Abstract

Topology discovery is a key task for several computer network applications such as diagnosis, routing and network management. Traditional approaches for topology discovery cannot always be used in dynamic and decentralized networks, such as unstructured peer-to-peer networks and wireless ad hoc networks. This paper introduces a strategy based on mobile agents and swarm intelligence for topology discovery in such environments. The proposed strategy is inspired by ant colonies, employing simple agents that disseminate information about the topology and communicate through stigmergy. Experimental results show that the nodes obtain descriptions which are very close to the real network topology. It is also shown that the stigmergy-based method for the selection of agent destinations produces better results than a random selection, and that the number of agents can be dynamically adjusted as the size of the network changes.

1 Introduction

Knowledge about the topology of a computer network is useful, or even crucial, to several distributed applications. Algorithms considering topology aware nodes can be employed in areas such as resource location in peer-to-peer networks [2], management [8], routing [1], and diagnosis [9]. Several strategies have been proposed to automatically build topology descriptions, containing representations for the nodes and communication links, as well as their organization [4, 3]. Those strategies are usually centralized, with the description being built by a single node, which can share the description with other nodes, if necessary. Furthermore, traditional strategies, either centralized or distributed, usually assume that the network topology stays unchanged at least for a finite period of time.

However, centralization is undesirable, or even impossible, in some networks. In these cases, topology discovery must be performed by several nodes, which exchange information with each other. Moreover, in some networks the topology is dynamic, i.e. it may change over time, with nodes and links being added or removed continuously. Such networks require a permanent topology discovery process, instead of a task which eventually comes to an end. This work considers the problem of topology discovery in such decentralized and dynamic networks.

Dynamism and decentralization characterize some increasingly popular network architectures, such as unstructured peer-to-peer (P2P) networks [12] — P2P networks which don’t have centralized servers, structured topology, or rigid schemes for resource positioning. An example of a network like this is the Gnutella [11] network. Mobile ad hoc networks [5], in which the nodes are portable devices that communicate through wireless interfaces, are also dynamic and decentralized.

In this work, we introduce a new strategy for topology discovery in such settings. The strategy employs mobile agents, capable of migrating across the nodes in a network. The proposed strategy is inspired by an “ant colony” [10], employing simple agents that exhibit swarm intelligence when interacting. The agents have some degree of randomness in their behavior and communicate through stigmergy — an indirect scheme in which the agents do not communicate directly, but leave “signs” in the environment, usually in the form of “pheromones”. Experiments conducted over a simulated environment show that the strategy can produce topology descriptions that are very close to the real network topology, and that the agents can use pheromones to control their population and select better itineraries.

The rest of this paper is organized as follows. Section 2 introduces the proposed strategy for topology discovery. Section 3 describes the experiments performed to evaluate this strategy. Section 4 points to related work. Finally, section 5 discusses the conclusions and future work.
2 Proposed Strategy for Topology Discovery

This section describes the proposed strategy for topology discovery. The aim is to build topology descriptions in the nodes, so that each node has its own “view” of the network. As the network is dynamic, the strategy is always running — topology discovery is a continuous process, not a task that eventually comes to an end. Subsection 2.1 defines the system model and subsection 2.2 describes the basic mechanism behind the strategy. The other subsections detail several important issues which must be considered.

2.1 Network Model

The system is modeled based on some features shared by unstructured P2P networks and mobile ad hoc networks. There is no central manager, nor there is a global clock shared by the nodes. Besides, the topology may change constantly and unpredictably — its size is arbitrary.

