

Robust data Replication Algorithm for MANETs with Obstacles and Node Failures

Andrea Detti[†], Lorenzo Bracciale[†], Francesco Fedi^{*}

[†]Electronic Engineering Dept., University of Rome “Tor Vergata”, Italy, {andrea.detti,lorenzo.bracciale}@uniroma2.it

^{*}Space Software Italia (Elsag Datamat), Rome, Italy, fedi@ssi.it

Abstract—In this paper we present a **Robust data Replication Algorithm (RRA)** for MANETs. RRA ensures persistent data availability by duplicating data over replica-nodes, while minimizing the consumption of radio resources. RRA is the first algorithm thoroughly designed to operate in environments with node failures and radio obstacles (e.g., buildings, walls). Our challenge is to consider such impairments as “accidental”; i.e., their occurrence neither can be run-time predicted, nor can be a-priori quantified in terms of probability. As instance, this is the case of nodes wandering in an unknown territory. To cope efficiently with the novel challenge, we suggest a practical assumption: during a refresh period, only a single accidental impairment may occur. Following this assumption, we succeed in formalizing (and solving) the resource minimization problem as the finding of a minimum-cost-set-cover. We assess RRA performance through NS2 simulations. Simulation code is released as open-source.

Index Terms—Ad-hoc network, replication scheme, data availability, obstacles, nodes failure

I. INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a network composed by nodes that are free to move. A node can communicate with a neighbor node over a wireless link; otherwise, it can communicate with a far node by using a set of intermediate nodes as relay. The mobility makes the network topology dynamic, and routing protocols (e.g. OLSR [1]) are properly designed to operate in such environments. Nevertheless, the breakage of some critical links, as well as the failure of some critical nodes, may split the MANET in to several disjointed partitions. Nodes can not access to data stored outside their own partition.

A data can be mission critical or not. While it could be tolerable that a not critical-data is discontinuously available, the availability of a critical-data has to be *persistent*, otherwise mission could fails. Furthermore, if a critical-data changes during the time, the data query should return up to date information. Data replication techniques improve the probability of accessing to critical-data by duplicating it over a set of replica-nodes. The set of replica-nodes should guarantee data accessibility versus forthcoming network partitioning. In general, there are several sets of possible replica-nodes offering the same level of data accessibility; therefore, it is worth to carry out an optimization. As

instance, an optimization goal could be the minimization of wireless resource consumed for replication purposes, as well as the best exploitation of storage capability [6].

Data replication is carried out by a protocol that periodically or reactively retrieves the required information (e.g., current topology), selects which are the replica-nodes, accomplishes the critical-data distribution and removes obsolete version of critical-data. Data replication protocol has to be assisted (or could interoperate) by a data discovery protocol, which properly handles data query.

In this paper we present Robust data Replication Algorithm (RRA), a replication algorithm devised to minimize the wireless resource usage in case of dynamic critical-data. The impairments faced by RRA are:

- (imp1) link breakages due to the excessive increase of link length;
- (imp2) accidental link breakages due to movement of nodes behind unknown obstacles;
- (imp3) accidental node failure due to unexpected hardware damage.

The facing of impairment 1 is a classical issue [4][5][7][8], and a lot of literature solutions strive to predict the link breakage by monitoring network status. On the contrary, to the best of authors’ knowledge, RRA is the first replication algorithm designed to cope with the accidental impairments 2 and 3. The “accidental” adjective means that their occurrence i) neither can be predicted, ii) nor can be a-priori evaluated in terms of probability [3].

We have been motivated in considering accidental impairments, seeing that MANET paradigm is perfectly tailored for scouting missions, accomplished by persons or robots operating in an “unknown” territory.

Currently, RRA does not take into account storage constraints. Moreover, RRA is just a distributed “algorithm” and it is not a protocol; this means that we do not address here any protocol related issues, like control messages structure, state-machine architecture, possible cross-layering and so on.

