BlåtAnt: Bounding Networks' Diameter with a Collaborative Distributed Algorithm

Amos Brocco *, Fulvio Frapolli, and Béat Hirsbrunner

Department of Informatics, University of Fribourg, Switzerland
{amos.brocco,fulvio.frapolli,beat.hirsbrunner}@unifr.ch

Abstract. In this paper we describe BlåtAnt, a new algorithm to create overlay networks with small diameters. BlåtAnt is a fully distributed and adaptive algorithm inspired by Ant Colony Optimization (ACO), which targets dynamic and evolving networks without requiring a global knowledge. Simulation results show that our approach results in networks with a bounded diameter. This algorithm, implemented and empirically tested, will be the foundation of a fully decentralized resource discovery mechanism optimized for networks with small diameters.

1 Introduction

The "small-world phenomenon", first observed during social studies by Stanley Milgram in the 1960's [1], reveals that people are linked by short chains of acquaintances; that observation gave birth to the myth of "six degrees of separation". Although not all experiments done by Milgram were successful, the interest on the topic increased, and through further research many other real-world networks (such as plane routes, power grids, etc.) were found to be instances of small-world networks. In mathematics, small-world problems concern graph theory and the distance (number of edges) between vertices in a graph. Research focuses on modeling such graphs, and reproduce their main characteristics, such as short diameters[1]. In computer science, being able to reach any node following a short path is particularly interesting in distributed systems such as P2P networks or grids, even without implementing all features of small-worlds. Resource discovery is a good example of a problem that benefits from short paths.

This paper presents BlåtAnt, a distributed algorithm that augments an existing network with a minimal number of new logical links in order to minimize its diameter. The algorithm will be applied to a grid system to support efficient monitoring and resource discovering; in particular, by lowering the network diameter, we aim at being able to query virtually every peer on the network in a minimal number of hops. Our contribution is based on the collaboration between different types of mobile agents inspired by ants. In contrast to some existing algorithms, the one we detail in this paper does not enforce a fixed topology,

---

* Research supported by the Swiss Hasler Foundation ("ManCom Initiative", project Nr. 2122).

[1] The diameter of the graph being the maximum of all shortest paths' lengths.

is completely distributed and decentralized, and does not require any kind of global knowledge of the network.

The rest of this paper is organized as follows: section 2 presents research in the topic of graph augmentation algorithms, and resource discovery in small-diameter networks. Section 3 presents a description of the underlying rules used in the algorithm. Section 4 details the BlåtAnt algorithm. Section 5 provides an empirical evaluation of the distributed algorithm in static and dynamic scenarios. Finally, section 6 presents conclusions on the work done and some insights on future research directions.

2 Related Works

In our vision, the BlåtAnt[2] algorithm is the base for a fully decentralized resource discovery protocol that exploits the small-diameter of the network.

Although there exist various models for generating small-worlds or networks with simply a bounded diameter, see for example [2], there are only very few distributed algorithms to achieve what is commonly called small-worldization of a generic network. One of these few examples is [3], which proposes a distributed algorithm for the construction of networks with small diameter by adding a single additional link to each node. Even though this algorithm is not directly comparable to the work presented in this paper, it shows that decentralized construction of small-diameter networks based only on local information of the original topology is indeed possible.

By turning the focus to research done on P2P and grid networks, it is possible to find many distributed systems [4, 5] that try to keep short distances between nodes in overlay networks even without referring them explicitly as small-worlds. These solutions are generally geared toward the problem of resource discovery, and the efficient routing of queries.

Current distributed solutions are commonly based on two techniques: distributed hashtables (DHTs) and flooding. Whereas resource discovery using DHTs [4, 5] forces structured networks in order to optimally partition the resource space and intelligently route queries to nodes that are likely to store the desired information, flooding is used mostly in unstructured networks and from a simplistic point of view, it involves querying as many nodes as possible.

