Monitoring-based Validation of Functional and Performance Aspects of a Greedy Ant Colony Optimization Protocol

Raul FUENTES*, Ana CAVALLI*, Wissam MALLOULI†, and Javier BALIOSIAN‡
*TELECOM Sudparis, Evry, France. E-mail: {fuentess, ana.cavalli}@telecom-sudparis.eu
†Montimage, Paris, France. E-mail: wissam.mallouli@montimage.com
‡ University of the Republic, Montevideo, Uruguay. E-mail: baliosian@fing.edu.uy

***Abstract*—Delay Tolerant Networks (DTN) are well adapted for situations where the network nodes suffer from intermittent communications due to the high mobility of the nodes and the constantly changing environment. Several research works tried to address this problem and lately, an ants-based protocol named GrAnt, has been proposed as one of the best solutions. In this paper we firstly assess GrAnt performance in much more challenging conditions than those presented by its authors, and secondly, we present a generic methodology based on MMT, an online security monitoring tool that enables real-time analysis of network traffic, to correlate the performance of a DTN protocol (such as GrAnt) with the information stored in its messages in order to validate the reliability of the protocol.**

I. Introduction

Delay Tolerant Network (DTN) is an approach to computer network architecture that seeks to address the technical issues in heterogeneous networks that may lack continuous network connectivity. DTN was proposed by the NASA in 2007 to model satellite communication [1]. Their initial studies concluded that 1) DTN decreases the labor costs of the operations control center OPEX; and 2) DTN decreases the infrastructure costs CAPEX [2].

In this paper, we present the validation of the GrAnt protocol concerning the functional and performance aspects. The validation has been performed using two tools, the simulator Opportunistic Network Environment (ONE) and a monitoring tool, MMT [10], developed by the Montimage society. Different experiments have been performed on a case study and the results show the advantages and limitations of the GrAnt protocol. In order to illustrate the application of the GrAnt protocol, we consider an underground mining scenario which shows all the previous restrictions. It is assumed that all the workers in a mine carry devices supporting DTN for the transmission of data between deep places in the tunnels and a main server in a central point at the surface. We use the MMT Tool for detecting the social interaction between nodes in the mine and to evaluate the performance of GrAnt based on the messages conditions when reaching the main server.

The paper is organised as follows. Section II presents a short description of the GrAnt protocol. Section III introduces the ONE simulator and the results on performance analysis and Section IV presents the application of monitoring techniques to analyse the behavior of the protocol. Finally, section V gives the conclusions of this work.

II. The GrAnt protocol

The original Ant Colony Optimization (ACO) algorithm, known as the Ant System [6], was first proposed in the early nineties [7], [8] inspired in the manner in which ants mark and choose their paths. In ACO, a number of agents simulating ants build solutions to a given optimization problem and exchange information on the quality of these solutions via a communication scheme that is reminiscent of the one adopted by real ants [8].

The ACO algorithm has been often considered as a good choice for routing Mobile Ad-hoc Networks (MANET), however, the ACO's random nature makes it slow in front of sudden network changes [6].

Figure 1 presents an execution of GrAnt in a small network. Node s sends an FA k with destination d together with a data message m (Figure 1(a)). The path to d is computed based on some knowledge acquired by this FA, which decides where to be forwarded at each node and tries to infer the capability of good next forwarders towards d. While being forwarded, each FA k collects the quality of every node x (Q_x) along the path to d (Figure 1(b)). After reaching its destination, a Backward Ant (BA) is sent in reverse through the path recorded in the FA. The BA stores the followed path's total quality (Q^k_{path}) and deposits a pheromone which is proportional to (Q^k_{path}) at each link of the reverse path from d to s (Figure 1(c)). Shall new messages be forwarded to d, the already deposited pheromone is reinforced, guiding the forwarding of future FAs to d (Figure 1(d)). To direct DTN traffic to the most promising contacts, GrAnt uses information about opportunistic social connectivity between nodes. All the details on how this is performed can be read in [6].

