Dynamic Agent Population in Agent-Based Distance Vector Routing

Kaizar A. Amin and Armin R. Mikler
Department of Computer Science
University of North Texas
Denton, TX 76203
(amin, mikler)@cs.unt.edu

Abstract

The Intelligent mobile agent paradigm can be applied to a wide variety of intrinsically parallel and distributed applications. Network routing is one such application that can be mapped to an agent-based approach. The performance of any agent-based system will depend on its agent population. Although a lot of research has been conducted on agent-based systems, little consideration has been given to the importance of agent population in dynamic networks. A large number of constituent agents can increase the resource overhead of the system, thereby impeding the overall performance of the network. Hence, it is imperative to find the optimal number of agents in the system that would maximize the efficiency of the agent-based mechanism in the network. This optimal value cannot be determined manually, thereby emphasizing the need for an adaptive approach that manipulates the number of agents in the system based on its resource availability. This paper discusses an agent-based approach to Distance Vector Routing, referred as Agent-based Distance Vector Routing and also describes an adaptive approach controlling the number of agents in the network using pheromones and discusses their limitations.

1. INTRODUCTION

Agents, Software Agents, and Intelligent Mobile Agents are terms, which describe the concept of mobile computing or mobile code. The mobile agent paradigm has attracted attention from many fields of computer science. The appeal of mobile agents is quite alluring - mobile agents roaming the network could search for information, meet and interact with other agents that roam the network or remain bound to a particular machine. An agent manifests four distinct characteristics, namely, *Intelligence*, *Communication*, *Autonomy*, and *Mobility* [Minar et al. 1998, Di Caro & Dorigo 1997]. Intelligence is the ability of the agent to adapt itself and/or change its environment based on the information available to it. Communication is the property of an agent whereby it coordinates with other agents residing on the same node in exchanging data, making decisions to merge itself [Mikler & Chokhani 2001], or planning its future strategies. Through autonomy, the agent has the authority to control its actions and strategies without the necessity of human control. Mobility is the property of the agents that make them conducive for distributed network applications. Mobile agents have the ability to migrate throughout the network, performing specific tasks at each node reaching towards a global goal.

The agent-based paradigm can be applied to many intrinsically parallel network application. A large number of applications in communication networks have been identified that can benefit from an agent-based approach. Some examples on the system level include load balancing, network management and network routing [Di Caro & Dorigo 1997, Minar et al. 1999]. In this paper we apply the agent paradigm to network routing. Most of the work in agent-based network routing is based on insect colonies. It relies on the principles that individual insects exhibit a simple behavior while collective communities exhibit complex problem solving capabilities. For example, individual ants have limited abilities, however ant colonies are capable of performing tasks that are remarkably complex. Considerable research has been conducted in mapping the foraging activities of ants to routing and network management activities of mobile agents. Real ants are represented as artificial agents that traverse the network collecting specific information from their environment and coordinate their actions through *pheromones*. On the basis of this information the agents make several decisions to adapt their behavior (Reactive Agents) and/or change the existing environment affecting their future inputs (Proactive Agents).

This paper focusses on a new implementation of Distance Vector Routing Algorithm (DVR) using an agent-based approach – Agent-based Distance Vector Routing (ADVR). DVR is a simple, iterative, asynchronous and completely distributed routing algorithm [Bertsekas & Gallager 1987]. Certain implementations of DVR such as Routing Information Protocol (RIP) are used widely in many networks as they can be easily configured and maintained. However, it has been shown that a large number of update messages exchanged by adjacent nodes in a network constitute considerable resource overhead [Bertsekas & Gallager 1987]. Reducing the resource overhead may allow for DVR-class algorithms to be deployed in a wide range of networks (wireless, Ad-Hoc) which require a simple, resource efficient routing protocol due to limited availability of resources (memory, bandwidth). The number of messages in ADVR, at a given instance, is bounded by the number of constituent agents in the network. In ADVR, the exchange of the metrics and the process of route discovery moves from the nodes to the agents. Hence in this approach, the route discovery is manifested in the movement of agents carrying routing information from one node to another rather than the propagation of individual update messages. The performance of any agent-based system will depend on its agent population. Most of the agent-based implementations assume a fixed number of agents in the network. Certain systems create agents at regular intervals and destroy them once the required task is accomplished [Di Caro & Dorigo 1997]. Although the latter approach provides some degree of flexibility it does not adapt to sudden changes in the network topology. It is difficult to know, a *priori*, the optimal degree of concurrency or the number of agents in the system since it depends on the network dynamics and availability of resources. Hence, autonomous agent-based systems should be capable of adapting to their environment and controlling the agent population as a function of their working environment.

