Tutorial: Mobile Software Agents for Dynamic Routing

Kwindla Hultman Kramer, Nelson Minar, and Pattie Maes
MIT Media Lab Room E15-305
20 Ames Street, Cambridge, MA 02139 USA
http://www.media.mit.edu/~nelson/research/routes/

March 12, 1999

Abstract

As portable digital devices of all kinds proliferate, wireless networks that allow for flexible, timely and efficient data communication become more and more important. Networks for mobile devices are quite difficult to design for several reasons, chief among them the problem of routing packets across networks characterized by constantly changing topology. In this article we describe ways to address the routing problem using a new technique for distributed programming, mobile software agents.

1 The Problem: Building Infrastructure for Highly Dynamic Networks

A wireless network serving a population of frequently mobile nodes presents a moving target to systems designers. Any scheme for managing routing across such a network has to be flexible enough to adapt to continuous and unpredictable change in three fundamental characteristics — overall density, node-to-node topology, and usage patterns. The goal for such a system must be to provide optimal service even as the "rules of the game" change.

The approaches we examine in this paper use three tools to manage this dynamicity: multi-hop routing, decentralized management, and mobile software agents.

1.1 Multi-hop Routing

We assume that a typical data packet will need to move more than one step across the network in order to reach its ultimate destination. This reliance on multi-hop routing makes for efficient use of a key system resource: communications bandwidth. Figure 1.1 illustrates this, comparing the effect of a direct send-and-reply between two nodes (A and B) to the same communication as relayed through intermediates (X and Y). The shaded area (the direct send-and-reply) is obviously much larger than the white area (the multi-hop version). The two nodes at the top of the diagram (I and J) are within transmission range in the direct case, and must be silent during the communication between A and B. In the multi-hop case, however, these two

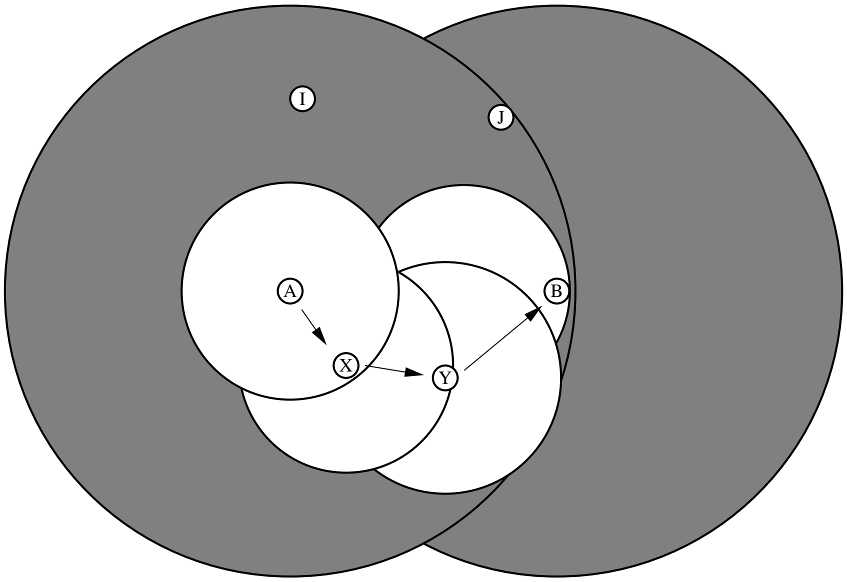


Figure 1: Comparative bandwidth usage of multi-hop routing and direct communication.

nodes are out of range of A, B, X, and Y, and therefore can carry on their own, simultaneous, conversation.

Two more things about this diagram are worth noting. First, the shaded areas can be thought of as radiated power (rather than as occupied bandwidth). Battery life is a limited and valuable resource for most portable devices. The sender and receiver obviously use much less power in the multi-hop case than when communicating directly. Even from a global perspective, counting the power that those two nodes "borrow" from the intermediary, the multi-hop transmission is far more efficient.