II. RELATED WORKS

Replication approaches coping with link breakages due to the increase of link length (i.e., impairment 1) have been previously addressed by Wang et al. [4] and by Derhab et al. [5]. Such approaches identify imminent MANET partitions by predicting link breakages. The prediction is carried out in

This work has been funded by the Italian Department of Defense, with the grant P.N.R. N° 04/07, “SAMAS”, a project addressing cooperative Multirobot systems specifically suited for Humanitarian Demining.

different ways and relies upon different information: GPS or node distance measurements in [4], topology structure in [5]. What widely distinguishes such a works from this one is they are focused on “service” replication, rather than “data” replication. The optimization carried out in [4][5] is meant to limit the number of replica-servers; on the contrary, RRA optimization is aimed to minimizing the wireless resource usage and we will show that these two optimization goals are strongly contrasting. Furthermore, critical-data changes over time, unlike service does not. So, service-replication has not the burden of limiting information inconsistency.

Valuable works regarding replication of dynamic critical-data have been accomplished by Hara et al. [6]. Nevertheless, Hara’s works are more focused on storage constraints than on minimization of wireless resource consumption. Finally, for a comprehensive and up-to-date survey on the possible replication approaches we remand the reader to [7] and [8].

III. ROBUST DATA REPLICATION ALGORITHM

A. Overview

RRA ensures persistent accessibility to a critical-data in a MANET by identifying possible partitions before they occur and providing them with at least one replica-node. In doing so, data accessibility is ensured, whatever impairments will occurs. Data replication is accomplished while minimizing the network resource usage.

We consider a MANET formed by a set of disjointed partitions. RRA is periodically executed by the *replication-manager*, which is a node of the partition that has the most updated version of the critical-data. RRA is feed with the following inputs:

- (input1) topology information about the wireless connectivity among nodes of the partition;
- (input2) measurements of either distances between neighbor nodes (DIST-mode) or their speed and position (GPS-mode);
- (input3) identifiers (e.g. IP addresses) of nodes of the partition that holds the critical-data with the related version identifier (e.g., a time-stamp);
- (input4) channel model information from which is possible to derive i) transmission radio coverage R and ii) distance information from power measurements, thus to avoid GPS equipments in DIST-mode.

At the start of a refresh period, RRA accomplishes the following tasks:

- (t1) discover all possible forthcoming partitions;
- (t2) select the cheapest set of replica-nodes that provides the presence of critical-data in each possible forthcoming partition. The cost of a set of replica-nodes is the amount of radio resource consumed for replication purpose;
- (t3) upload critical-data on replica-nodes that do not already have an updated version, and remove outdated versions of critical-data from other nodes.

In the next subsections, we better explain how RRA

carries out tasks t1 and t2. To support comprehension, nevertheless, it is necessary to firstly describe the methodological approach used by RRA to cope with accidental link breakages and node failures (see Section I).

B. How to cope with accidental impairments?

We observe that *a whole protection versus all possible accidental link breakages or node failures would require the flooding of critical-data on each node of the partition*. All links, indeed, could break simultaneously.

Nevertheless, the wireless resource usage of data flooding is likely unbearable in MANET environment. To overcome this impasse we decided to face a narrower set of accidental events. We design an algorithm that assures persistent data accessibility in cases of *single accidental impairment per refresh period*. If a single link accidentally breaks or a single node accidentally fails, our algorithm maintains data accessibility, whatever is the specific link or node failed. Conversely, if more than one links or nodes accidentally fail (e.g., due to radio channel correlation), our algorithm could not guarantee data accessibility. We show through simulations that this design approach represents a good trade-off between complexity, data persistence and bandwidth usage; moreover, it is as more effective as the refresh period is shorter.

C. Discovery of possible forthcoming partitions (t2)

In what follows, we briefly call “network” the MANET partition wherein RRA is running. The discovery of possible forthcoming partitions consists in three steps:

- (s1) representation of the network topology as a graph $G=\{U,V\}$, where vertices are the nodes of the network and edges are the direct wireless links between nodes;
- (s2) prediction of wireless links $\{l_i\}$ that could break before the next refresh period because of excessive length and subsequent enumeration of connected sub-graphs $\{G_i\}$ that come out from G by pruning links $\{l_i\}$.
- (s3) location of bridges and articulation points of each sub-graph G_i and subsequent enumeration of the sub-sub-graph $G_{i,j}$ coming out from G_i by pruning one at a time such bridges and articulation points.

The step s2 discovers the possible forthcoming partitions $\{G_i\}$ caused by impairment 1 (see Introduction). Because of either an accidental link breakage (impairment 2) or an accidental node failure (impairment 3), a sub-graph G_i could be further partitioned in sub-sub-graphs. Step s3 properly identifies the possible set of sub-sub-graphs $\{G_{i,j}\}$. We stress that we are coping with the occurrence of only one accidental impairment, as previously discussed.