III. GrAnt Analysis using ONE Simulator

In the following GrAnt behaviour analysis we try to asses if GrAnt's implementation satisfies its functional requirements and if it performs correctly with a given application design and configuration.

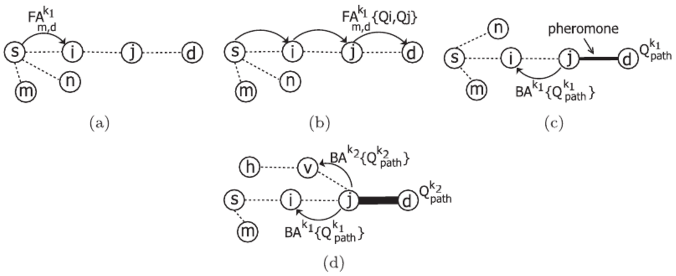
$FA^{k_1}_{m,d}$
s
i
j
d
m
n
(a)
$FA^{k_1}_{m,d}\{Qi,Qj\}$
(b)
pheromone
$Q^{k_1}_{path}$
$BA^{k_1}\{Q^{k_1}_{path}\}$
(c)
h
v
$BA^{k_2}\{Q^{k_2}_{path}\}$
$Q^{k_2}_{path}$
$BA^{k_1}\{Q^{k_1}_{path}\}$
(d)

Fig. 1: Overview of the GrAnt protocol execution (Source [6]).

For the experiments we use the same simulation platform used by GrAnt creators, the Opportunistic Network Environment (ONE) simulator [9]. This simulator is adapted to study DTNs and has the ability of using different realistic mobility models such as Point of interest movement mode (POI) and Working day Movement Model (WD) [9]. We take as a starting point the best settings for GrAnt that are proposed by its authors.

All tests were performed with the same movement patterns, execution time (800.000 seconds), number of messages (11 458), the same number of nodes with the same technology and all moving in a model of the centre of Helsinki. These tests were divided into 3 categories: 1) Stress Scenario; 2) Impact of TTL; and, 3) saturation buffer.

Experiments on Buffer Saturation with Fixed Sizes

While dropping messages due to TTL expiration is a natural element of the algorithm, messages removed due to buffer overflow is not. If this happens too often there is a risk of losing key communication messages in an application. In the worst case, a message could be an FA removed that has not yet expired, has not been duplicated and, thus, its contents will be completely lost.

In this first series of tests, we vary message size (as a percentage of buffer size) keeping buffer size and TTL constant:

- Exp. 12 - TTL: 400 min, BS: 4MB, MS: 5% (200KB)
- Exp. 17 - TTL: 400 min, BS: 4MB, MS: 12% (480KB)
- Exp. 2 - TTL: 400 min, BS: 4MB, MS: 37.5% (1.5 MB)
- Exp. 4 - TTL: 400 min, BS: 4MB, MS: 57.5% (2.3MB)
- Exp. 10 - TTL: 400 min, BS: 4MB, MS: 75% (3 MB)
- Exp. 11 - TTL: 400 min, BS: 4MB, MS: 95% (3.8 MB)

In these tests, we set the message size as a percentage of the buffer. Table I shows the results obtained from the simulations.

If the message size is too small, the number of duplicated FA increases with a small impact on the number of dropped or aborted messages. Remember that duplicate FA ants carry the same messages but follow different paths. The reason that the number of aborted messages has so small impact on the throughput is that when the message is extremely small, any chance of passing it will be sufficient. The contact time may last only few seconds and in these experiments we can effectively see that when the message size increases, the proportion of aborted messages also increases.

TABLE I: Fixed-size messages.