Second, this diagram is misleading to the extent that the time domain is not represented. The multi-hop communication takes twice as long as the direct communication, so the geographic localization of bandwidth comes at the expense of some additional use of that same bandwidth over time. This tradeoff of end-to-end time for smaller effected area is an important characteristic of a multi-hop architecture.

1.2 Decentralized Management

Centralized management depends, by definition, on restricting important decisions to one or a few nodes on a given subnetwork. These special nodes become performance and administrative bottlenecks in a dynamic system. If a subnet increases in density or size its decision makers are required to manage an increased load. This may or may not be possible, depending on the circumstances. In addition, a centralized architecture concentrates network vulnerability at the controllers; failure or poor performance on the part of a management node poses serious difficulties for the entire subnetwork dependent on it.

Centralized networks can be partitioned in order to limit load on individual controllers, but each such subdivision requires an increase in resources dedicated to inter-controller coordination. There are practical limits to how small such partitions can become, and diminishing returns as these limits are approached.

In contrast, a fully peer-to-peer architecture offers potential advantages in scalability. We believe that decentralized architectures will be critical to the success of next generation wireless networks. The current explosive growth in cellular telephone, pager, personal digital assistant, and laptop computer usage shows no signs of slowing, and future networks will be so dense and heterogeneous that centralized system designs will prove unworkable.

1.3 Mobile Software Agents

In addition to scaling poorly, centralized network management strategies have difficulty accommodating various or changing patterns of usage. Because the "intelligence" in a centralized network resides in a small number of specialized nodes, most devices on the network will be capable of only a limited range of behaviors. Asking devices with specific, hard-wired functionalities to adjust themselves in response to changing contexts — to use a different back-off algorithm, for example, or to structure their packets in a new way — will usually be impossible. And direct negotiation between two regular nodes that happen to have an interest in a particular kind of communication — perhaps the construction of a low latency dedicated connection, or perhaps just the opposite, the scheduling of a periodic, reliable data ping — is out of the question.

Mobile agents serve as a framework on top of which decentralized infrastructure services can be built. By embedding functionality in mobile software agents and distributing these agents across the network, we push the intelligence traditionally centralized in a few controlling nodes out into the system at large. Every node can be capable of hosting mobile software agents; every node can be a full network citizen. There is no need for privileged arbiter nodes, as network functionality can be built from the ground up by the cooperative behavior of all the individual nodes in the network and that of the agents moving across them.

The following are important characteristics for mobile infrastructure agents:

- Agents encapsulate a thread of execution along with a bundle of code and data. Each agent runs independently of all others, is self-contained from a programmatic perspective, and preserves all of its state when it moves from one network node to another. This is *strong mobility* as defined in [2].
- Any agent can move easily across the network. The underlying infrastructure provides a language-level primitive that an agent can call to move itself to a neighboring node.
- Agents must be small in size. Because there is some cost associated with hosting and transporting an agent, they are designed to be as minimal as possible. Simple agents serve as building blocks for complex aggregate behavior.
- An agent is able to cooperate with other agents in order to perform complex or dynamic tasks. Agents may read from and write to a shared block of memory on each node, and can use this facility both to coordinate with other agents executing on that node and to leave information behind for subsequent visitors.
- An agent is able to identify and use resources specific to any node on which it finds itself. In the approaches described in this paper, the nodes are differentiated only by who their neighbors are and how locally congested the network is. In a more heterogeneous network, certain nodes might have access to particular kinds of information — such as absolute location derived from a global positioning system receiver — that agents could leverage.

2 Research Approaches

Mobile agents have become an active and exciting area of study in recent years, especially as contemporary tools — such as the Java programming language — have made mobile agent systems easier to implement and deploy. Arguments as to why the mobile agents approach is useful in a generalized network context can be found in [7], [6], and [13].

We believe that mobile agents have particularly interesting applications at the network infrastructure layer. The use of simple, cooperative, highly distributed building blocks to organize connectivity makes good technical sense for the networks we are interested in, which are themselves heterogeneous and inherently distributed. Moreover, mobile agents serve as a powerful cognitive tool, allowing system designers to think about complex, macro-level interactions through clear and useful abstractions [8]. We will describe several research efforts using mobile-agent-like approaches to solve network infrastructure problems: active networking, ant based problem solving, and our own work on agent ecologies at the MIT Media Lab.