The topology is defined by a graph \( G(t) = (V_t, E_t) \), in which, at time \( t \), each vertex \( v_{ijt} \in V_t \) is a node \( i \) and each edge \( e_{ijt} \in E_t \) is a communication link between nodes \( i \) and \( j \). A link may be physical (e.g. a wire) or logical (e.g. an Internet connection). It can be seen that the topology is a function of time: given \( t_1 \) and \( t_2 \) so that \( t_1 \neq t_2 \), \( G(t_1) \) may be different of \( G(t_2) \). We assume the following:

1. A node knows at all times and without error its neighborhood (the nodes to which it is connected by a link).
2. An agent can migrate between two nodes without error in a non-zero, but bounded, time. The bound is known.
3. Even though there is no global clock, the timescale (the difference between time \( t \) and the following time \( t + 1 \)) is the same for all the nodes.
4. The links are bidirectional: \( e_{ijt} = e_{jiti} \).

2.2 Basic Mechanism

In our strategy, several discovery agents migrate across the network nodes, disseminating information given as graphs that describe the topology. The agents do not have a fixed destination, remaining in the network indefinitely. Each node keeps locally a topology description, which has at least the node, its neighbors, and the links between them. Any other information must be brought to the node by the agents. Agents and nodes interact as follows:

1. A discovery agent arrives at a node.
2. The topology description is updated with information brought by the agent.
3. The agent remains idle for some time.
4. The topology description is copied by the agent.
5. The agent migrates to one of the node’s neighbors.
6. The cycle is restarted.

2.3 Information Aging

As the time needed for an agent to migrate is non-zero, there is an interval between the time a piece of information is generated and its arrival in a node. The topology may change in this interval, and a node can end up receiving information that does not correspond to reality anymore. The same problem can occur with information already in a node — a piece of information will become outdated if the topology changes and the node is not aware of this.

As the probability a piece of information has of becoming outdated increases as time passes, nodes and agents must get the most recent information they can. When a topology description is updated, only pieces of information that are more recent than the ones already in the description are used. Some information may even be deemed too old to be reliable: when a piece of information “ages” beyond a certain limit, it is discarded by nodes and agents. Special attention must be given to this limit: if it is too high, the descriptions are more prone to contain outdated information. On the other hand, if the limit is too low, the descriptions will have to be updated more frequently.

Determining the age of a piece of information is also a problem: as the nodes do not share a global clock, the exact time an agent takes to migrate is unknown, and so is the amount of time the information should age when the agent migrates. To overcome this, the age is estimated using the bound for the time an agent takes to migrate between two nodes, which is known. This bound is added to the age of the information carried by an agent when it migrates. That way, the estimated age is always greater than or equal to the real age, and no information is deemed more recent than it really is. Information present in the topology descriptions can follow the node’s local clock, because it is assumed that all the nodes have the same timescale.

2.4 Agent Migration

Another important issue is defining the paths traversed by the agents: as they disseminate information to the nodes, the performance of the strategy is directly related to those paths. If all agents follow similar paths, little or no relevant information will be taken to some nodes, while other nodes will be visited infrequently or even will not be visited at all. Thus, the agents must “spread” across the network, taking information to as many nodes as possible.
We assume that all the decisions regarding migration are taken by the agents themselves, with the destination being chosen just before the agent migrates from the current node. Centralized approaches, as well as approaches that include extra communication or coordination between nodes and agents are undesirable, or even unfeasible in the settings considered in this work. That means an agent should use only information local to the node it is currently in.

We propose two approaches for the selection of agent destinations. The first is a simple random selection: the agent randomly selects its next destination among the current node’s neighbors. We also propose a stigmergy-based approach, in which the agents communicate by depositing “pheromones” in the nodes. These pheromones make the agents “flee” from each other, spreading more efficiently across the network. Each node has a pheromone deposit associated with each of its links. Before migrating from node $i$ to node $j$, an agent updates the pheromone concentration $C_{ij}(t)$ associated to link $e_{ij}$ in $i$, as shown in equation 1.

$$C_{ij}(t) = C_{ij}(t) + 1$$

As time passes, pheromones in a deposit “evaporate” and weaken. Evaporation occurs at a rate $0 \leq \rho < 1$ for each unit of time. The higher the $\rho$, the faster the pheromones evaporate. An evaporation rate of 0 indicates that pheromones never weaken. The concentration at time $t$ is given as a function of the concentration in a previous time $t_0 < t$, according to equation 2.

