

Mobile agents for personalized information retrieval: When are they a good idea?

Ravi Jain and Farooq Anjum
Applied Research
Telcordia Technologies
Morristown, NJ
{rjain,fanjum}@telcordia.com

Abstract—Mobile agent technology has been proposed as an alternative to traditional client-server computing for personalized information retrieval by mobile and wireless users from fixed wired servers. In this preliminary paper we develop a very simplified analytical model that examines the claimed performance benefits of mobile agents over client-server computing for a mobile information retrieval scenario. Our evaluation of this simple model shows that mobile agents are not necessarily better than client-server calls in terms of average response times; they are only beneficial if the space overhead of the mobile agent code is not too large or if the wireless link connecting the mobile user to the fixed servers of the virtual enterprise is error-prone. We quantify the tradeoffs involved for a variety of scenarios and point out issues for further research.

I. INTRODUCTION

There has been tremendous interest in the past few years in using mobile agent technology for automating the task of retrieving, organizing and filtering information located at widely dispersed sites [5], [3]. Mobile agent systems have been built and demonstrated that indicate mobile agents can be used for information retrieval activities. It has been argued that mobile agents are particularly suitable for supporting mobile and wireless users, who may be subject to frequent voluntary or involuntary disconnections from the network and may be operating using mobile devices with limited resources and communicating over links of limited bandwidths and potentially high costs [5], [2], [3]. In that case, it may be beneficial to have the mobile user simply launch a mobile agent on the fixed network, and then disconnect, where the mobile agent performs a search for the desired information and retrieves it, rather than having the mobile user stay continuously connected.

In this paper, we consider the following question: is the mobile agent paradigm “better” than using traditional client-server computing for information retrieval tasks? Of course, a question that immediately arises is, what is a mobile agent [4]? As a working definition, in this paper a *mobile agent* consists of a self-contained piece of software that can migrate and execute on different machines in a dynamic networked environment, and that senses and (re)acts autonomously and proactively in this environment to realize a set of goals or tasks.

Most justifications for using mobile agents [1], [2] can be divided into two broad categories, which we call *performance benefits* and *software engineering benefits*. The performance benefits of mobile agents include reduction in network bandwidth consumption, reduced latency, reduced computation, and increased fault-tolerance. The software engineering benefits state that the mobile agent paradigm can help application programmers and designers conceptualize solutions better in that

the paradigm may be more naturally suited to certain types of applications, that they can help improve code modularity and reusability, that they can help hide network, system and protocol heterogeneity, etc. In this preliminary paper we focus on quantifying the performance benefits of mobile agents over traditional client-server computing information retrieval tasks.

In the following section we describe a very simple preliminary analytical model for comparing the performance benefits of using mobile agents rather than client-server computing. In sec. III we describe the results of evaluating that model for example scenarios, and in sec. IV we end with some conclusions.

II. PERFORMANCE MODEL

Consider a user accessing information from Web servers using a network where the last hop (from base stations to the user) is wireless. See Fig. 1 for a sketch of the system architecture. For simplicity in this paper we assume the user is not mobile. The information-access application the user is enjoying can take two approaches. In the *client-server* approach the client repeatedly issues method calls or sends messages (*queries*) to the server(s) and obtains *results*. In the *mobile agent* approach the user’s device launches an agent that contains the queries, visits the server(s) and returns with results.

The model has the following parameters and assumptions. There are N servers to be searched for information, known a priori. We assume that the errors over the wireless link result in packets being lost with probability p . (Note that the packet loss rate can be derived from the more commonly used bit error rate in a straightforward manner given the specifics of the link and the system.) We assume that the size of a client query message in bytes is unity; the size of the agent is then M . Agents especially can be helpful if they summarize and filter results before returning to the client. To model this the parameter R is the size of the result returned in the client-server approach compared to that returned by the agent; when $R = 1$ the agent performs no filtering, and when $R > 1$ the agent does.