In a more general way, the problem of decentralized search in small-world networks has been analyzed in [6, 7], and some examples of construction of overlay networks with small-world properties are reported. Other approaches [8–10], let the small-world phenomenon emerge by clustering peers with similar information, whereas [11] constructs a small-world overlay network to improve the availability of resource under heavy loading. As pointed out in [12], hierarchical solutions work well for static content, but typically suffer from the mutable and heterogeneous nature of resources shared in a grid. Thus, resource discovery in unstructured and dynamically evolving networks is usually performed using

---

[2] From Harald Blåtand, the king thought to have reunited Denmark, Norway, and Sweden under a unique kingdom.

flooding algorithms. The detailed analysis on flooding mechanisms in [13] reveals that a requirement to avoid large network overheads is to limit the search space and prevent forwarding multiple copies of the same query. An *a priori* knowledge of the diameter may be used to restrict the maximum distance traveled by queries, thus limiting one of the problems of flooding.

In the same spirit, the BlåtAnt algorithm constructs and maintains an overlay network with short diameter, to provide a foundation for an optimized flooding-based resource discovery mechanism. In contrast to similar approaches, the construction step is completely separated from the resource discovery task, and it is independent from the underlying topology, the distribution of the resources, and their type. This way, beside resource discovery, other kind of distributed algorithms benefiting from the small diameter can be implemented.

3 General Idea

The BlåtAnt algorithm is meant to be executed in a distributed way on a network which will be represented, without loss of generality, as a finite graph G. In this section the general idea behind the proposed algorithm is introduced.

The goal is to augment an existing network (either physical or logical) with a minimal number of additional logical links in order to bound its diameter into a certain interval determined by a parameter D. For the rest of this paper we will refer to this process simply as *rewiring*. By applying these rewiring rules in any order, for any undirected finite graph G, with a global knowledge and in a finite number of steps, it is possible to produce a graph with a diameter less than $2D-1$; a formal proof is provided in [14].

3.1 Connection and Disconnection Rules

Our algorithm rewires the network according to two simple rules. The first rule is used to create an edge if the distance between two nodes is greater than a fixed threshold. The second rule is used to remove those edges that do not contribute to the solution. These rules only depend on a single integer parameter $D > 0$.

Rule 1 (Connection Rule). *Let n_i and n_j be two non-connected nodes in the network graph G, and $d_G(n_i, n_j)$ the minimal distance from n_i to n_j in G. We connect n_i to n_j if:*

$$d_G(n_i, n_j) \geq 2D - 1 \tag{1}$$

Rule 2 (Disconnection Rule). *Let n_i and n_j be two connected nodes in the network graph G, $i \neq j$. Let $G' \leftarrow G \setminus \{n_i\}$, and N_i be the set of all nodes adjacent to n_i. Node n_i is disconnected from $n_j \in N_i$ if:*

$$\exists\ n_k \in N_i, k \neq j \ : \ d_{G'}(n_j, n_k) + 1 \leq D \tag{2}$$

From the definition of Rule 2 it is clear that, for $D > 2$, the resulting graph has a clustering coefficient equal to zero. In other words, graphs created with our algorithm will not have full small-world characteristics.

4 BlåtAnt Algorithm Description

This section provides a description of the BlåtAnt data structures and algorithm. The idea is to globally optimize the network through successive local optimizations done by single nodes. Each node in the network executes independently by creating new ants and evaluating the creation or deletion of links; its actions are based only on local information about other nodes, which is updated by mean of ants wandering across the network. A detailed description of the algorithm is provided in [14].

4.1 Node Data Structures

The algorithm requires each node n_i to maintain some data structures containing local information and data produced during the execution.

Alpha Table $\boldsymbol{\alpha_i}$ Each node n_i has a α_i table which is used to store local information about the network. This table is constantly updated by mean of the information gathered through the activity of ants. As the table has a fixed maximum size, old information is purged from the table depending on the time it was last updated. Each entry contains information about another node $n_j, j \neq i$, such as the estimated distance, neighbors, local time, and remote time.