Section 2 describes the routing mechanism used in ADVR. Section 3 provides an overview of the migration strategy implemented by ADVR. Section 4 emphasizes the need for a dynamic agent population in ADVR. The paper concludes with Section 5 summarizing the paper and giving a brief overview of ongoing research.

2. ADVR

Distance Vector Routing (DVR) algorithms exchange a *metric* that represents the distance from a node n_i to any destination n_j. In most implementations of DVR, this information (metric) is exchanged among adjacent nodes in the form of triggered updates, which is initiated when there is a change in the routing table of one of the neighboring nodes. After receiving the update information from a neighboring node, a node n_i updates its own routing table in the following manner [Hedrick 1988, Bertsekas & Gallager 1987]:

$$D(i,j) = \begin{cases} 0 & \forall \quad i = j \\ min[d(i,k) + D(k,j)] & \forall \quad otherwise \end{cases} \tag{1}$$

where $D(i,j)$ represents the metric of the best route from node n_i to node n_j currently known to n_i. $d(i,k)$ represents the cost of traversing the link from node n_i to node n_k

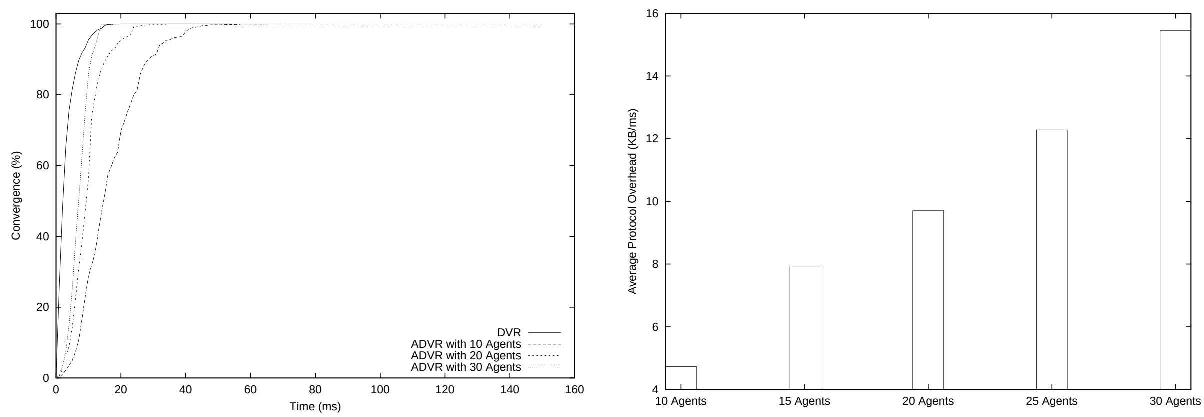
While the message activity in conventional DVR can escalate to consume significant amounts of network resources, the number of messages in ADVR, at a given instance, is bounded by the number of constituent agents in the network. In ADVR, the exchange of the metrics and the process of route discovery moves from the nodes to the agents [Minar et al. 1999]. Hence in this approach, the route discovery is manifested in the movement of agents carrying routing information from one node to another rather than the propagation of individual update messages. An agent can be formally described as: $\Lambda(i, x, y, R_x, \gamma)$, where Λ is an Agent with ID i migrating from node n_x to node n_y, carrying the routing table R_x and using the migration strategy γ to move among adjacent nodes. R_x is a subset of r_x, the routing table of n_x.