2.1 Active Networking

In an *active network*, packets are more than just passive data; they are programs executed by the nodes that are passing the packets along. This facility allows packets to make on-the-fly decisions about how to route themselves, and to do more besides. Tennenhouse et al [12] provides a survey of this work.

Active networks research grew out of a community of DARPA-funded scientists interested in new ways to build networks. To these researchers, a transition from passive to active packets seems a natural next step following the evolution from circuit- to packet-switched architectures. Tennenhouse writes that the development of active networking techniques has been motivated by some central problems endemic to today's networks, in particular "the difficulty of integrating new technologies and standards into the shared network infrastructure, poor performance due to redundant operations at several protocol layers, and difficulty accommodating new services in the existing architecture model."

The active networking approach presents a few new and interesting problems of its own. Chief among them is the unresolved issue of how active switches and routers can achieve the raw throughput performance necessary for large scale networking. (Or, more subtlely, how to build new systems such that the gains in efficiency from run time malleability and per packet individualized routing are not washed away by the cost of the more complex hardware required.) History, here, would seem to be on the side of proponents of active networks. Problems that boil down to hardware cost versus flexibility and functionality are often rendered moot by technological trends that bring ever cheaper computational power. More interesting questions, we believe, revolve around issues of ensuring security and scalability in an active packets context.

2.2 Social Insect Metaphors

While the active networks community has concentrated on per packet mechanisms for shuttling data across the network, other researchers have borrowed metaphors from biological systems to develop higher level network management frameworks. The "ant-based," or "social insect," paradigm focuses on achieving cooperative, bottom up results using large populations of simple, autonomous software agents. Of particular importance is a technique for indirect inter-agent communication, called "stigmergy," in which agents leave information caches (which other agents can use) behind on nodes they have visited. Stigmergy, it is argued, serves as a robust, adaptive mechanism for information sharing. Worker ants that leave pheremone trails when they venture outside their nest are using stigmergic communication, and several researchers cite this ant behavior as an explicit influence on their work [11] [5] [3].

An important early paper in this domain is *Mobile Software Agents for Control in Telecommunications Networks* [1]. Appleby and Steward argue that centralized control strategies are a poor match for inherently distributed systems and that the technique of programming with mobile objects is a natural extension of contemporary software engineering principles — specifically of object oriented programming — that allows the construction of sophisticated distributed algorithms. They propose that mobile agents be designed using subsumption architecture techniques borrowed from behavioral roboticists [4], techniques themselves inspired by ethological research.

2.3 Mobile Agent Systems

In our work at the MIT Media Lab we are applying techniques similar to both active networks and social insect research to the problem of managing extremely dense and dynamic wireless networks. Our approach is to use inter-dependent collections of specialized agents that cooperate to provide system services. We have implemented a small software simulator to study the feasibility of using mobile agents as routing infrastructure in such a context. The design of our simulator is motivated by real-world wireless systems being designed and deployed at the Media Lab, which consist of large numbers of small, low power nodes scattered throughout our building. We will summarize our simulation results here; full details are available in our papers [9] [8].

The nodes in our simulations are modeled as low power, relatively short range, radio frequency transceivers distributed randomly throughout a two dimensional space (figure 2.3). The total network diameter is roughly 20 hops. Individual nodes move slowly, following random vectors, so that radio links form and break as the nodes move in and out of range of each other. As a result, the network topology is quite dynamic. (We do assume, however, that physical links are reliable, bidirectional, and easily detectable. Every node knows who its neighbors are.)

There are two kinds of agents in each of our simulated scenarios. Routing agents move around the network, gathering information about the topology of the system and updating routing tables stored on the nodes. Message agents are "smart" data packets; they use those tables to route themselves from a node of origin to a specific destination.

The message agents of our system act somewhat like the active packets in the *capsule* tradition of active networking research; they are minimal programs that carry a payload of communications data across the network [12]. They make their decisions about where they need to go based on information that routing agents, with which they share the network, accumulate and cache. Routing agents explore the network and work cooperatively in order to create and repair routing tables, and in this they resemble the mobile, self directed, collaborative agents of the ant based approach.