$$C_{ij}(t) = C_{ij}(t_0) \cdot (1 - \rho)^{t - t_0}$$

It is then possible to determine how pheromone concentrations affect agent migration. An agent’s next destination is selected probabilistically. The pheromone concentration in each deposit affects the chance a link is chosen by a constant $\alpha > 0$, which indicates the pheromone “strength”. The higher the $\alpha$, the higher the weight of each pheromone unit. Given $|e_{it}|$ the number of neighbors of node $i$, and $E_{it}(k)$ the $k$-th neighbor of node $i$ at time $t$, the probability $P_{ij}(t)$ of link $e_{ij}$ being chosen is given by equation 3.

$$P_{ij}(t) = \frac{(C_{ij}(t) + 1)^{-\alpha}}{\sum_{k=1}^{|e_{it}|} (C_{iE_{it}(k)}(t) + 1)^{-\alpha}}$$

2.5 Agent Population

As the size of the network can change over time, the number of agents used for discovery must be adaptable. The same pheromone trails left when the agents migrate are used to control the population. When the pheromone concentration in a node reaches an upper threshold, all the agents arriving at the node are destroyed before migrating. If the concentration reaches a lower threshold, the node creates an agent. Each node uses only its own local pheromone concentrations for this task. The total pheromone concentration $C_i(t)$, which must be compared to the minimum and maximum thresholds, is the sum of the concentrations in all of node’s $i$ deposits. Given $|e_{it}|$ the number of neighbors of node $i$; $E_{it}(k)$ the $k$-th neighbor of $i$, and $C_{ij}(t)$ the pheromone concentration associated to link $e_{ij}$ at time $t$, the total concentration $C_i(t)$ is given by equation 4.

$$C_i(t) = \sum_{k=1}^{|e_{it}|} C_{iE_{it}(k)}(t)$$

It is expected that a higher number of agents passes through nodes with more neighbors. That way, the minimum and maximum thresholds are altered by the number of neighbors of each node, according to a constant $\delta \geq 0$. Given a minimum threshold $\gamma_{\text{min}} > 0$ and a maximum threshold $\gamma_{\text{max}} > \gamma_{\text{min}}$, the total pheromone concentration $C_i(t)$ must be compared to values $L_{\text{min}}(t)$ and $L_{\text{max}}(t)$, defined respectively by equations 5 and 6.

$$L_{\text{min}}(t) = \gamma_{\text{min}} \cdot |e_{it}|^\delta$$
$$L_{\text{max}}(t) = \gamma_{\text{max}} \cdot |e_{it}|^\delta$$

3 Experimental Evaluation

The performance of the strategy was evaluated through experiments performed in an event-driven simulator, developed for this work. The simulator allows the construction of network topologies with properties following power-law distributions. This type of property is found in networks such as the Internet at domain level [6] and the Gnutella P2P network [11]. The initial topology is generated using the algorithm presented in [6]. The simulator also allows the scheduling of events, given in files which describe nodes and links being added to or removed from the topology. The events may refer to specific nodes and links or to “wild-cards” (e.g. random node, node with most neighbors, etc.).

The most important metrics used to evaluate the proposed strategy are the description convergence and divergence, which quantify the difference between the topology descriptions in the nodes and the real topology. The convergence is a measure of how many links that are part of the network appear in a topology description at a given time. For example, if a node knows 80% of the links in the network, its topology description has a 0.8 convergence. The divergence measures the amount of information in the description that does not exist in the network. These errors appear when a link fails or ceases to exist and the description becomes outdated. Thus, if a description has 100 links but 10 of them are not in the network, the description has a divergence of 0.1.
In an ideal situation, all the nodes have descriptions with convergence equal to 1 (containing the entire topology) and divergence equal to 0 (no erroneous information).