Parameters	EXP 1	EXP 2	EXP 4	EXP 10	EXP 11	EXP 12
Started	57108	31928	22801	18621	16516	78324
Relayed	56975	31293	18037	8232	4672	78275
Aborted	133	635	4764	10388	11844	48
Dropped	29427	34639	27094	18632	15577	4212
Removed	19005	3822	1091	484	252	43265
Delivered	7105	2426	852	426	222	9515
Hopcounts FA	12450	3302	1073	459	235	20949
Hopcounts BA	6422	2055	617	259	108	8461
Drop ratio rate	0.43	0.8103	0.9186	0.9463	0.9657	0.0469
Delivered ratio	0.62	0.21	0.07	0.04	0.02	0.83

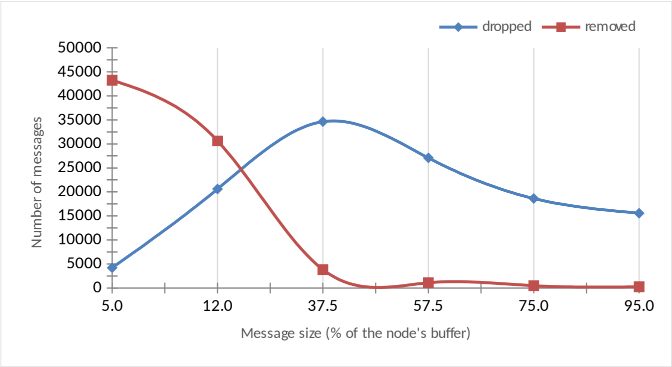

Fig. 2: Comparison between dropped and removed messages (EXP 12).

Regarding removed and dropped messages something interesting occurs. Experiment 12 has the highest value of messages removed and a small number of messages dropped. Regarding messages removed, we can observe that the number decreases when the size of the message is increased. This is not the case for dropped messages which seem to follow this trend but suddenly increases and then decreases. We can better observe the behavior of both fields in Figure 2.

The behavior regarding the number of removed messages as the message size grows is easy to understand. The bigger the messages to be transmitted, the number of messages that fits in the node decreases; therefore, although each new message could force the removal of old messages, the total number of messages generated is also much lower. This is shown by the almost 78% decrease of messages initiated in Experiment 11 (3.8 MB) compared to Experiment 12 (200 KB).

So why did the Experiment 12 with the smaller message sizes have to remove so many ants from its buffers? Note that the total number of generated ants is highest for the messages and virtually all were transmitted. The high number of small messages saturated the nodes' buffers before the TTL of the ants began to expire. That is why in experiment 12 (5%), compared to Experiment 1 (12%) and finally to Experiment 2 (37.5%) the behavior allows the following observation: the larger the messages, the lower number of messages, reducing the number of messages aborted and increasing the number dropped.

However, the peak between experiments 2 (37.5%) and 4 (57.5%) is due precisely to the completely different behavior in the first three experiments. The nodes are not capable of storing more than one message at a time, so that the number of messages removed drastically decreases. But in turn, messages dropped increase at a completely different rate than the first three scenarios.

We could say that applications that need to handle only small messages with respect to buffer sizes give a good result as the percentage of successfully delivered messages increases. However, this is not always possible. Nodes that have huge buffers would be fine but this implies a very high cost. If nodes have a buffer of moderate size (as in the simulations) we would then need to limit the size of messages, which may limit the ability of applications. As we can see, DTN environments require making consensus between the size of the application messages and the size of the buffers of the nodes.

Our conclusion is that GrAnt has to work with a message size of less than a half of the buffer size in order to transmit data at acceptable rates.

IV. Network Traffic Analysis using MMT

A. GrAnt Trace Analysis

Telecom SudParis has designed and implemented a plugin for communication analysis of applications using the GrAnt algorithm for routing messages. This plugin allows extracting and analyzing the following components in the messages:

- **Type** - Identifies type of ant, possible values: FA or BA.
- **ID** - Message ID.
- **I-N** - Identifies intermediary I-Node. When a message is created by first time I-N will be the same as S-N.
- **J-N** - Identifies intermediary J-Node. When a message is created by first time I-N will be the same as D-N.
- **Size** - Size of the application's message being passed (in Bytes).
- **HOPS** - Total hops the ant has performed.
- **TTL** - (remaining) TTL value.
- **Timestamp** - Time stamp with the message's time of creation.
- **S-N** - Source node.
- **D-N** - Destination Node.
- **Path** - A list with the intermediary nodes the FA or BA has already visited.
- **Priority** - Priority of the message, possible values are: Expedite (BA) and Normal (FA)
- **(q|Q)uality** .- Sum of path's quality (FA) or Total Quality Inverse Path (BA)

MMT allows updating the plugin to suit different transport technologies, in other words, on a real GrAnt implementation, it would be easy to update the plugin to recognize new communication mediums, e.g., Bluetooth or 802.11n protocols. These changes do not affect the elements designed and MMT will continue offering the same fields.

Regarding the fields we use in GrAnt, we can observe quite simply the paradigms differences between IP and DTN. In protocols such as GrAnt, elements for making routing decisions are appended to the message, whereas in IP, the message format is agnostic of the routing protocol. To avoid confusion, packets passed from one node to another will be called ants, where a Forward Ant (FA) carry the message generated by an application, and a Backward Ant (BA) serves as a message reception acknowledgment.

The ONE traces are processed by the MMT plugin and then analysed following two approaches: *security rules* to be validated and *attack behaviors* to be detected. We put special emphasis on the former security rules, which are useful for checking the ants' fields and thus, it is possible to detect anomalies in the topology.

Security rules define expected behavior of functionality and/or security concerns of the element that is being monitored, while the attack describes malicious behavior that can be an attack model, a vulnerability or a description of node misbehavior.

MMT rules are defined in an XML format using properties composed of a context and a trigger. The context describes the state of the system at a given time, and the trigger is the event or set of events that constitute the attack or misbehavior that occurs when the context holds. Depending on whether the property is a security rule or an attack we interpret it as described bellow:

- If the context is true, and the trigger is detected, then the security rule has been satisfied and everything is ok. Here we state that the security rule has been respected.
- If the context is true, but the trigger is not detected for a certain time period then the rule has been violated. Here we have a misbehavior, a vulnerability or an attack since the rule was not followed as required.

As mentioned in the Section I, an underground mine is an ideal environment for a DTN. Given the strong attenuation that mine walls impose to radio signals, the workers movements cause intermittent communications between their devices that follows closely the workers' scheduling. Almost all the workers, or even mobiles objects such as carts, will part from the central point to different parts of the mine, and some of them, will be moving from tunnel to tunnel or returning periodically to the central point. Using MMT security rules it is possible to characterize the behaviour of GrAnt on a given topology and pattern of movements, detecting weaknesses in the communications potential of the workers' scheduling by inspecting the ants' fields.

B. Rules to Trace

In general, the information flow ends at the exits of mines, such as information captured by air quality sensors, conditions of machines and parts, etc. This information is transported by carriers that could be mine workers or other moving elements in the mine and even could be the sensors themselves, that are able to deliver messages to a central point for decision-making. For any real world application using DTN protocols, the nodes flow will be the key for how messages are forwarded. Usually, we have nodes collecting information to be sent to headquarters or central points. If there is no nodes moving between those sensors and central points the messages will never be delivered on time. If nodes do not have a proper movement schedule messages can be lost.

Who makes decisions, what they will be and where they will be sent depends on the application. Nevertheless, what is needed is general purpose analysis means for DTNs, in particular, tools to confirm that the nodes are moving in with the expected pattern and, if not, identify the anomaly.

We define three MMT rules with the objective of detecting the anomalies, the first rule aims to detect when the distance between the central point and the nodes is too far, the second is used for analyzing the social metrics of the nodes and the last is focused in detecting abnormal movements.

1) Critical distance: It is necessary to assure that the movement of all nodes successfully allowed the message to be transmitted from the tunnels to the central point. Otherwise, there is a risk that the ants will expire before reaching their destination.