The probability of succeeding in finding the relevant information at a server i is p_i whereas the time taken to process the query at server i is denoted by t_i . The latency over the wireless link from the mobile user to the basestation is denoted by l_m while the latency of the link from the basestation to the i th server is denoted by $l_{0,i}$. The latency of the link from the i th server to the $i + 1$ th server is denoted by $l_{i,i+1}$. Let \hat{l}_{0i} denote the sum of the latencies due to the wireless link and the wireline link between the i th site and the basestation. Regard all delay parameters as

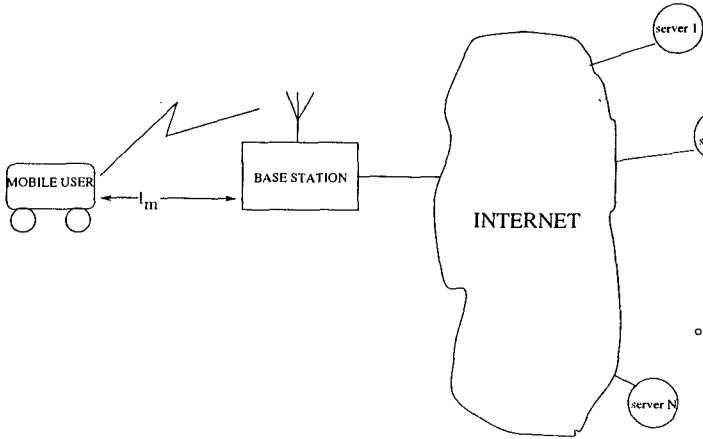


Fig. 1. System model for wireless information retrieval

means.

Then the expression for the average delay D_{cs} in finding the information using client-server calls is given by

$$\begin{aligned}
 D_{cs} &= p_1(2\hat{l}_{01} + t_1) + p_2(2\hat{l}_{01} + t_1 + 2\hat{l}_{02} + t_2) + \dots \\
 &\quad + p_k(2\hat{l}_{01} + t_1 + 2\hat{l}_{02} + t_2 + \dots + 2\hat{l}_{0k} + t_k) + \dots \\
 &\quad + p_N(2\hat{l}_{01} + t_1 + \dots + 2\hat{l}_{0N} + t_N) \\
 &= \sum_{j=1}^N p_j(2\hat{l}_{01} + t_1 + 2\hat{l}_{02} + t_2 + \dots + 2\hat{l}_{0j} + t_j) \quad (1)
 \end{aligned}$$

while in case of mobile agents the corresponding expression will be

$$\begin{aligned}
 D_{ma} &= p_1(2\hat{l}_{01} + t_1) + p_2(\hat{l}_{01} + t_1 + l_{12} + t_2 + \hat{l}_{02}) + \dots + \\
 &\quad p_k(\hat{l}_{01} + t_1 + l_{12} + t_2 + \dots + l_{(k-1)k} + t_k + \hat{l}_{0k}) + \\
 &\quad \dots + p_N(\hat{l}_{01} + t_1 + \dots + t_N + \hat{l}_{0N}) \\
 &= \sum_{j=1}^N p_j(\hat{l}_{01} + t_1 + l_{12} + t_2 + \dots + l_{(j-1)j} + t_j + \hat{l}_{0j})
 \end{aligned}$$

where

$$\hat{l}_{0i} = l_{0i} + \frac{2pl_m}{1-p} + l_m \quad (3)$$

The last equation incorporates the effect of message retransmissions over the wireless link, where it is assumed that a sender knows within one round-trip time whether a packet needs to be retransmitted.

III. MODEL EVALUATION

We evaluate the model above for a number of information retrieval scenarios, each of which can be implemented using a client-server or mobile agent approach. In the first scenario (*ALL*), all N predetermined server sites are searched, and all information thus obtained is returned. Thus in terms of the model above, $p_i = 0$ for all i , except $p_N = 1$. In the second scenario, the servers are searched sequentially until the first server with