To accomplish step s2, we may use GPS information (GPS-mode) or node distance measurements (DIST-mode). We infer the future link length as explained in next section III.C.a. To detect the connected sub-graphs $\{G_i\}$ we use a run of the Depth First Search algorithm (DFS). Within the same DFS run, we also locate graph bridges and articulation points, respectively defined in graph theory as: the links and nodes whose every “single” removal will partition the graph. Therefore, we obtain all the possible sub-sub-graph $\{G_{i,j}\}$ by

iteratively removing bridges and articulation points, one at a time.

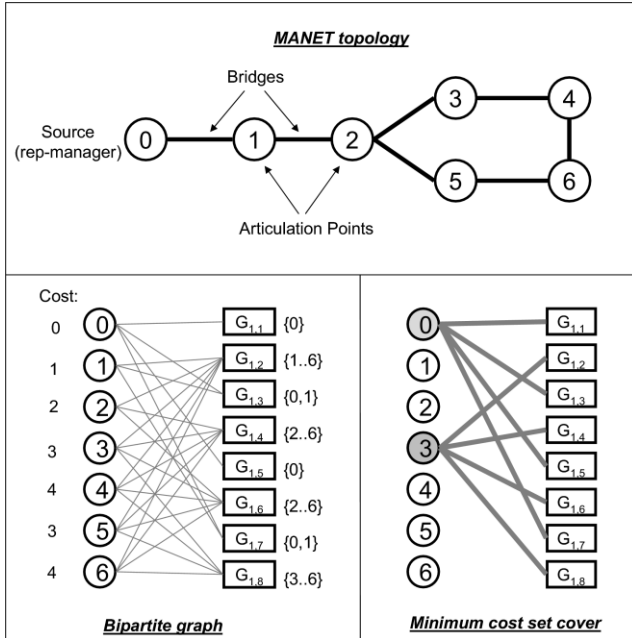


Figure 1 –Example of still MANET topology, related bipartite graph with cost and minimum cost set cover.

To support comprehension, we discuss the discovery of possible forthcoming partitions $\{G_{i,j}\}$ in case of the still MANET topology reported in Figure 1. We use a still MANET for easy of illustration. Since nodes do not move, step s2 returns a single sub-graph G_1 which topology is equal to the one of current MANET. Regarding step s3, the bridges of G_1 are the links 0-1 and 1-2. The articulation points are nodes 1 and 2. By removing the link 0-1, we split the network in two partitions: $G_{1,1}=\{0\}$ and $G_{1,2}=\{1,2,3,4,5,6\}$, where in the brackets we report the number of the composing nodes. By removing the bridge 1-2 we partition the network in $G_{1,3}=\{0,1\}$ and $G_{1,4}=\{2,3,4,5,6\}$. For what concerns articulation points, the fail of node 1 creates the partitions $G_{1,5}=\{0\}$ and $G_{1,6}=\{2,3,4,5,6\}$, while the fail of node 2 creates the partitions $G_{1,7}=\{0,1\}$ and $G_{1,8}=\{3,4,5,6\}$.

a) Link length prediction

The link length prediction of RRA assumes that during the refresh period nodes' movements are rectilinear. The adopted prediction formulas depend on the (input2) information. In GPS-mode the prediction formula is straightforward, since the position of nodes and their speed are known. Otherwise, in DIST-mode we only have distance measurements, so link length prediction is more complex and less accurate. In what follows we describe the RRA link length prediction algorithm, where the configuration parameters are: i) maximum node speed, $mspeed$, and ii) RRA refresh period, Tr .

