Biologic-inspired Routing Algorithms for Distributed Collaborative Mobile Applications

Tiago Santos, Hélder Vieira, Teresa Vazão, Pedro Diniz
Instituto Superior Técnico / Tagus Park / INESC-ID
Technical University of Lisbon / Tagus Park
2780-990 Porto Salvo, Portugal
{tiago.casimiro,holder.vieira,teresa.vazao}@ist.utl.pt

João M.P. Cardoso
Department of Informatics Engineering
Faculdade de Engenharia da Universidade do Porto (FEUP)
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
jmpc@acm.org

Abstract—Mobile devices such as cell-phone and embedded systems have continuously increased in computing and sensing capacity. It is now practical to interconnect via wireless many of these devices enabling them to collectively cooperate in sophisticated discovery and service providing tasks. These applications, however, still need to address challenges such as unreliable and unsecured networks, transient distributed and centralized computing failures to deliver robust and reliable services. This paper describes an experience in building and evaluating a software and hardware architecture using MindStorm robots that communicate with cell-phone and a centralized "station". Collectively, with their own sensors these devices discover a simple scenario along with their obstacles. We present results from two types biologically-inspired message routing algorithms, namely and epidemic-based algorithm and an algorithm based in the behavior of ants. The results reveal that the ant-inspired algorithm tends to perform well in larger areas as it allows for multi-hopping of messages to reach remote nodes, whereas epidemic-inspired algorithm tends to perform well in more localized settings where the transmission capacity of the nodes reaches all nodes in that region thus avoiding the need for maintaining routing tables.

I. INTRODUCTION

The relentless pace of system miniaturization alongside the continued advances in mobile communication systems and embedded systems have enabled mobile devices such as smartphones to be truly ubiquitous computing and communication platforms. Even modest devices include WiFi and/or bluetooth interfaces, internal sensors such as accelerometer and GPS, high-resolution displays and internal processor computing devices. These enhanced capabilities have allowed the rapid development of sophisticated applications, not only as passive devices (i.e. as stand-alone receivers) but more importantly as cooperative agents in distributed computations such as social-networking.

The versatility and sheer computing power of current mobile devices present tremendous opportunities. At the individual level they can be simple extensions of existing networks to deliver services catered to each user’s needs. With the help of sensors they can provide an important role in individual health-care monitoring devices. In addition, their role as active nodes, allows them to participate in collective tasks, possibly in an opportunistic fashion, such as message relaying nodes when global network coverage is limited. These opportunities enabled by these devices, however, present numerous challenges, most notably, security, service level or performance and consequently energy use.

In this paper we focus on the experimentation and evaluation of biologically-inspired message routing algorithms in a system that includes mobile communication devices, such as smartphones and robots, to collectively undertake a distributed task. As depicted in figure 1, the system is composed by multiple mobile nodes each of which consist of one smartphone and one mobile robot. The devices communicate via are bluetooth and WiFi and are programmed using J2ME technology to implement two different biological routing algorithms, respectively, an ant-inspired and an epidemic-inspired message routing algorithm. We evaluate key message transmission parameters such as calculate processing and communication time, number of packets necessary to send data and quantity of exchanged data. We further evaluate the resilience of both algorithms in a limited scenario, to the momentarily loss of one node in the network.

This article is organized as follows: Section II describes the biologically-inspired message routing algorithms used in our work. In Section III we describe our system hardware, software architecture and routing algorithms. In Section IV we outline the implementation of a specific collaborative application used in our experimental evaluation described in Section V. We cover related work in Section VI and conclude in Section VII.
II. Bio-inspired Message Routing Algorithms

Biological systems either in the form of colonies of individuals (nodes in our analogy) or simply an amorphous gathering of organisms with the same structure, exhibit features that are desirable for ad-hoc routing networks, providing a model for their message routing algorithms [1] [2], namely:

- Concurrency and Distribution: The colony is distributed and thus without a centralized control entity or point of failure. Natural concurrency allows members to process the information independently and in parallel thus improving overall system performance.
- Robustness: Some members may fail or even die leading to a temporary reduction of system performance, or response time, but not affecting the functionality of the system. This is achieved at the expense of redundancy and/or increase complexity as members need to adapt to environmental changes.
- Autonomy: The members of the colony can modify the environment according to their perceptions and stimuli. In the face of critical events they can react and change overall mission goals.
- Scalability: The sophistication of the members allow them to develop adaptive strategies to cope with increasing complexity of problems. Examples include hierarchical structure that allows them to maintain overall order and functionality.

The work described in this paper focuses on the use and evaluation message routing algorithms for mobile robots. The routing of messages between robots is accomplished using biological-inspired algorithms, respectively, an ant-inspired algorithm [3], [4] and an epidemic-inspired algorithm [5], [6].