Our simulations show that populations of mobile agents are able to cooperatively construct and maintain reasonable routing tables even in contexts where the connections between network nodes are constantly changing. The mobile agents approach requires a significant shift

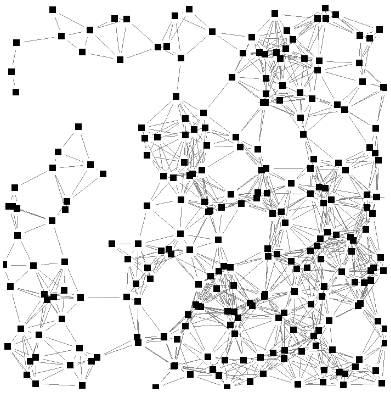


Figure 2: Screenshot of Simulated Network

in how we think about networks, however, and we will have to learn a new set of strategies for this kind of network design. By way of example, we will briefly discuss two lessons for designing successful mobile agent infrastructures that have emerged from our research.

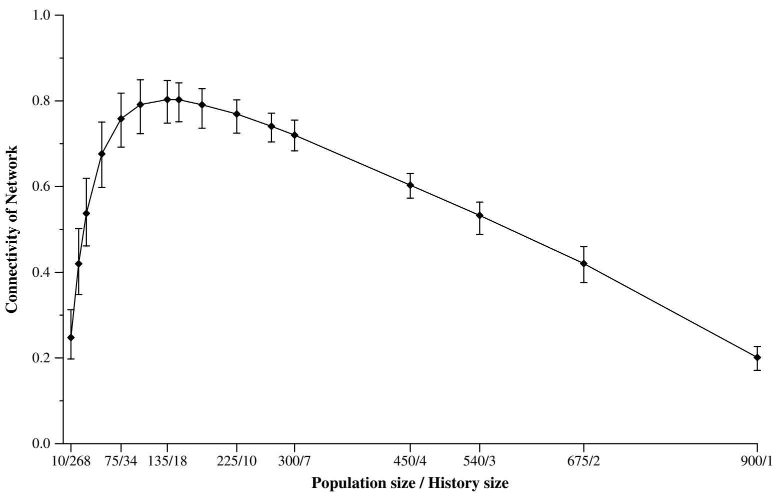
Our first experiments demonstrate the importance of maintaining diversity between agents in a system [9]. We performed three series of simulations for homogeneous populations of three types of routing agents. Each type of agent used a different algorithm to determine its exploratory movement: the first type moved randomly, the second explored using a depth first search strategy based on its own movement history, and the third type pursued a depth first search based on its own knowledge plus that learned from each agent it encountered.

We expected this third strategy to be the most efficient, as it involved the most agent cooperation (and therefore each agent was factoring a larger store of information into each of its decisions). But we found that this was not the case. The aggressively cooperative algorithm performed marginally worse than the simpler independent depth first search. Further examination showed that as the aggressively cooperative agents shared information with one another they converged on homogeneous exploration strategies. All agents were using the same depth first algorithm, so when they operated from identical stores of knowledge they moved in synchrony around the network. The simpler strategy of relying on individually gathered information ensured that agents made diverse search decisions, which made the system as a whole more efficient.

In our second experiments [8], we studied the tradeoff between overall system performance, as measured by how quickly and efficiently message agents were able to traverse the network, and the resource overhead that the routing agents consumed in their task of exploration and mapping.

For these experiments, routing agent overhead came in two forms: the total number of agents operating in the network, and the size of the memory store for each agent. The system would always work better with more agents and with longer agent memories; however, additional agents and increased memory come at some cost.

Figure 2.3 demonstrates the performance curve for our agent system given a fixed overhead budget. This budget can be spent either as additional routing agents (on the right side of the graph), or as additional agent memory (on the left). The y-axis shows the relative connectivity of the network given several choices about how to apportion overhead resources: the graph clearly demonstrates that there is an optimal point for this system given this tradeoff. In these



Effect of trading off history and population overhead on network connectivity
Interval represents max and min connectivity, point is average.