Let us assume that at the start of k -th refresh period two nodes $N1$ and $N2$ have a wireless link which length is $d(k)$. We point out that since motion is assumed to be rectilinear, *the evolution of distance follows a parabolic behavior*. Thus, RRA predicts what will be the distance $pd(k+1)$ at the

end of refresh period $\#k+1$ as follow:

- (p1) if measurements $d(k-2)$, $d(k-1)$ and $d(k)$ are available, then $pd(k+1) = 3(d(k)-d(k-1))+d(k-2)$ (i.e., the formula of next value while moving on a parabola with a regular step);
- (p2) if only the measurements $d(k-1)$ and $d(k)$ are available, we assume that $pd(k+1)=d(k)+\min(d(k)+d(k-1), 2*mspeed*Tr)$. In this formula we are conservatively considering the worst case that nodes crossed between the two measurements instants. Consequently, the forthcoming link length would be $d(k)+ (d(k)+d(k-1))$. The minimum operator account that the maximum link length increase is $2*mspeed*Tr$ (i.e., nodes moving away in a opposite direction)
- (p3) if only the measurement $d(k)$ is available, we conservatively consider the maximum increase of link length; thus $pd(k+1)=d(k)+2*mspeed*Tr$

D. Selection of cheapest set of replica-nodes ($t3$)

So far, we have identified all the possible forthcoming partitions $\{G_{i,j}\}$. Now we need to find out the “cheaper” set of replica-nodes, in order that each $G_{i,j}$ contains at least a replica-node. The cost of a replica-node is the number of radio transmissions needed to bring critical-data to the replica-node. If a replica-node already has an updated version of the critical-data, then the cost is equal to 0. Otherwise, the cost is equal to the number of traversed wireless links between replication-manager and replica-node.

Let us build a bipartite graph B which left vertices are the network nodes (belonging to the partition of the replication-manager) and right vertices are the possible forthcoming partitions $\{G_{i,j}\}$. There is an edge between a left and a right vertex, if the node associated with the left vertex belongs (“covers”) to the partition associated with the right vertex.

As instance, in Figure 1 we report the bipartite graph relevant with the partitions discovery discussed in the section III.C.

The bipartite graph represents available choices and costs to provide each forthcoming partition with a replica-node. In the case of Figure 1, if node 3 was chosen as a replica-node, it would provides critical-data on the partitions $G_{1,2}$, $G_{1,4}$, $G_{1,6}$ and $G_{1,8}$. Moreover, the cost of choosing node 3 would be 3, since this node is 3 hops away from replication manage (node 0).

Exploiting the bipartite graph representation, we argue that the problem of providing all possible forthcoming partitions with at least a replica-node, while minimizing the consumption of radio resources, can be formalized as *the finding of the cheaper set of left vertices that “cover” all the right vertices*. That selection problem is equal to the well-know *minimum-cost-set-cover*¹ [9].

The minimum-cost-set-cover is proved to be NP-Hard, thus we resort to a classical greedy approximation [10] that iteratively works as follows: at each iteration, choose the left

¹ In terms of minimum-cost-set-cover formalization, a “element” is a partition $G_{i,j}$, a “set” is formed by all partitions $G_{i,j}$ covered by a node. Aim

vertex V with the lower ratio between i) the number of yet uncovered right vertices that V covers and ii) the cost of V . The iteration ends when the selected left vertices cover all right vertices. As instance, in Figure 1 we report the minimum cost set cover relevant with the bipartite graph previously described. In this case, replica-nodes selected by RRA are node 0 and 3, with an overall cost equal to 3. We observe that, node 5 and node 3 cover the same partitions and have the same cost; in this case, RRA has chosen randomly node 3.

We conclude that description of RRA by highlighting that the major computational complexity of RRA is the execution of a Depth First Search algorithm, i.e. $O(N^3)$ where N is the number of MANET nodes.

IV. PERFORMANCE EVALUATION

In this section, we analyze and discuss the results of a simulation campaigns aimed at assessing RRA performance.

A. Simulation details and performance metrics

We use Network Simulator 2 and publish the simulation code in [11], thus fellow researchers can verify and benchmark RRA.

We simulate a scenario (Figure 2) where nodes wander in a squared area, and their transmission range is set to 50 meters. Within the area, we dispose a grid of L-shape obstacles with a side length of 25 meters; grid rows and columns are spaced of 50 meters; between consecutive rows there is a horizontal offset of 25 meters. When the line-of-sight between a couple of nodes intersects an obstacle, nodes can not directly communicate [2].