A. Ant-inspired Routing Algorithm

In the ant algorithm, depicted in Figure 2, each node has a list of possible routes to reach any other node in network using intermediate nodes. When communicating with other nodes data packets are preceded by a specific header message (also called Forward Ant or FANT). In a starting phase of the overall algorithm each robot sends a presentation message to all other robots in network advertising its identity. Associated with each route link (or section) there is a weight (also named pheromone) which is incremented each time a FANT or data packet uses that specific link. The weights are decreased by a periodic process that mimics decay of the chemical intensity of the pheromone. With this adaptive approach, the good connections persist whereas less efficient connections eventually disappear.

As ad-hoc network evolve over time, another process is required to allow for self-discovery of a new preferred topology. As such periodically new FANT are broadcasted allowing nodes to be directly reach, which were previously only reachable via multi-hop.

B. Epidemic-inspired Routing Algorithm

In the epidemic-inspired routing algorithm, depicted in Figure 3, a node establishes communication only with nodes that are within its range. Every time two nodes come in "contact" or proximity, they exchange unseen messages. In this approach, and while it is not possible to explicitly reach nodes outside each nodes’ vicinity, messages eventually reach other nodes by hopping through intermediate nodes.

The software application in each of these devices focus on the software application in each of these devices and the corresponding algorithms they implement in the space discovery application.

A. Hardware Devices

Our system includes two types mobile devices, namely smartphones and mobile robots.

We use Nokia N78 and Nokia N80 smartphones as the main computing nodes as they provide ample processing capacity. They support J2ME (Java 2 Micro Edition) for application development. Special attention was devoted to input/output given the limited screen capabilities of these devices. The application we have developed records data throughout its execution and communicates with an external PC reporting the information independently and in parallel thus improving overall system performance.
overall execution metrics such as sequence of issues control commands and networking loading and packet routing statistics. This allows for a smoother operation of the control algorithm (described in section IV-C) given the mobile robot real-time control requirements.

Considering that the aim of this project does not involve the use of complex and robust hardware. Regarding the mobile robots, we opted for the relatively inexpensive and simple Lego Mindstorms NXT . Each mobile robot includes various sensors, motors and a processing central unit (the NXT brick), responsible for all Bluetooth and USB communication.

These mobile robots support the inclusion of special sensors to assist in exploring the environment and through which one can acquire data about the environment. The data each robot collects is sent to the controlling smartphone. The smartphone uses this information to determine and send control messages to each robot, thereby controlling its movements.

In this work we made use of colored markings and specific marker objects to aid the robots in the detection of objects. The identification of landmark and specific domain features is an issue orthogonal to the work presented here.

B. Software/Communication Architecture

Overall the mobile robots are controlled in real-time by a single smartphone which assumes the key role as the main processing and decision unit. Smartphone supports the inter-robot communication as well as controls the robots movements. In current implementation, mobile robots communicate exclusively with the smartphone via a Bluetooth interface.

Mobile robots operate as a platform for testing algorithms and application development. This capability provides a very flexible development and testing facility for the visualization of the behavior of algorithms implemented and also the discovery of any errors, when compared with simulation.

Figure 4 depicts the software components of the implementation which includes a set of modules, namely:

- On the smartphone two communication modules, respectively a Bluetooth for communication with the Mobile robot and a WiFi for communication with the other Mobile robots.
- Configuration module used to configure the network interface of the smartphone, to allow the establishment of an ad hoc network, by allocating a unique IP addresses to each device.\(^2\)
- Routing module manages messages in the ad hoc network. This module includes a server application to respond requests and receive data from other Mobile robots and a client application to send data to other robots. In this module it is important to understand the minimum amount of information vital to implement the solution without failure.
- Path planning module that is responsible for the physical movements of each mobile robot (advance, rotate or stop). This module read mobile robots’ sensors (touch, ultrasound and color sensors) and makes very limited decisions according to sensor values. For instance of the ultrasound detects an object ahead no forward movement is possible.

![Software Architecture](image)

An advantage of using an ad hoc network lies in the simplicity in the creation of a network without physical routers, or any kind of backbone. The organization and control of the network is distributed among the nodes, as there is no predetermined topology nor centralized control. The ad hoc network we have implemented is asymmetric, because the ability of communication vary from node to node, and it is influenced by the radius and processing capabilities of each node. Each node communicates through TCP sockets rather then using UDP sockets given its reliability. Nevertheless, an additional overhead is achieved, due to the connection setup. Using a reliable mechanism on the top of UDP may solve this issue, but introduces a significant processing time which is not adequate due to the limited resources of the smartphones.

IV. Applications

Our application is built on the Java platform and organized as two cooperating entities, one residing in the smartphone another in each of the mobile robots.