Figure 3: Overhead vs. Performance in Mobile Agent Routing

experiments, we find that a network of 250 nodes is best served by 135 agents with a memory store of 18 moves. The more general lesson here is that there are ways to structure mobile agent systems that allow them to be adjusted in comprehensible ways, with designers and administrators able to trade off various costs to tune for desired behaviors.

3 Future Directions

Mobile agents research is a very young field, and much work still focuses on proving the fundamental practicality of this new approach. In the following section we would like to outline some of the longer term goals that guide our work in this area, and some of the possibilities we see for fully realized mobile agent systems.

3.1 Agent Specialization and Run-time Flexibility

Mobile agent populations need not be homogeneous. We are interested in exploring the idea that a diverse collection of interacting mobile programs forms an *agent ecology*, with useful macro-level characteristics of such an ecology varying along with the numbers and proportions of its constituent agents. Human network managers can induce and manage the diversity of the agent ecosystem, monitoring the state of a sub-network and adjusting the agent population to match changing circumstances. Or agent populations can be designed to regulate themselves. For example, a very simple self adjustment strategy would be for each agent belonging to a certain class to monitor the resources that it and its peers consume, and to die off or spawn new copies of itself depending on supply and demand.

Different agents in an ecology would fill different roles. Several distinct kinds of specialization are possible:

- Agents can specialize with regard to usage patterns across the network.

- Specialized agents could work to manage requirements in specific areas of a network.
- Agents can adapt the network infrastructure to changing needs over time.
- Agents can specialize on behalf of specific users.

This last type of specialization points at perhaps the most distinctive new feature of mobile agent architectures: the ability for individual users to inject code into a system that changes how local infrastructure works. Collections of agents can be dispatched to serve a particular user's needs; the network is no longer "one size fits all."

There are significant difficulties to be addressed, of course, before user level configurability is practical. Issues of network efficiency, stability, and predictability must be addressed. System security in a mobile code environment is an extremely important research topic [10]. Mobile agent tools may give us more, rather than less, opportunity to safeguard our information and infrastructure (by providing ways to specify trust relationships that can be verified on the the fly, for example).

3.2 Dense Wireless Connectivity

Network connectivity is more and more important, but we lack general purpose systems that allow mobile, flexible, wireless data communications. Cellular telephone and pager networks cover much on the globe, but are expensive installations optimized for very particular usage. Satellite systems are even more expensive and require prohibitively large power output and antennae. Wireless LANs (based on standards like 802.11) free laptop users to move through buildings but will never enjoy widespread coverage and are so power hungry and complex as to be unsuitable for small, simple devices.

There are no good options for widespread, high bandwidth, low power, low overhead wireless connectivity. This reality has directed our research at the MIT Media Lab towards networks based on mobile agents. We work in a building that is populated with laptop computers and PDAs in great numbers. We also have hundreds of smaller devices built as part of research projects, most of which incorporate ad-hoc radio or infrared network links. We very much need to integrate our various wireless networks; to create a future in which dense collections of heterogeneous devices can all easily communicate with one another.

A next generation network appropriate for our environment must address many design constraints. Networks must admit a wide variety of usage models and data rates, because the devices we build vary greatly. They must use common bandwidth very efficiently, because we build and deploy thousands of devices in a small area. Low power operation is extremely important, as battery life tends to be a limiting resource for much of our hardware. And the communications links and mechanisms need to be abstract or transparent enough that a busy researcher will find it easy to incorporate a standard communications component. Finally, gateways to the Internet at large need to be ubiquitous, which means they must be cheap to build, easy to deploy, and simple to maintain.

We are in the process of constructing the initial version of a system that we hope will eventually make all of this possible. We have designed and fabricated a simple node, based on an inexpensive 16 bit microcontroller and capable of executing mobile agent code written in a subset of Java. This hardware will become the basis for a network built from and managed by mobile agent techniques, and is currently in use at the core of several communications-centric children's toys. The next step in our research is to settle on a high bandwidth, very short range radio transceiver, and to begin implementing on this new system the agents algorithms that we have thus far explored only in simulation.