To do so, we designed a security rule, named *critical distance*, that detects when a BA reaches destination with a TTL value below a certain threshold. If this occurs, we can assume that other messages are being lost.

2) Confirm scheduling planning: For any application to have success transmitting their messages from source to destiny on DTN environments, one or more nodes need to have a very good social parameters. For real world environment where those nodes are people with a clearly scheduling this should be enough to guarantee a high social degree. However, this is difficult to confirm.

MMT is used to follow the meetings between nodes to confirm that their scheduling is truly helping the communication flow.

Defining properties composed of contexts and triggers are not sufficient to obtain this information; however, MMT allows more than this, since when these rules are detected, we can perform additional actions that could be, for instance, generating a report for each node indicating the number of encounters with other nodes.

For this property, MMT generates one additional report on the social metrics (i.e., Centrality and Betweenness) for each node.

3) Tracking unexpected movements: While the Schedule planning rule is intended for determining the centrality of a particular node which is supported by its position within the mine, here what we want is to detect unusual patterns.

Such patterns could be due to nodes that normally should not possess a centrality greater than a certain threshold, but do. In this case, the rule is executed for all nodes and their centrality is verified, at the same time as the previous rule, but only the nodes not in the group defined by the rule schedule planning are listed.

V. Conclusions

In this paper we have provided performance-based and functional analysis of the GrAnt routing protocol for DTNs. The analysis has been performed using two tools, the ONE network simulator and the monitoring tool MMT. In order to illustrate the application of the GrAnt protocol, we consider an underground mines scenario that presents all the characteristics of DTNs. Using MMT we have performed several experiments to analyse the behaviour of GrAnt protocol on a given topology and pattern of movements, detecting some weaknesses in the communications potential of the mine workers' scheduling. The experiments results show limitations that should be considered during the development of real DTN applications. We have also developed a general methodology based on MMT, to correlate the performance of a DTN protocol (such as GrAnt) with the information stored in its messages. For future work we are planning the development of attacks rules based on MMT, in order to test the robustness of the GrAnt protocol.

References

[1] NASA, "NASA Successfully Tests First Deep Space Internet," NASA, 18 November 2008. [On line . Available: http://www.nasa.gov/home/hqnews/2008/nov/HQ_08-298_Deep_space_internet.html.

[2] NASA, "Disruption Tolerant Networking for Space Operations (DTN)," 03 March 2013. [On line]. Available: http://www.nasa.gov/mission_pages/station/research/experiments/730.html.

[3] Google/Jet Propulsion Laboratory; NASA/Jet Propulsion Laboratory; The MITRE Corporation; Intel Corporation; SPARTA, Inc., "RFC 4838: Delay-Tolerant Networking Architecture," IETF Trust, 2007.

[4] M. Dorigo y G. d. Caro, Ant Colony Optimization Meta-Heuristic, New Ideas in Optimization, McGraw-Hill, 1999.

[5] M. Dorigo, V. Maniezzo y A. Colorni, "The ant system: Optimization by," *IEEE Transactions on Systems, Man,,* nÂº 26, pp. 29-41, 1996.

[6] K. V. Ana Cristina, M. Anelise, R. D. Myriam y C. V. Aline, "GrAnt: Inferring best forwarders from complex networks' dynamics," nÂº 57, pp. 997-1015, 2012.

[7] M. Dorigo, M. Birattari y T. Stützle, "Ant Colony Optimization, ARtificial Ants as a Computational Intelligence Technique," *IEEE Computational Intelligence Magazine,* November 2006.

[8] M. Dorigo, V. Maniezzo y A. Colomi, "Positive feedback as a search strategy," *Dipartimento di Elettronica, ,* pp. 91-016, 1991.

[9] A. Keränen, J. Ott y T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," ICST, Rome, 2009.

[10] W. Mallouli, B. Wehbi, E. Montes de Oca y M. BourdellÃ¨s, "Online Network Traffic Security Inspection," de *9th workshop on system testing and validation (STV)*, Paris, France, 2012.