The experiments always consider the average convergence and divergence for all the nodes. The same set of parameters was used in all the experiments. These parameters were chosen empirically, using a 100-node static network as a model. The maximum age a piece of information can have before being discarded by nodes and agents is 1 minute. The time an agent needs to migrate between two nodes is a random value uniformly distributed between 1 and 500 ms. An agent stays idle for 200 ms before migrating. The parameters for the pheromone-based destination selection and population control are: $\alpha = 5$, $\rho = 0.01$, $\gamma_{\min} = 0.1$, $\gamma_{\max} = 3.4$, $\delta = 1.6$. Unless stated otherwise, the pheromone-based destination selection was used.

### 3.1 General Evaluation and Migration

The first experiment compares the proposed methods for the selection of agent destinations: random and pheromone-based. It also shows the quality of the obtained topology descriptions. Ten networks with 100 nodes and 10 networks with 200 nodes were generated, with different initial topologies, and events occurring in random intervals. In the 100-node networks, the intervals are of up to 5 min. for node creation, up to 10 min. for node removal, up to 3 min. for link creation and up to 6 min. for link removal. In the 200-node networks the intervals are respectively of up to 4 min., 8 min., 2 min. and 4 min.. Despite their dynamic size, the networks will be referred to as 100 and 200-node networks.

Simulations were performed on each network using both destination selection methods. Figure 1 shows the average divergence for the 100-node networks: no specific pattern can be observed. The overall average was 0.0064 for the random selection and 0.0057 for the pheromone-based selection.

For the 200-node networks, the averages were 0.004 and 0.0041, respectively; also with no observable pattern. These results indicate that the selection method has little influence over the description divergence.

Divergence was below 0.01, or 1% in most cases, indicating the descriptions usually have little erroneous information, compared to the total amount of information. There is a clear difference between the average values for the 100 and 200-node networks. This occurs because networks with more nodes and links have more information available, reducing the impact of each piece of erroneous information.

Figure 2 shows the average convergence obtained for the 100-node networks. Overall average was 0.9903 for the random selection and 0.9977 for the pheromone-based selection. The graphic for the 200-node networks is very similar, with the respective averages being 0.9909 and 0.9984. The standard deviation was about 3 times higher for the random selection, indicating a greater variation in the convergence.

A high convergence was obtained: if the pheromone-based destination selection is used, a random node selected at any time will know about 99.8% of the real network topology. Besides, for a given destination selection method, the convergence was very similar for the 100 and 200-node networks. This was confirmed by a test-t. The test indicated a statistical equality between the convergences in the 100 and 200-node networks, despite their differences in size and dynamism. The results also indicate that the pheromone-based destination selection produces descriptions with higher convergence than the random selection — hypothesis confirmed by another test-t.

---

1 Statistical test that tells if two sets of observations can be considered equivalent in a given error interval — the interval was of 1% in all the tests performed for this work.
3.2 More Complex Settings

Two experiments were performed to evaluate the strategy in more complex settings. In the first one, the networks have the same initial topologies as the 100-node networks from the first experiment, but are more dynamic: the maximum time between events was divided by 10. The second experiment uses networks with 500 nodes, with the same dynamism as the 100-node networks.

The average convergence for the 500-node networks was 0.9997, and the average divergence was 0.0016. The convergence was very close to what was obtained with the 100 and 200-node networks, and the divergence, as expected for a larger network, was reduced. Those results suggest that the strategy can scale well to larger networks.

In the more dynamic networks, the average convergence was 0.9896, and the divergence 0.0497. There was a loss in the convergence — that, despite of this, is still high — and a great increase in the divergence, which reached in some cases values around 0.09, or 9%. This performance loss is expected, as a greater dynamism implies in more outdated information. All the equalities and differences were confirmed by tests-t with a 1% error interval, which compared these results to the ones obtained in the first experiment.