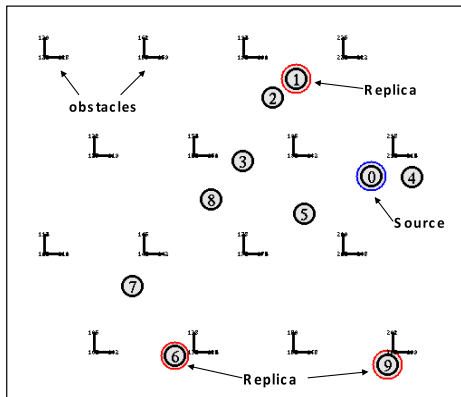


Figure 2 –Snapshot of a simulation scenario

We consider a Random Waypoint Mobility model, with constant speed and constant pause time of 5 sec. To account obstacle presence, we slightly modify the random way point logic in what follows: after a pause period, a node randomly selects a destination, the selected destination is valid whenever the linear movement toward it does not requires the crossing of any obstacle; otherwise another destination is randomly selected. To visualize the simulation scenario, as well as the effect of RRA, a network animation based on NS2 NAM tool can be retrieved at [11].

The source of critical-data is placed on node 0, the data changes every 30 seconds and at every change the version identifier of the data is incremented by one unit.

We test RRA operating in GPS-mode, i.e. exploiting GPS positioning measurements, and RRA operating in DIST-mode, i.e. exploiting only node distance measurements. The refresh period of replication algorithm is 5 sec. We compare RRA performance with the case of absence of replication and with a trivial flooding technique, which has the following logic: every refresh period the replication-manager pushes the critical-data to all the other nodes that have an outdated version². Performances achieved in absence of replication and through flooding are useful reference “bounds”.

Each simulation runs for 60 minutes and we collect statistics after a warm-up period of 30 minutes. To simulate an harsh node failure situation, from the 45th minute and for at most 5 times, we sequentially remove one replica-node every 6 seconds.

Since our goal is limited to test the algorithm, during simulation we do not exchange any message and ideally provides each node with all the inputs needed to accomplish the replication process.

During the simulation, we gather three kind of performance parameters:

- *Data-availability*: probability that a generic node is able to retrieve the critical-data either from a replica-node or from the source.
- *Update-cost*: on average, how many times per second the critical-data is transferred on the wireless interface for replication purposes.
- *Replica-age*: average age of replicas in the system. The age of a replica is the difference between the version identifier of the replica and the actual one of the source.

We plan a set of simulations aimed at showing: i) the need of an algorithm expressly designed for scenarios with obstacles and node failure; ii) the difference between the minimization of the number of replicas [4][5] and the minimization of the wireless resource usage; iii) RRA performance versus node speed, area size and number of nodes.

B. Effects of obstacles and node failures

In this section we briefly compare RRA with an its “limited” version. Limited-RRA has link breakage prediction mechanism but does not implement the protection toward accidental impairments (i.e., step s3 of Section III.C is skipped). We deem Limited-RRA to be representative of literature approaches based on link breakages prediction, carried out by monitoring the network status.

We consider a 250x250 m² area with 20 nodes at 1 m/s. Although not reported, we tested limited-RRA in a plain area

² We strived to find a literature benchmark for RRA and we initially decided to implement in our simulator the ServCov algorithm described in [4] (and simulator in [11] actually has ServCov). Nevertheless, after some simulations, we have considered unfair to use ServCov [4] in data-replication scenarios since ServCov is devised for service-replication. For the same reason we have avoided to implement other literature algorithms (e.g., [5][6]) with different goals from RRA.

of the problem is to cover all the elements by using a minimum cost collection of sets.

(i.e., no obstacles and no node failures) and the algorithm provided persistent data availability. In a plain area, links break because nodes get far away from each other and the monitoring of network status is enough to predict and cope these impairments.

When we introduce only obstacles (Figure 3a) or obstacles and nodes failure (Figure 3b), data availability temporary goes down, especially when node failure occurs at time 45 min; limited-RRA, indeed, only copes with predictable impairments. On the contrary, the complete version of RRA (Figure 3c) practically succeeds in every time. Only in three very limited time intervals (32, 37 and 42 sec) the retrieval probability is less than one. Indeed, in these instants more than one accidental link breakage occurs.

