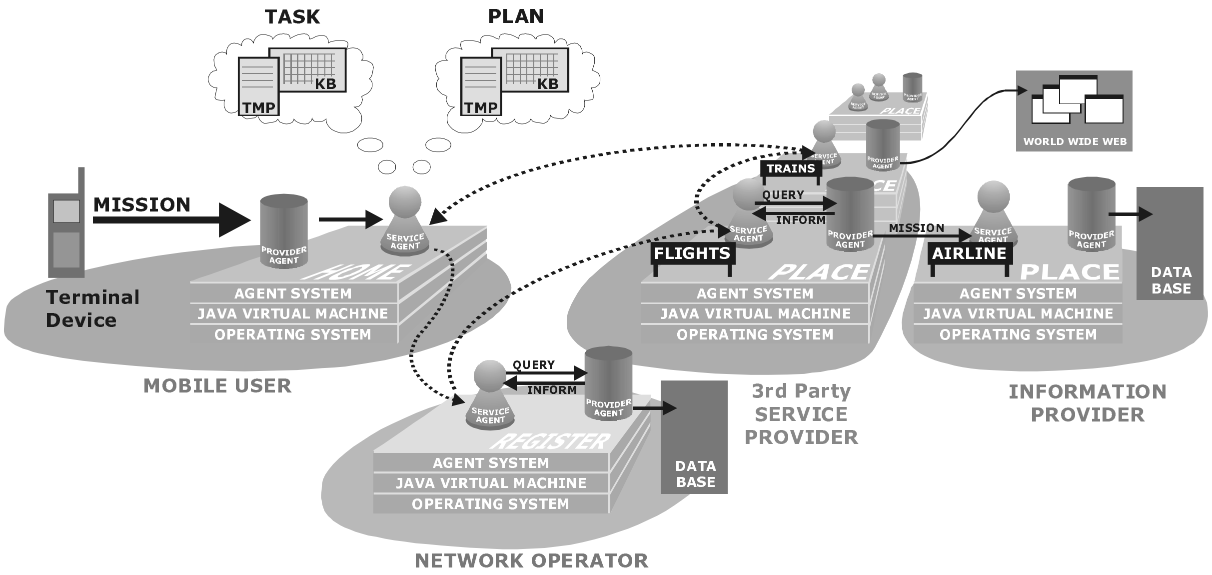


Figure 6 - Comparison, agent-based homebanking in GPRS – agent-based homebanking in conventional GSM

The billable time for this example in the traditional GSM scenario is 63 seconds (see figure 2), independent of the network load. This is valid if the user gets a free time slot, which should always be the case.

Based on the performance model for agent-based applications in a circuit switched environment and the simplified model for GPRS, a comparison of the two homebanking scenarios for the chosen example of two money remittances can be derived as shown in figure 6.

In this example the three higher GPRS priorities are significantly better alternatives than the circuit-switched solution. The network load on the GPRS nodes must be about 75% if priority level 3 should break even with the GSM solution, which is implausible. Please observe that this comparison is valid for this example only.



Example User Query:
Find cheapest connections to London on October 24th before 11.00 a.m.

Figure 7 – Architecture of the Information Retrieval System

IV. INFORMATION RETRIEVAL SCENARIO

The following section discusses simulation results for the information retrieval scenario, which was also implemented on Voyager [2]. The implementation is an agent-based service offering the user an efficient retrieval service for information, that is distributed over different platforms in heterogeneous networks. An example is the automatic booking of flights or other travel arrangements in the Internet. The user delegates certain functions, e.g., find a flight at a date and time to a certain city, without having to know about details of the task execution. Any information (e.g., WWW pages, LDAP databases, etc.) with data important for the user can be accessed. The mobile agent has a certain degree of autonomy and collects the information and returns with an evaluated result back to the current terminal of the user, see figure 7.