Beta and Gamma Pheromone Trails Ants coming from n_j increase the pheromone concentration $\beta_i[n_j]$ on n_i. If this pheromone evaporates completely, n_j is assumed to be dead, and a disconnection procedure is initiated. Conversely, when an ant moves from n_i to a neighbor n_j, trail $\gamma_i[n_j]$ is reinforced; trails with high concentration becomes less desirable, thus a complete coverage of the network is ensured.

4.2 Discovery, Link and Unlink Ants

We distinguish three species of ants: Discovery Ants, used to collect and spread information about the network, Link Ants, used to perform connections between two nodes, and Unlink Ant, used to disconnect nodes.

At regular intervals, with some probability each node generates a new Discovery ant. This, combined with a maximum ant lifetime (number of wandering steps), regulates the ant population and prevents complete extinction in the event of node or network crashes.

Regardless of their species, all ants can only access local pheromone trails and information, and remember the details of the node n_j where they come from (i.e. neighbors N_j, and timestamp t_j). Information about the last visited node is handed out to the current node and used to update the Alpha table. Ants have a limited lifespan that is determined by their mission.

4.3 Frozen Connections

As methods to recover from network disconnection have not yet been implemented in the algorithm, to avoid accidentally disconnecting the network, we *freeze* user-created links (including links existing before the execution of the algorithm), i.e. we do not allow the algorithm to remove them.

4.4 Timing and Pheromone Reinforcement and Evaporation

Each node n_i maintains a logical time t_i proportional to the number of incoming and outgoing ants. Using such a logical time instead of real time regulates pheromone evaporation according to the traffic, and prevents nodes with limited ant flows from clearing their information too quickly.

4.5 Algorithm Phases

The algorithm is divided in four phases: inform, evaluation, connection, and disconnection. The inform phase is executed continuously by the discovery ants while moving across the network. At regular intervals, each node evaluates if new connections need to be made and if existing connections are redundant, and can be removed. For the rest of this section, we describe the algorithm from the perspective of a node n_i.

Inform Phase During the inform phase, discovery ants collect information and pass it to each visited node n_i, updating α_i.

Evaluating a Connection To determine if new connections are necessary, a node has to evaluate its distance to other nodes, and check if Rule 1 applies. Since this process is based only on local information, the first step is to construct a graph $\tilde{G}$ using the information available in the α_i table and the neighbor set N_i. Then, the distance $d_{\tilde{G}}(n_i, n_j)\ \forall n_j \in \tilde{G} \setminus \{n_i\}$ is computed. For each node n_j satisfying condition (1), a connection procedure is initiated by sending a Link Ant from node n_i to n_j.

Evaluating a Disconnection The process of evaluating a disconnection is similar to the one used for a connection procedure, but it depends on Rule 2. A graph $\tilde{G}$ based on α_i and N_i is constructed, but because frozen connections cannot be removed an additional restricted neighbor set N_i' is used beside N_i. N_i' is defined as $N_i' \leftarrow N_i \cap \Lambda \setminus \{n_j \in N_i \mid \text{link from } n_i \text{ to } n_j \text{ is frozen}\}$, where Λ contains all valid keys found in the alpha table. Thus, N_i' is the set of all neighbors with a non-null entry in the alpha table, and whose connection with n_i is not frozen. For each node in $n_j \in N_i'$, the distance $d_{\tilde{G}}(n_j, n_k)\ \forall n_k \in N_i \setminus \{n_j\}$, is computed, and condition (2) is checked. Eventually, disconnection of a node is achieved by sending an Unlink Ant.

Connection and Disconnection Procedures A node n_i connects to another node n_j by first adding n_j to N_i and then updating the information in the α_i table. Conversely, disconnection from a node n_j is performed by first removing n_j from the local neighbor set N_i, thus preventing Discovery ants from migrating to n_j.