3.3 Conclusion

In this paper we have discussed three approaches to building next generation networks. These methods are fundamentally similar; they are all predicated on the idea that networks can be made more flexible and useful by embedding code mobility deep into their infrastructures. Active networking researchers are building systems consisting of highly configurable routers and smart packets. Social insect models provide robust, distributed mechanisms for adapting to changing circumstances. And in our work at the MIT Media Lab we are researching open ended, run time manipulable architectures in which large collections of mobile agents cooperatively manage network services such as mapping and routing.

References

[1] Steve Appleby and S. Steward. Mobile software agents for control in telecommunications networks. *BT Technology Journal*, 12(2):104–113, April 1994.

[2] Joachim Baumann. Mobility in the Mobile-Agent-System Mole. In *CaberNet: 3rd Plenary Workshop*, 1997. http://www.informatik.uni-stuttgart.de/ipvr/vs/Publications/1997-baumann-05-paper.html

[3] E. Bonabeau, F. Henaux, S. Guérin, D. Snyers, P. Kuntz, and G. Theraulaz. Routing in Telecommunications Networks with “Smart” Ant-Like Agents. In *Intelligent Agents for Telecommunications Applications ’98*, 1998. http://www.santafe.edu/sfi/publications/Abstracts/98-01-003abs.html

[4] Rodney Brooks. A Robust Layered Control System Architecture. *IEEE Journal of Robot Automation*, RA–2(1), 1986.

[5] G. Di Caro and M. Dorigo. Mobile Agents for Adaptive Routing. In *Proceedings of the 31st Hawaii International Conference on Systems*, January 1998. ftp://iridia.ulb.ac.be/pub/dorigo/conferences/IC.22-HICSS31.ps.gz

[6] D. Chess, C. Harrison, and A. Kershenbaum. Mobile agents: Are they a good idea? In *”Mobile Object Systems: Towards the Programmable Internet”*, volume 1222 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997. http://www.research.ibm.com/massive/mobag.ps

[7] Danny B. Lange and Mitsuru Oshima. Seven Good Reasons for Mobile Agents. *Communications of the ACM*, 42(3):88–89, March 1999. http://www.acm.org/pubs/citations/journals/cacm/1999-42-3/p88-lange/

[8] Nelson Minar, Kwindla Hultman Kramer, and Pattie Maes. Cooperating Mobile Agents for Dynamic Network Routing. In Alex Hayzelden, editor, *Software Agents for Future Communications Systems*, chapter 12. Springer-Verlag, 1999. ISBN: 3-540-65578-6. http://www.media.mit.edu/~nelson/research/routes-bookchapter/

[9] Nelson Minar, Kwindla Hultman Kramer, and Pattie Maes. Cooperating Mobile Agents for Mapping Networks. In *Proceedings of the First Hungarian National Conference on Agent Based Computation*, 1999. http://www.media.mit.edu/~nelson/research/routes-coopagents/

[10] Tomas Sander and Christian F. Tschudin. Protecting mobile agents against malicious hosts. In Giovanni Vigna, editor, *Mobile Agents and Security*, volume 1419 of *Lecture Notes in Computer Science*, chapter 4, pages 44–61. Springer-Verlag, 1997. ISBN: 3540647929. http://www.icsi.berkeley.edu/~tschudin/

[11] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz. Ant-based Load Balancing in Telecommunications Networks. *Adaptive Behavior*, 5(2):169–207, 1997. http://www-uk.hpl.hp.com/people/ruud/abc.html

[12] D. Tennenhouse, J. Smith, W. Sincoskie, D. Wetherall, and G. Minden. A Survey of Active Network Research. *IEEE Communications Magazine*, 35(1):80–86, January 1997. http://www.tns.lcs.mit.edu/publications/ieeecomms97.html

[13] James E. White. Telescript Technology: Mobile Agents. In Jeffrey Bradshaw, editor, *Software Agents*. AAAI Press/MIT Press, 1996. http://www.genmagic.com/agents/Whitepaper/whitepaper.html