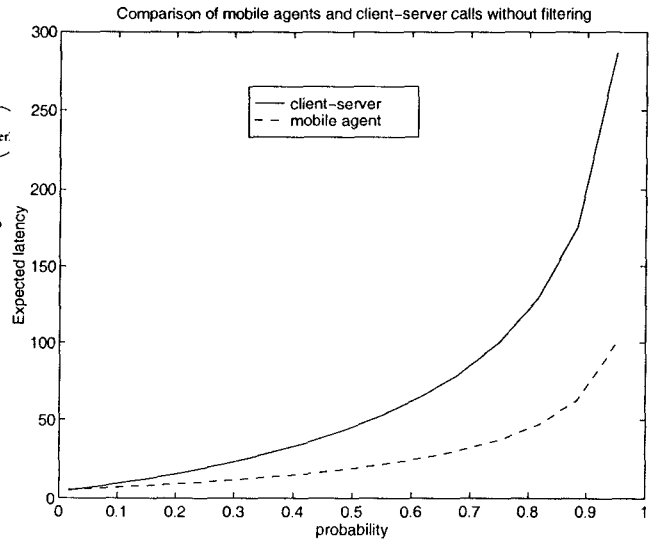


Fig. 2. Visiting all 15 servers with no need for filtering by mobile agents

the required information is found, at which point the search is halted. It is assumed that the information is *sure* to be found on one of the N servers, i.e., the sum of the success p_i is unity. The servers are searched in descending order of the success probability p_i , which is assumed to be known a priori; this scenario is called *SURE-DECREASING*. The third scenario is like the second, except it does not assume that the p_i are known. Thus the servers are searched in random order of success probability; this scenario is known as *SURE-RANDOM*. Finally, in the fourth and fifth scenarios we relax the assumption that the information is sure to exist on one of the N servers. In fact we assume the servers are unrelated, and so the scenarios are like *SURE-DECREASING* except that the sum of p_i need not equal unity. In the fourth scenario, *MAYBE-LARGE*, $p_i \leq 1$; in the fifth scenario, *MAYBE-SMALL*, $p_i \leq 0.1$.

For each scenario we compare the use of client-server (CS) and mobile agent (MA) approaches. For simplicity each of the parameters $p_i, l_{i,i+1}$ is assumed to be independently distributed (2) between 0 and 1.

Scenario: ALL. We assume $N = 15$ sites are all visited and mobile agents are relatively large, with $M = 5$. In Figure 2 we assume the size of the result is comparable to the query, i.e., $R = 1$. For low packet loss probabilities on the wireless links mobile agents result in greater expected latency for the information retrieval than client-server calls, primarily due to the agent size. As the link quality deteriorates mobile agents become more attractive as they avoid repeated client-server calls over the wireless link. Figure 3 is the same scenario with $R = 5$; observe that as the result size increases the expected latency for CS goes up dramatically, while for MA it is assumed to remain unchanged due to filtering by the agent.

SURE-DECREASING. Assuming that the servers are visited in order of decreasing success probability, and that agent sizes are small, Figure 4 shows that for $N = 15, M = 1, R = 1$, as the wireless link deteriorates mobile agents are far more attractive than client-server calls. Comparing with Figure 2, we see that searching the sites in decreasing order of success probabil-