The performance study is based on the messaging and agent migration protocol of Voyager and Java RMI and the parameters of the implementation, e.g., the size of an agent. The Voyager protocols were studied in detail. Based on our analysis of the different Voyager communication protocols in form of message sequence charts [6] formulas were derived for RMI, asynchronous messaging and agent migration in relation to both time elapsed and data transferred:

$$B_{rmi} = B_{proxy} + I * (B_{req} + B_{rep})$$

$$B_{mig} = B_{proxy} + B_{movereq} + B_{overhead} * P + B_{data} + B_{state} + B_{code} * P + B_{ref} + B_{rep} * \alpha$$

$$B_{mig(VCT)} = B_{movereq} + B_{proxies} * P + B_{data} + B_{state} + B_{code} * P + B_{ref} + B_{rep} * \alpha$$

$$T_{rmi} = T_{proxy} + 2 * I * \delta + (B_{rmi} - B_{proxy}) * (u + 1/net)$$

$$T_{mig} = T_{proxy} + 4 * \delta * (1 + P) + (B_{mig} - B_{proxy}) * (u + 1/net)$$

$$T_{mig(VCT)} = 2 * \delta + 14 * \delta * P + B_{mig(VCT)} * (u + 1/net)$$

Notation:

$\mathbf{B}_{rmi}$	transmitted data amount to accomplish a task by RMI
$\mathbf{B}_{mig}$	transmitted data amount by agent migration of Voyager
$\mathbf{B}_{mig(VCT)}$	transmitted data amount by agent migration (VCT)
$\mathbf{T}_{rmi}$	time to accomplish a task by RMI
$\mathbf{T}_{mig}$	time to accomplish a task by agent migration
$\mathbf{T}_{mig(VCT)}$	time to accomplish a task by agent migration (ver. VCT)
$\mathbf{B}_{proxy}$	transmitted data amount for fetching a proxy
$\mathbf{B}_{proxies}$	data amount of proxies for dynamic aggregation
$\mathbf{B}_{movereq}$	transmitted data amount for the migration request
$\mathbf{B}_{data/state}$	transmitted data amount for agent data/state
$\mathbf{B}_{code}$	data amount for agent code
$\mathbf{B}_{ref}$	data amount for agent reference
$\mathbf{B}_{req}$	data amount for request
$\mathbf{B}_{rep}$	data amount for actions (call back function) result
α	the selectivity factor
δ	net delay
u	the marshalling rate
net	effective bandwidth of the network

Figures 8 and 9 show the mean amount of data to be transferred and the mean time needed in dependence of the agent data. In this example pure agent migration and pure RMI are compared with optimisation methods. Based on the implementation the total size of the agent classes was set to 25 Kbyte. We further need to set parameters related to the task of the agent. For every task it is assumed that 1 Kbyte of agent data are added. The maximum number of hosts is 200. The number of subtasks per task is set to 30 and maximum number of interactions per subtask is also set to 30. Table 4 summarises all parameters. Based on the formulas given above simulations of the scenario were performed to evaluate these mean values and compare them with different optimisations.

agent code	25000 Byte	proxy	5000 Byte
agent state	500 Byte	hosts	200
subtasks	30	migration request	500 Byte
interactions	30	net delay	30 ms
bit rate	30 Kbyte/s	marshalling rate	200 Kbyte/s
max. request	200 Byte	max. results	2000 Byte
min. request	20 Byte	min. results	20 Byte

Table 4: Parameters (note: the agent enlarges its knowledge base during its life. It is assumed, that 1 Kbyte is added for every task.)