A. Smart-Phone

As mentioned previously, the application for smartphone will be developed according to J2ME technology. This will be the main system’s application. It collects data received by robot’s sensors and makes decisions according to values obtained, communicate with other smartphones using ant or epidemic algorithms to propagate messages through network and generates a map with mobile robot’s positions.

Through Wi-Fi communication between smartphones, is possible to change maps with positions of all mobile robots, which are aggregated into a single map. This feature allows the

\(^2\)For this module were used network options defined in Nokia’s firmware, as the firmware installed will not allow the application developed under this project set some options.
systems cooperate with each other and create environment’s map they are exploring more quickly.

B. Mobile Robot

As mentioned above leJOS firmware was installed in mobile robot’s NXT brick, to replace the original Lego firmware. This firmware is based on Java programming language. The far NXJ provides a well documented API that facilitates the development of the application to communicate with the smartphone via bluetooth technology. This firmware includes a Java virtual machine (JVM).

The application developed for the mobile robot is a server whose function is to send data collected by sensors to smartphone as well as receive and interpret the orders to move.

C. Map Application

We have implemented a specific communication protocol to allow each mobile robots to exchange information of its own virtual world with the others. When mobile robots start their activity, smart-phones are not synchronized because we do not want to have them transmitting simultaneously, as we want to minimize collisions. Thus, each robots periodically exchange with its neighbor a message containing all the new positions it had explored, since the previous message. Each message only contains the coordinates (X,Y) of the position, the physical state (0 - means free position, 1 - means occupied and 2 - means the target object) and a counter that represents the number of times the robot had explored that position. To avoid overload the network, no confirmation message is sent, as we are using TCP and thus messages are reliable received.

As an example, consider a scenario in which robot A has explored positions (1,1) and (1,2) and had transmitted them in a message to the its neighbors robots B and C. After this step, robot A explored the positions (2,2), (2,1) and (1,1). If there are no modifications in the map regarding the first positions (1,1) and (1,2), the second message will only carry the information for the positions (2,2) and (2,1). This strategy of incremental data transmission promotes small messages and thus minimizes the probability of collisions.

V. EXPERIMENTAL RESULTS

We evaluate the performance of the routing algorithms described in section II for a scenario with up to 3 mobile nodes. The main goal of this experimental evaluation was to identify the strengths and weaknesses of each of these algorithms and determine which types of routing algorithms are better suited for a wireless ad-hoc network communication as a platform for robotic exploration.

We evaluated both routing algorithms for an application whose objective is the discovery of the topology and possible obstacles in a shared two-dimensional space as described in section IV-C. The testing scenario consisted of a large convex surface divided into equally shaped 160 squares. Each square denoted a position of the space that can be represented in the discovery map as either empty or occupied. Each robot moves autonomously to create its own virtual map of the space, in a collaborative fashion and use sensors to avoid collisions.

A. Impact of Data Volume in Communication Time

Given the limited size of our setting and the relatively large range of wireless network cards for mobile devices, the implementation is insensitive to signal strength. Any two mobile robots can communicate from any position of the space and signal strength is in practice constant through the space. Given this initial experiments, we have emulated multi-hop scenarios by artificially omitting a communication to one or more mobile devices as described in section V-C.

The chart in figure 5 also presents transmission’s time between a computer and any mobile device. We attribute lower transmission’s time by computer due to a higher processing and network card capacities.

As the transmission time of individual packets is in practice constant, the time to transmit a message with map data is thus linearly proportional to the number of data packets that compose that message. The figure 6 depicts the relationship between the number of required packets for each set of map positions transfered. Given this linear relationship the transmission time between nodes is this proportional to the number of map positions transfered.

B. Algorithm Evaluation

We now turn to the evaluation of each of the two route-finding algorithms.

In figure 7 we report the time observed for the transmission of a varying number of positions between two mobile devices. The number of positions range from 1 to 10, and thus corresponding to a single packet (see fig 6). Given that the mobile
devices communicate frequently and do not explore the space very quickly, the number of effective positions exchanged are always in the range of 1 to 10.

As can be observed there is no significant difference between the two algorithms. The epidemic-inspired algorithm out-performs the ant-inspired algorithm by an average transmission time of 6%. We observed a similar trend for larger number of positions transmitted (graph not shown for space considerations). We attribute this slight performance degradation of the ant-inspired algorithm to the higher transmission times incurred from the above implementation of the algorithm and its initial state of route discovery. When the number of positions to be transmitted increases, the difference between the algorithms is smaller as a substantial portion of the transmission time is occupied with packets and not with the route discovery phase of the ant algorithm (similar to ARA - Ant Routing Algorithm presented in [4]).

Fig. 7. Comparison of algorithms transmission time for low number of transmitted positions.