In ADVR, agents start at arbitrary nodes and migrate to adjacent nodes using γ. Upon arriving at a node n_y, an agent $\Lambda(i, x, y, R_x, \gamma)$ updates the routing table R_y based on the following equation:

$$D(y,j) = min(D(y,j), [d(y,x) + D(x,j)]) \ \forall \ n_j \text{ in } R_x \tag{2}$$

where $D(x,j)$ is an entry in R_x. While equation(2) is based on equation(1), it is performed less frequently in ADVR as compared to DVR. After performing the update, the agent selects R_y and migrates to an adjacent node using migration strategy γ.

At every node the agent has to make a decision regarding the routing table data it would carry to the next node. This decision plays an important role in providing a resource efficient solution with ADVR. If the agent carries the entire routing table available at each node, it would incur excessive overhead in transferring redundant data. On the other hand if the agent selected a subset of total routing data



(a) Path-Cost Convergence

(b) Average Routing Overhead until Convergence

Figure 1: Analysis of Routing Overhead in DVR and ADVR

available at the node, it would unnecessarily delay the propagation of important routing information. Hence, agents identify routing table entries that have been modified, yet have not been transferred to a particular neighbor. Associated with every entry e_{xi} in routing table r_x is a vector V_{xi} of boolean flags for each of the neighbors nodes of n_x. $\mid V_{xi} \mid = \mid H_x \mid \ \ \forall e_{xi}$, i.e., the size of each of the V_{xi} is equivalent to the size of the neighborhood of a node n_x, H_x. Upon selecting a neighbor n_y of the current node n_x, an agent $\Lambda(i, x, y, R_x, \gamma)$ will carry only those entries e_{xi} in R_x for which $V_{xi}[y] == 1$. The agent copies each entry e_{xi} that is to be transferred to neighbor node n_y to its data segment, and sets the corresponding boolean flag $V_{xi}[y] == 0$. At startup, all the flags are set, i.e., $V_{xi} := 1 \ \forall e_{xi}$.

It was observed that DVR is characterized with parallelism due to the broadcast storm incurred by it making it highly reactive to small changes. On the other hand, ADVR is restricted in its parallelism by the number of agents in the network which controls the outburst of routing packets. Figure 1(a) displays the aggressive messaging of DVR manifested in its path-cost convergence. It can be seen although, a small agent population cannot compete with the convergence of ADVR, a substantial number of agents can outperform DVR in terms of convergence. Intuitively, an arbitrarily large agent population would converge significantly faster, and hence, would be desirable. Nevertheless, a larger agent population has a significantly larger average message overhead (see Figure 1(b)). A large average overhead implies that a substantial number of agents traverse the network concurrently, thereby, imposing resource requirements on the network. For scalable systems, the average overhead should be as low as possible. Therefore it is apparent that significantly large agent populations, resulting in high average overhead and therefore hamper the scalability of ADVR. On the other hand, a very small agent population will hinder the performance of ADVR, in terms of convergence times and route discovery in dynamic networks.

3. AGENT MIGRATION STRATEGY

It has been shown in the previous section that, in ADVR, agents migrate among nodes, thereby establishing routes for every pair of nodes in the network in a distributed way. Hence, the efficiency of ADVR, in terms of the route discovery, is characterized by the migration strategy of the agents. It is important that the agents migrate intelligently, since an imprudent strategy can severely affect the performance of ADVR. It is imperative that an agent-based system carefully chooses its migration strategy as there is no consensus on a single global optimal strategy. A method suitable for one system can produce unwanted side effects for other systems.