5 Evaluation

To evaluate the BlåtAnt algorithm, we conducted simulations on different topologies (including dynamic ones). We have tested our algorithm with the following topologies: a path graph of 1024 nodes, a 2D grid of size 32x32, a hypercube of 1024 nodes, and a LAN of 1281 nodes [3]. For each scenario, a simulation run consisted of 1280 iterations, where an iteration corresponds to a complete migration of the entire population of ants[4]. Each run was executed 42 times; details on the parameter values used during all tests, as well as additional results can be found in [14]. We present both maximum standard deviation σ_{max} of all topologies at the 1280^{th} iteration, and the maximum mean standard deviation σ'_{max} over all iterations.

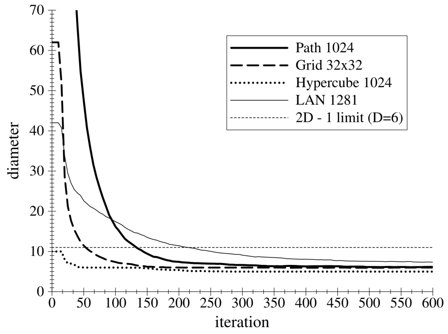

Fig. 1. Convergence of the diameter

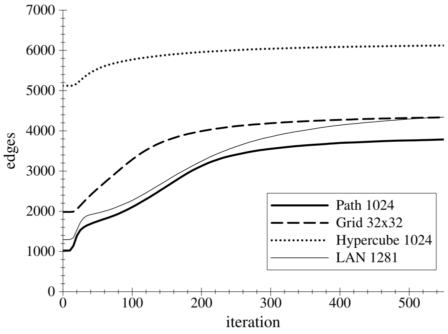

Fig. 2. Number of edges

Convergence Figure 1 shows the evolution of the network diameter in the four considered scenarios ($\sigma_{max} = 0.32$, $\sigma'_{max} = 1.16$). The diameter converges exponentially under the upper bound $2D - 1 = 11$, to a value close to $D = 6$. The *LAN 1281* topology takes more iterations because of its lower degree of connectivity, forcing ants to a longer exploration before nodes with a sufficient distance are found.

---

[3] https://networkx.lanl.gov/browser/networkx/trunk/doc/examples/lanl.edges

[4] Typical execution time per iteration is 200 ms for a population of 500 ants in a 1024 nodes topology on a dual-core 2 GHz.

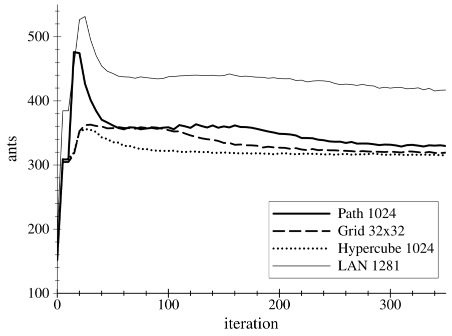

Fig. 3. Ant population size

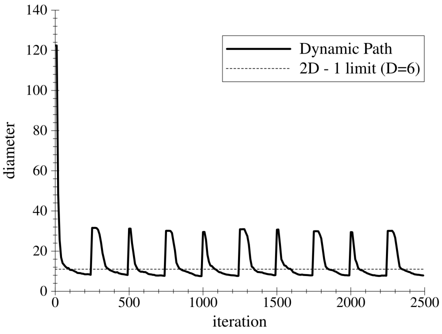

Fig. 4. Dynamic scenario

Graph Complexity A desired property of networks generated by our algorithm is not only small diameters but also a minimal number of edges. We evaluate the minimality of the solution by computing the number of edges during the execution. Results are shown in Figure 2 ($\sigma_{max} = 51.85$, $\sigma'_{max} = 91.24$). By comparing this result with Figure 1 it is possible to notice that the number of edges stabilizes as soon as the diameter reaches its minimum.

Network Load Figure 3 shows the number of ants created by the algorithm during its execution, which can be used to estimate the network load ($\sigma_{max} = 21.97$, $\sigma'_{max} = 21.62$). Each topology starts with roughly the same population of Discovery ants, which can be estimated to 15% of the total number of nodes. When the rewiring process begins, multiple Link and Unlink ants are instantiated. As soon as an optimal diameter is found, the population starts to decrease.