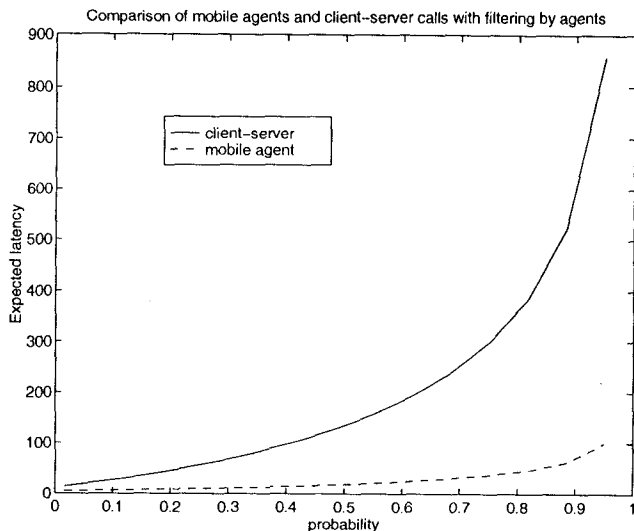


Fig. 3. Visiting all 15 servers with filtering done by mobile agents

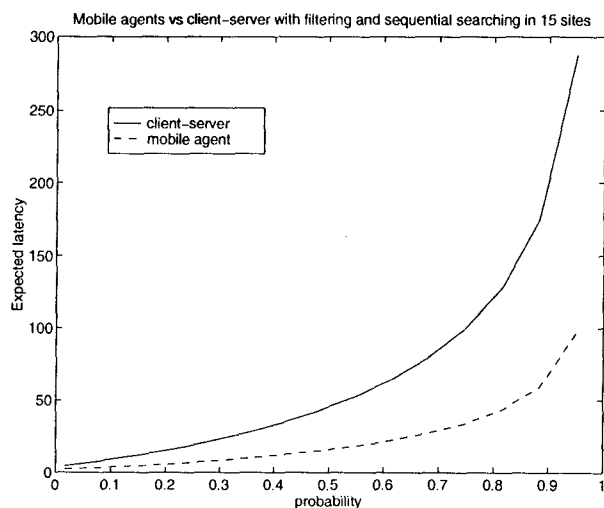


Fig. 5. Effects of wireless losses over mobile agent and client-server paradigms with 10 servers searched in sequential order

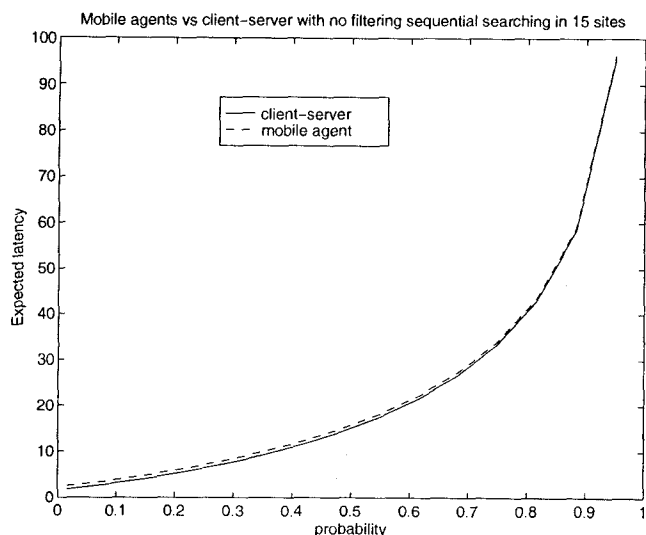


Fig. 4. Effects of wireless losses over mobile agent and client-server paradigms with 10 servers searched in sequential order

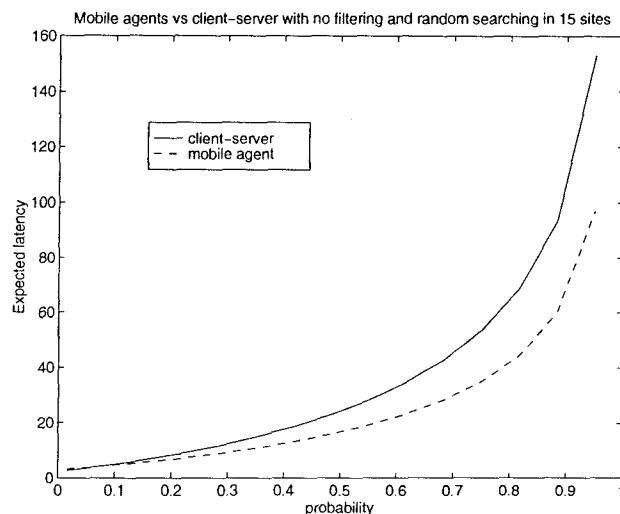


Fig. 6. Effects of wireless losses over mobile agent and client-server paradigms with 10 servers searched in random order

ity reduces the expected latency for CS significantly, and eliminates the advantage of MA since CS makes few traversals over the wireless link. However, comparing Figure 4 with Figure 5, where $R = 5$, shows that if result filtering is necessary MA is still substantially faster.

SURE-RANDOM. Clearly knowing success probabilities a priori is not likely. In Figure 6 we consider searching $N = 15$ sites in random order of success probability and also assume $R = 1$. Comparing with Figure 4 we see that searching the sites in random order generally increases the expected latency for CS but not MA, making MA more attractive in this situation.

MAYBE-LARGE and MAYBE-SMALL. If the servers are unrelated the success probabilities p_i lie between 0 and 1 and need not sum to unity. In Figure 7 we assume $N = 15$, $M = 5$, $R = 1$, and sites are searched in order of decreasing success probability, which is known a priori. Comparing with Figure 4 we see that the high success probabilities decrease the expected

latency for both MA and CS significantly; however, for large agent sizes the reduction for MA is smaller, i.e., CS is preferable to MA. On the other hand, comparing with Figure 8, which is identical except that success probabilities $p_i \leq 0.1$ shows that with low success probabilities, CS suffers compared to MA as it makes more traversals over the wireless link.