A *Random Walk* is a naive agent migration strategy whereby an agent residing at a node n_x randomly selects any one of the neighbors of n_x [Amin et al. 2001, Minar et al. 1998]. Although this method is simple to implement, it fails to exploit the intelligence of mobile agents. Another method implemented by systems could be a *Least Recently Used* (LRU) on nodes whereby agents perform a *depth-first-search* of the network based on network information carried by them, such as agent migration history [Minar et al. 1998]. Systems implementing LRU could benefit from population of agents exchanging their information. Multiple agents on the same node exchange their data and make migration decisions based on the overall data. However, it was observed that by exchanging their data, all the agents on that node contain the same data thereby making similar decisions. This results in clustering of agents in specific parts of the network. Motivated by the success of insect colonies, certain systems use a population of naive, autonomous agents performing complex tasks using *Stigmergy* [Di Caro & Dorigo 1997, White 1997]. Stigmergy is the mechanism for naive individuals to communicate with each other by changes in the environment. Most of the migration strategies based on insect colonies are mapped to the foraging activities of ants. Although this strategy has shown impressive results in certain applications, it tends to favor migration patterns. Hence, it may not be a feasible solution for systems that require agents to explore the entire network with equal probabilities.

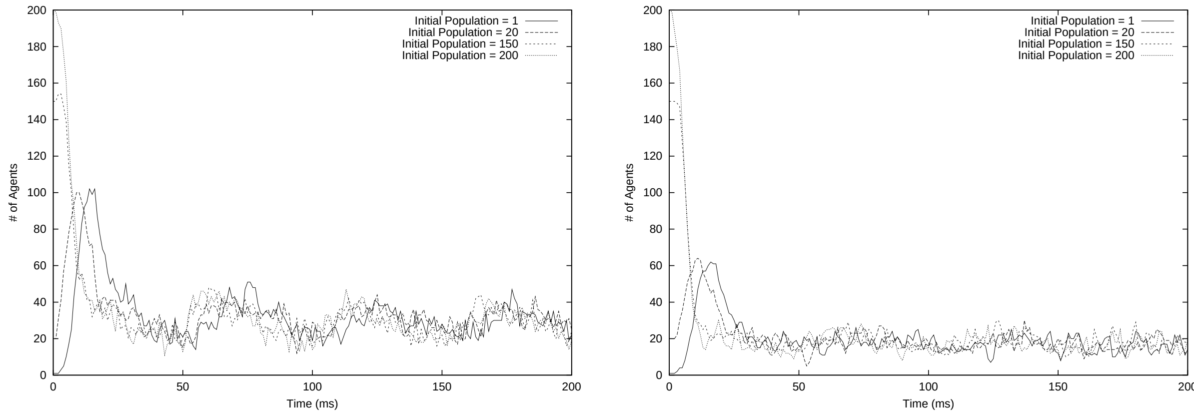
For ADVR we have combined the strengths of both the approaches, the stigmergetic feature of the insect colonies and the exploratory feature of the depth-first-search. That is, with very little knowledge of the network, the agents communicate with each other via the environment and perform the depth-first-search on the network as a community. The agents do not carry any network information; they simply indicate their presence using pheromone trails. A pheromone is a volatile chemical released by insects in the environment indicating their presence. Ants use pheromone trails to follow the path of the successor ant. While the ant pheromones are used to attract other members of the community [Schoonderwoerd et al. 1997, Di Caro & Dorigo 1997, White 1997], in our approach, pheromones repel other agents. An agent traversing a link xy from node n_x to n_y deposits a pheromone on xy. Another agent migrating from n_x will chose a link with the weakest pheromone value thereby migrating to a least recently visited region of the network. This paper refers to this class of pheromones, that assist in agent migration strategy as *Link Pheromones*. It can be observed that such an approach exploits the stigmergetic behavior of insect colonies and avoids the clustering of agents in specific regions of the network. By changing and retrieving information from the environment as opposed to carrying the network information [Minar et al. 1998], agent in ADVR impose minimal resource requirements.

4. DYNAMIC AGENT POPULATION

Most applications implementing an agent-based approach assume a static agent population and analyze their results. Some systems implementing insect colonies are oblivious to their environment, creating/deleting agents periodically. A large population of agents would increase the parallelism of the system, thereby converging faster [Amin et al. 2001]. However, it is extremely important to analyze the agent overhead in terms of bandwidth consumption and computational cycles. It is imperative to understand that increasing the agent population will increase resource demands in the network which would indirectly affect the network performance. Hence, as explained in Section 2, it is essential to control the agent population in order to minimize the average routing overhead of ADVR without impeding the its path-cost convergence. However, the unpredictable behavior of dynamic networks make it very difficult to estimate *a priori*, an appropriate number of agents in the network. Thus, it is critical that an adaptive system dynamically alters the agent population in response to its resource availability.