Dynamic Networks Although the BlåtAnt algorithm does not implement any mean of preventing network partitioning when a node disconnects or crashes, we propose an initial test of its behavior in dynamic networks. For that purpose we used an initial path topology consisting of 100 nodes: every 250 iterations a chain of 25 nodes is added to a random node in the graph, and every 50 iterations a node is removed. Figure 4 shows the evolution of the diameter: a minimal diameter is restored in about 100 iterations after a chain was added.

6 Conclusion and Future Works

In this paper we presented BlåtAnt, a collaborative and distributed algorithm inspired by ACO, to bound the diameter of a network without requiring a global knowledge. The algorithm uses different species of ants with different tasks in order to collect and propagate information across the network, and to create and remove links. Pheromone trails are used to direct ants toward underexploited paths, and to detect the departure of adjacent nodes. Simulations on different topologies validated the behavior of the algorithm in a fully distributed environment. Furthermore, when applied to an evolving network, the adaptive nature of the algorithm illustrated its ability to rapidly control the diameter.

There are several issues that are worth further investigation. First, a thoughtful analysis of the algorithm in large scale networks would allow us to validate its scalability. Additionally, although the algorithm performed well in the simulated dynamic network, full fault-tolerance is still lacking.

In conclusion, we believe BlåtAnt can be a foundation for a wide range of distributed algorithms that will exploit the network's shallowness, for example, a flooding based resource discovery, or an optimized network monitoring.

References

1. Milgram, S.: The small world problem. Psychology Today **2** (1967) 60–67
2. Kleinberg, J.: The small-world phenomenon: An algorithmic perspective. In: Proceedings of the 32nd ACM Symposium on Theory of Computing. (2000)
3. Duchon, P., Hanusse, N., Lebhar, E., Schabanel, N.: Towards small world emergence. In: SPAA '06: Proceedings of the 18th annual ACM symposium on Parallelism in algorithms and architectures, New York, ACM (2006) 225–232
4. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable content-addressable network. SIGCOMM Comp. Com. Rev. **31**(4) (2001) 161–172
5. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the 2001 ACM SIGCOMM Conference. (2001) 149–160
6. Kleinberg, J.: Complex networks and decentralized search algorithms. In: Proceedings of the International Congress of Mathematicians (ICM). (2006)
7. Sandberg, O.: Searching a small world. Master's thesis, Chalmers University (2005)
8. Zhang, H., Goel, A., Govindan, R.: Using the small-world model to improve freenet performance. INFOCOM 2002. 21st Annual Joint Conference of the IEEE Computer and Communications Societies. **3** (2002) 1228–1237 vol.3
9. Akavipat, R., Wu, L.S., Menczer, F.: Small world peer networks in distributed web search. In: WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, New York, NY, USA, ACM (2004) 396–397
10. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: A distributed anonymous information storage and retrieval system. In: Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability. (2000) 46–66
11. Hui, K.Y.K., Lui, J.C.S., Yau, D.K.Y.: Small-world overlay p2p networks: construction, management and handling of dynamic flash crowds. Comput. Netw. **50**(15) (2006) 2727–2746
12. Iamnitchi, A., Foster, I.T.: On fully decentralized resource discovery in grid environments. In: GRID '01: Proceedings of the 2nd International Workshop on Grid Computing, London, UK, Springer-Verlag (2001) 51–62
13. Dimakopoulos, V.V., Pitoura, E.: On the performance of flooding-based resource discovery. IEEE Trans. Parallel Distrib. Syst. **17**(11) (2006) 1242–1252
14. Brocco, A., Frapolli, F., Hirsbrunner, B.: Shrinking the network: The blatant algorithm. Technical Report 08-04, Department of Informatics, University of Fribourg, Fribourg, Switzerland (2008, http://diuf.unifr.ch/pai)