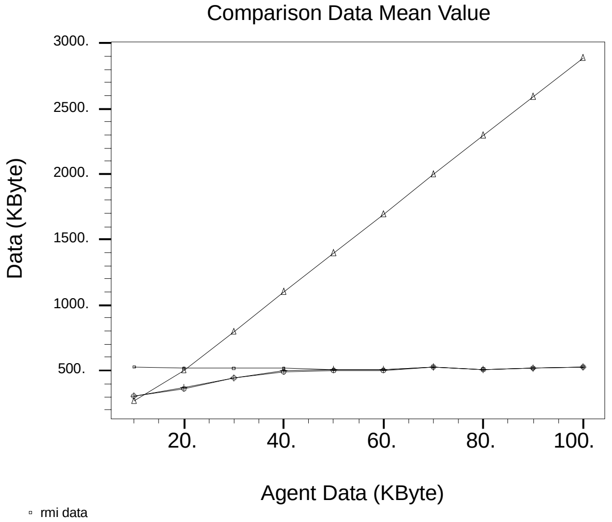


Figure 8: Comparison of mean data for RMI, Agent Migration and Optimisation

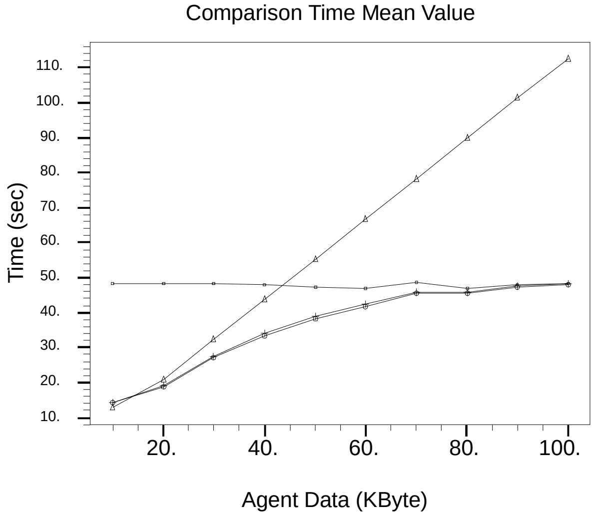


Figure 9: Comparison of mean time for RMI, Agent Migration and Optimisation

For practical purposes only a one-step optimisation should be considered for implementation. Optimisation with more look-ahead is not feasible in practise as the next destination of an agent can be determined dynamically. And as can be seen in the example it does not achieve a better result for this scenario. The results also show that the optimisation depends on the optimisation function, where data and time are the basic variables which can be combined in a cost function similar to the banking example depending on the networks and tariffs involved.

The tendency towards agent migration depends on the size of the agent, the agent data, the request, the reply, the amount of interactions and the selectivity of the agent, which is one of the main advantages of sending the code to the data, filtering the reply and sending only the selected answer back. Other major factors are the frequency of service usage and the amount of locations involved, because the code of an agent only has to migrate once to a new location, after that only the data/state has to migrate.

V. CONCLUSION

An important achievement of an agent system is that the network load is significantly reduced. This has been proven with the agent-based homebanking solution using the new German homebanking standard HBCI.

The comparison between the homebanking agent and RPC shows obvious advantages for the homebanking agent concerning amount of transmitted data and billable time. Another clear advantage shown by the homebanking service is the fact that the execution time in a server does not affect the costs by using a mobile agent. If an agent has sufficient intelligence and information for fulfilling its task, the user can send the agent to the server and consider the task as solved.

For the information retrieval scenario the performance depends on the specific task and amount of gathered information. It was shown here that all methods have their application areas and by choosing the optimal method in each case the best of both worlds is obtained.

In the framework of the ACTS CAMELEON [7,8] project the aspects mentioned above are being investigated to evaluate different UMTS/VHE models using agent technology.

Mobile agent-based applications in GPRS will be able to provide the GSM customers with even faster data services in the future. All connecting and log-in times will be eliminated, which gives GPRS a clear performance advantage when it comes to the time a customer spends using an application. Additionally, the possibility to stay online and reachable without being charged is a clear advantage in the GPRS system.

The above mentioned GPRS advantages comply with the use of mobile agents in GPRS, which offers rewarding possibilities for mobile agent-based applications in the future.

VI. REFERENCES

[1] Song, Wei. Requirement Analysis, Concept and Implementation of an Agent based Homebanking System, Master Thesis, Communication Networks, University of Technology Aachen, August 1998.

[2] Döll, Meik Andreas. Requirement Analysis, Concept and Realisation of an Agent based Platform-Independent Information Retrieval System, Master Thesis, Communication Networks, University of Technology Aachen, October 1998.

[3] Hartmann, Jens; Görg, Carmelita; Farjami, Peyman. Agent Technology for the UMTS VHE Concept. ACM/IEEE MobiCom'98, Workshop on Wireless Mobile Multimedia, Dallas, United States, October 1998.

[4] http://www.objectspace.com/products/voyager/

[5] Hartmann, Jens; Song, Wei. Agent Technology for Future Mobile Networks. Second Annual UCSD Conference on Wireless Communications in co-operation with the IEEE Communications Society, San Diego, USA, March 1999.

[6] Long, Hai. Performance Analysis of Mobile Agents for Future Telecommunication Services, Master Thesis, Communication Networks, University of Technology Aachen, February 1999.

[7] http://www.comnets.rwth-aachen.de/~cameleon

[8] http://www.fokus.gmd.de/cc/ima/climate/

[9] Evensen, Rune. Performance evaluation of agent-based applications for the cellular GSM mobile radio network, Master Thesis, Communication Networks, University of Technology Aachen, May 1999.