Changing the agent population dynamically in response to its environment is a non-trivial issue in the absence of a central controller. Individual agents lacking a bird's eye view of the system are unable to make global assessments regarding the environment. Therefore, it requires a high degree of coordination among agents to analyze the global environment from local information. To facilitate such a coordination, this paper exploits the stigmergetic properties of agents. Mobile agents with minimum cognitive capabilities communicate with each other using pheromones, establishing an infrastructure that assists them in assessing their environment. Pheromones that aid the agents in population control are referred to as *Node Pheromones* to distinguish them from *Edge Pheromones*.

The strength or intensity of *Node Pheromones* can be expressed by the equation $e^{-\lambda(\Delta t)}$, where λ represents the degree of volatility of the pheromone and Δt is the time since the deposition of the pheromone. Using this equation the agents can extract the value of the *Node Pheromone* at a given time and analyze the frequency of inter-agent arrival at that node. An agent visiting a node n_x at time t_2 extracts the value of the *Node Pheromone* that was deposited at time t_1 using the equation $e^{-\lambda(t_2-t_1)}$. If this value is above a certain *Termination Threshold* (Ψ) and the agent did not produce any routing update on n_x, the agent



(a) Variance in Agent Population with $\Psi = 0.8$, $\Omega = 0.3$, and $\lambda = 1$

(b) Degree of Volatility $\lambda = 0.5$

Figure 2: Dynamic Control of Agent Population

kills itself. However if the *Node Pheromone* value reduces below a *Cloning Threshold* (Ω), the agent *clones* itself. Before leaving n_x, the agent deposits additional *Node Pheromone* at time t_2. This approach controls the agent population based on the inter-agent arrival time expressed as a function of the *Node Pheromone*. If the inter-agent arrival time is small $(e^{-\lambda(\Delta t)} > \Psi)$ and the agent produced no updates in the existing routing table entries, it implies an excessive number of agents in the system. On the other hand, if the inter-agent arrival time is large $(e^{-\lambda(\Delta t)} < \Omega)$, it implies there are a sub-optimal number of agents in the system. However if $\Omega \leq e^{-\lambda(\Delta t)} \leq \Psi$, the agent neither clones nor kills. Killing requires the agent to destroy its instance along with its data segments. Cloning requires the agent to create another instance of itself with same attributes and privileges.

Figure 2(a) shows the convergence of the agent population in a 40 node network with average degree of 7. The results assume the $\Psi = 0.8$, $\Omega = 0.3$, and $\lambda = 1$. It was observed that irrespective of the initial population, the system converges to a fairly constant number of agents in the system (approximately 20). Networks initialized with a small number of agents escalate the agent population to a certain value thereby improving the network performance. However, the escalation of agent population occurs at every node flooding the network with excessive agents. Nevertheless, the system realizes the overhead of large number of agents and adjusts itself to dynamically reduce the population. On the other hand, networks initialized with a large number of agents realize the per-agent overhead and continuously reduce the population until it reaches a somewhat constant number.

Figure 2(b) displays the variance in agent population with *Node Pheromones* having reduced degree of volatility (λ). A reduced degree of volatility denote more stable pheromones. Hence their strength decay at lower rates reducing the frequency of cloning functions and increasing the frequency of killing functions. Low values of λ significantly reduce the initial burst of agents in systems initialized with small agent population. Although less volatile pheromones help to reduce the initial agent overhead, it also reduces the sensitivity of the system. ADVR implementing a dynamic agent population may start with a single agent or an arbitrary number of agents. Nevertheless, the agents coordinate themselves and converge to a particular range of population. Although this range may not represent the ideal number of agents required for the system to converge, it represents an optimal population based on the availability of resources. This range however depends on the values of Ψ, Ω, and λ. An adaptive system should adjust these values dynamically based on its resource availability. Dynamically adjusting these thresholds in a distributed fashion as a function of network topology, congestion level, etc. is a non-trivial problem and is beyond the scope of this paper.