3.3 Population Control

We must also check if the proposed approach for agent population control creates an adequate number of agents. In fact, the previous experiments already show that: in the experiments with the 100, 200 and 500-node networks, the simulations started with no agents at all, and a high convergence was obtained, with similar results even for differently sized networks. The median\(^2\) of the number of agents is of 27 agents for the 100-node networks, 50 agents for the 200-node networks, and 102 agents for the 500-node networks. Furthermore, the size of the networks in these experiments tend to grow as time passes. Figure 3 shows the average number of nodes and links and the median of the number of agents used in the 100-node networks. It can be seen that the network size increases with time, and that this is followed by an adjust in the agent population. The same behavior was observed in the 200 and 500-node networks.

To better illustrate the agent population adapting to a dynamic network, 10 simulations were performed in networks that start with only 10 nodes. The network grows abruptly to 130 nodes, and after some time, is reduced to 70 nodes. Each simulation was then repeated with a static population of 14 discovery agents, with the agents being uniformly distributed in the network when the simulation starts. Figure 4 shows the average number of nodes and the median of the agent population size as time passes.

\(^2\)The median was used because of the great variance in the results.

4 Related Work

Even though several solutions have been proposed for topology discovery, they are usually centralized [4, 3], and only a few consider decentralized and dynamic networks. Agent-based strategies are even more uncommon.

Worth of special mention is the strategy presented in [7], in which mobile agents discover the topology of a mobile ad
hoc network. The concepts of information aging and connectivity convergence presented in that work were adapted for the strategy presented here. Both were redefined, though their basic idea and role remain the same. Though highly desirable, a comparison between our strategy and that approach is not feasible, due to fundamental differences in the environments and assumptions: that strategy can only be used in ad hoc networks, assumes a static size for the network and agent population, and have counters which are started simultaneously in all the nodes. Our model has more realistic assumptions, with a dynamic size for the network and population, and no synchronization between the nodes. For that reason, that strategy will probably perform better than ours in the scenario they proposed — on the other hand, it does not work at all under our set of assumptions.

In a broader scope, the central idea of our strategy — agents disseminating information among the nodes in a dynamic network — can also be found in other applications. For example, in decentralized P2P networks, a node locates resources by sending query messages that must reach as many nodes as possible with minimum overhead [12]. There are agent-based “rumor mongering” approaches (e.g. [2]), which encapsulate the messages in agents able to communicate through stigmergy. The same principle can also be used for the creation of routing tables [1]. Indeed, in this case the agents carry information about network topology, and the task bears a high resemblance with topology discovery. The idea of agents using pheromones to “flee” from each other, employed by our strategy, was first presented in [1], but the equations and other aspects are different. Once again, these approaches cannot be directly compared with our strategy, given the different applications, environment and assumptions.

5 Conclusion and Future Work

This work presented a new strategy based on mobile agents and swarm intelligence for topology discovery in decentralized and dynamic networks. This scenario has characteristics which do not allow the use of traditional techniques, either centralized or distributed. Moreover, our strategy can work over a more realistic set of assumptions than previous work. We defined metrics of description convergence and divergence, and performed experiments which show that the obtained topology descriptions are very close to the actual topology.

Two approaches were proposed for the selection of agent destinations: random and stigmergy-based, with the experiments showing that the latter produces better results than the former. We also proposed a technique for agent population control, and show that it is able to adjust the number of agents automatically as the size of the network changes. The techniques used for selection of agent destinations and population control depend only on information local to each node, with no additional communication between the nodes.

Future work includes comparing the proposed approach with event driven-strategies, e.g. flooding the network with notification messages when the topology changes. Relaxing the assumptions, and evaluating the approach in larger or real networks is also planned as future work. The strategy can be also enhanced so that agents disseminate other types of information, such as services offered by the nodes. This can lead to more complete network descriptions, useful for specific applications.

References