We have also conducted scalability experiments for both algorithms to understand how the transmission time of these algorithms behaved in the face of increasing number of communicated map positions. The chart depicted in figure 8 reveals that both algorithms exhibit a linear relationship between transmission time and number of positions transmitted. Again the epidemic-inspired routing algorithm exhibits a light advantage which we attribute to both the lower algorithmic overhead and the fact that in the experiments scenario all mobile devices are within communication range, thus exploiting the multi-hop capability of the ant-inspired algorithm.

C. Message Hopping

Using message hopping an application such as environment discovery can cover a large physical area with a limited number of mobile devices, as each device can receive information from another mobile device even if they are not within range.

We now focus on the evaluation of the message hopping transmission strategy for the ant-inspired algorithm, as of the two routing algorithms explored in this paper it is the only one that directly supports it. The ant-inspired algorithm allows nodes to communicate via multi-hopping by explicitly maintaining an internal routing table that direct messages through intermediate nodes. This feature, however implies an increase of complexity in discovery and maintenance of routes. Provided that there is a possible path by which all mobile nodes can exchange information, the ant-inspired algorithm is expected to behave better than the epidemic-inspired algorithm in scenarios of large dimensions.

To evaluate the performance of the ant-inspired routing algorithm with message hopping we forced specific nodes to communicate via an intermediate node. Specifically whenever robot A wanted to send a message to robot B it would send a message through robot C.

The chart in figure 9 depicts the effect of routing through an intermediate node in the transmission time between two nodes. For this experiments we used messages that transmitted a large number of positions, in this case 160 positions that corresponded to the entire space to be explored.

Fig. 9. Multi-hop transmission time increase.

As can be seen the cost of transmission increases from an average of 70ms to 150ms and generally insensitive to the distance between nodes as all of them are within communication range. To better understand the sources of the additional overhead, other than the need to send 2 messages instead of 1, we observed the increase in the FANT messages or control messages as depicted in Figure 10.

3The ration was computed as the ratio of the time difference over the fastest algorithm, in this case the epidemic-inspired algorithm.
approaches in this class of algorithms avoid the constant send data to a given node of one of the links fails. While are created on-demand, only when it becomes necessary to (DSR) [9] routing algorithms, routing paths between nodes Distance Vector (AODV) [8] and reactive and consumes large bandwidth and energy resources. approach, although conceptually simple, has poor scalabil ity topology has to be propagated, potentially, to all nodes. Th is with all current routes to all nodes. Any changes in network' s approach, however, can easily accommodate this scenario by having an intermediate node either store a message for subsequent relay or discarding it.

VI. RELATED WORK

We now describe related work in the context of Mobile Ad-hoc Networks (MANETs) and their message routing algo rithms. For these networks the classical routing algorithms that rely on periodic updates of routing tables are not well suited as these updates occupy a significant percentage of networks bandwidth. In addition they consume a non-trivial amount of energy, a key concern for mobile devices.

To address the specific constraints of these networks re searchers have developed several routing algorithms that can be divided into three broad categories, namely, pro-active, reactive and hybrid.

In pro-active [7] algorithms nodes keep in memory tables with all current routes to all nodes. Any changes in network’s topology has to be propagated, potentially, to all nodes. This approach, although conceptually simple, has poor scalability and consumes large bandwidth and energy resources.

In reactive algorithms, such as the Ad hoc On Demand Distance Vector (AODV) [8] and Dynamic Source Routing (DSR) [9] routing algorithms, routing paths between nodes are created on-demand, only when it becomes necessary to send data to a given node of one of the links fails. While approaches in this class of algorithms avoid the constant update effort of routing tables, it also leads to undesirable delays in communication when the routes are not present in routing tables.

The work described in this paper evaluates two specific biologically-inspired routing algorithms on a specific communication infrastructure and agents (mobile robots) and a centralized mobile devices (smartphones). Rather than the evaluation of communication metrics for generic computations, we used a space-discovery application as the motivating application behind this work.

VII. CONCLUSION

Ad-hoc communication networks provide a reliable commun ication basis for extremely dynamic and often unstructured environments. In this work we have explored the use of biologically-inspired routing algorithms for ad-hoc networks in the context of a robotic domain-exploration application where mobile robots collaboratively interact to discover the properties of the domain they are immersed in.

In this paper we have focused on two types biologically-inspired message routing algorithms, namely and epidemic-based algorithm and an algorithm based on the behavior of ants. We conducted experimental on scenarios with limited number of mobile robots and enclosed physical domain. The results reveal that the ant-inspired algorithm tends to perform well in larger areas as it allows for multi-hopping of messages to reach remote nodes. Conversely, the epidemic-inspired tends to perform well in mode localized settings where the transmission capacity of the nodes reaches all nodes in that region and thus allows nodes to read consistent routing data.

Despite their complexity, these routing algorithms strike a good trade-off for mobile embedded systems with limited communication and computation capacity.

REFERENCES