5. SUMMARY

In this paper, we have described a distance vector routing scheme that is based on the mobile agent paradigm – Agent-based Distance Vector Routing (ADVR). ADVR reduces the parallelism in the network by controlling the number of constituent agents in the network. It was shown that ADVR exploits the stigmergetic features of insect colonies in its migration strategy. A large number of agents would ideally improve the performance of ADVR. However, an excessive number of agents impede the performance of other applications due to per-agent overhead. Therefore an optimal number of agents are required in the network that would improve the performance of ADVR without affecting the performance of other applications in the network. Nevertheless, network characteristics change over time, continuously varying the optimal population. Hence, the optimal number of agents cannot be determined manually. This paper describes an adaptive approach whereby agents retrieve local information from their environment and make decisions to either kill or clone themselves. The local information retrieved is in the form of pheromones deposited by other agents. It was shown that using the adaptive approach, a network initialized with an arbitrary number of agents would converge to a range of value comprising the optimal population of agents. It was further shown that this range was dependent on the *Termination Threshold* (Ψ), the *Cloning Threshold* (Ω), and the *Degree of Volatility* (λ). Ongoing research is focusing on overcoming the limitations of our current experiments. A completely adaptive system would manipulate the values of Ψ, Ω, and λ dynamically based on the resource availability. Current research is focusing on analytic functions that would manipulate these values dynamically and map them to the number of optimal agents in the system.

References

[Amin et al. 2001] K. Amin, J. Mayes and A. Mikler, *Agent-based Distance Vector Routing*. Proceedings of the Third International Workshop, MATA 2001, Montreal, Canada, August 2001.

[Bertsekas & Gallager 1987] D. P. Bertsekas and R. G. Gallager, *Data Networks*. Prentice-Hall, 1987.

[Di Caro & Dorigo 1997] G. Di Caro and M. Dorigo, *AntNet: a mobile agents approach to adaptive routing*. Tech. Report, IRIDIA/97-12, Universit Libre de Bruxelles, Belgium.

[Di Caro & Dorigo 1998] G. Di Caro and M. Dorigo. *AntNet: Distributed Stigmergetic Control for Communications Networks*. Journal of Artificial Intelligence Research, 1998.

[Di Caro & Dorigo 1998] G. Di Caro and M. Dorigo, *An Adaptive Multi-Agent Routing Algorithm inspired by Ants Behavior*. Proceedings of PART98 - 5th Annual Australasian Conference on Parallel and Real-Time Systems, pages 261–272. Springer-Verlag, 1998.

[Hedrick 1988] C. Hedrick, *Routing Information Protocol*. RFC 1058, June 1988.

[Mikler & Chokhani 2001] A. R. Mikler and V. Chokhani, *Agent Based Wave Computation: Towards Controlling the Resource Demand*. Proceedings of the International Workshop IICS (Innovative Internet Computing Systems) 2001, Ilmenau, Germany, June 2001.

[Minar et al. 1998] N. Minar, K. H. Kramer and P. Maes, *Cooperating Mobile Agents for Mapping Networks*. Proceedings of the First Hungarian National Conference on Agent Based Computing, 1998.

[Minar et al. 1999] N. Minar, K. H. Kramer and P. Maes, *Cooperating Mobile Agents for Dynamic Network Routing*. Proceedings of the Software Agents for Future Communications Systems, Springer-Verlag, 1999, ISBN 3-540-65578-6.

[Schoonderwoerd et al. 1997] R. Schoonderwoerd, O. Holland, and J. Bruten, *Ant-like Agents for Load Balancing in Telecommunications Networks*. Proceedings of the First International Conference on Autonomous Agents, pages 209–216. ACM Press, 1997.

[White 1997] T. White, *Routing with Swarm Intelligence*. Technical Report SCE97 -15, Systems and Computer Engineering Department, Carleton University, September, 1997.