IV. CONCLUSIONS

We have developed a very simplified analytical model to quantify the performance benefits of using mobile agent technology rather than traditional client-server techniques for retrieving information on behalf of a mobile user. We have evaluated the model for an example scenario, focusing on the response time to an information retrieval query as the metric of interest. The results show that it is not clear that mobile agent are always preferable to client-server computing, or vice versa. In particular, if, for a given query, the code size of the mobile agent

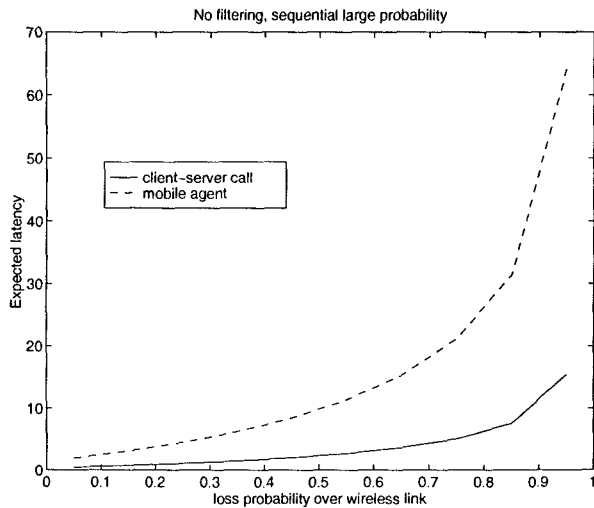


Fig. 7. Illustrating that use of mobile agents is disadvantageous for a scenario consisting of a fixed set of unrelated sites with large probability of success on each of the servers with servers being searched sequentially in decreasing probability

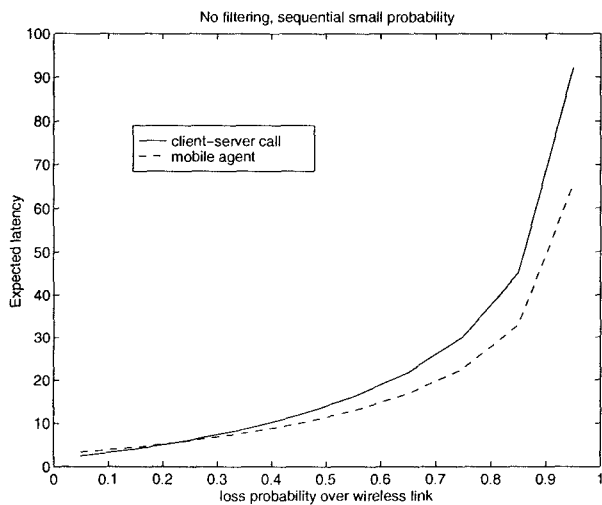


Fig. 8. Illustrating that use of mobile agents is advantageous for a scenario consisting of a fixed set of unrelated sites with small probability of success on each of the servers with servers being searched sequentially in decreasing probability

exceeds that of the corresponding client-server request message, the response time using mobile agents can be significantly more than using client-server calls. On the other hand, if the user has a wireless link connecting the client device to the fixed wired network, the improvement in response time using mobile agents compared to client-server computing increases as the quality of the link decreases.

We are continuing to address the issues raised in this paper, and are developing the simple analytical performance model to improve its accuracy and broaden its scope.

REFERENCES

[1] Harrison C.G. Chess D.M and A. Kershenbaum. *Mobile Agents: Are they a good idea*. IBM Research Report RC 19887, Oct 1994.
 [2] Lange D.B. and Oshima M. Seven good reasons for mobile agents. *Comm. ACM*, 42(3):88-89, March 1999.

[3] Brewington B. et.al. Mobile agents for distributed information retrieval. *Intelligent Information Agents*, Springer-Verlag, pages 355-395, 1999.
 [4] S. Franklin and A. Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. Proc. Third Intl. Workshop on Agent Theories, Architectures and Languages, Springer-Verlag, 1996.
 [5] Comm. of the ACM. *Multiagent Systems on the Net and Agents in E-commerce*, volume 42. Mar 1999.