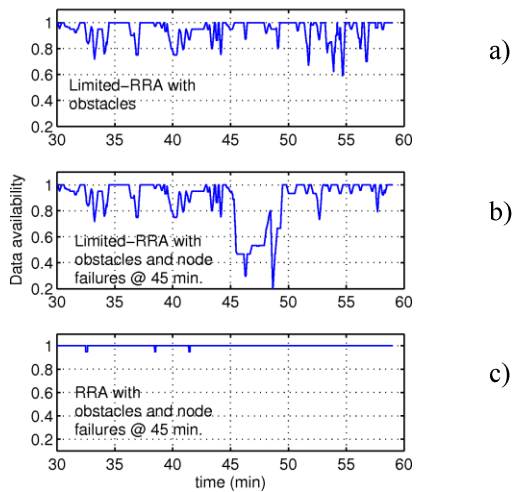


Figure 3 – Data availability versus time for limited-RRA with obstacles (a), limited-RRA with obstacles and node failures (b), RRA with obstacles and node failures (c).

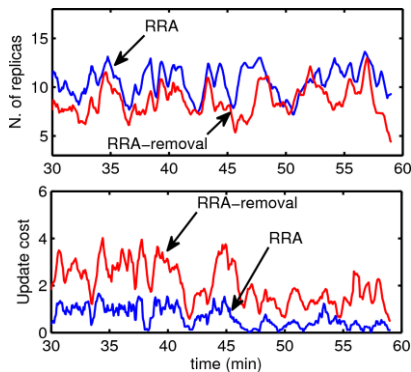


Figure 4 - Number of replicas and update-cost.

C. Number of replicas vs wireless resource usage

In this section we briefly show that minimizing the number of replicas does not lead to the minimization of radio resource consumption.

Figure 4 reports the behavior of number of replicas of the RRA against a modified version, named RRA-removal. In the modified version, every refresh period we remove all replicas except the one held by the replication-manager, then we proceed as in case of RRA. The considered scenario is a 250x250 area with 20 nodes at 1 m/s. We observe that RRA-

removal has lesser replica-nodes; nevertheless the update-cost is higher than RRA. Keeping already allocated replicas (even if their number was actually excessive for data availability purpose) does not consume wireless resources and may avoid future allocation of new replica-nodes.

D. RRA performance

Figure 5 reports the performances of RRA operating in GPS-mode, RRA operating in DIST-mode, flooding (flood) and the case of absence of replication (no-rep). Performances are analyzed versus node speed. The simulation scenario is an area of 250x250 m² with 30 nodes. Here, as well as in all the next simulations, we consider the presence of obstacles and node failure.

The data availability obtained without replication is almost 0.6. Such a constant behavior occurs since data access probability is directly related to node density, which does not change in this simulation set.

RRA-GPS, RRA-DIST and flooding achieve persistent data availability (i.e., equals to 1), as in all the next cases we are going to present. So, we do not further discuss this result any more.

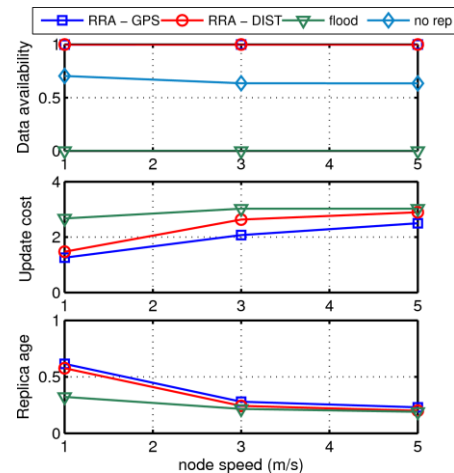


Figure 5 – Performance varying node speed.

Flooding provides the best performance in terms of replica age, but the worst one in terms update-cost.

Regarding RRA, we observe that the increasing in speed lowers replica age, because nodes meet themselves more frequently and updated versions of critical-data are spread more quickly.

In turn, the faster the distribution, the higher the update-cost. Moreover, we observe that RRA update-cost tends to the flooding one. We can motivate this asymptotic behavior as follows: the increasing in node speed results in a more occurrence of link breakages at the end of the refresh period. Consequently, the possible forthcoming partitions (derived by step s1 in Section III.C) tend to be formed by a single node and RRA tends to place a replica for node, like flooding does.

RRA-GPS has a lower update-cost than RRA-DIST, while other performances are practically equal. Moreover, the difference between update-costs increases as node move faster, up to the flooding upper bound. This is due because the more the node speed, the more the times RRA-DIST

operates without three distance measurements. That lack of distance measurements leads the prediction algorithm (see III.C.a) to place more replicas. In all the next simulations, we use a node speed of 1 m/s that vanishes the performance improvement given by GPS equipments. Thus, we will not comment anymore that RRA-GPS has same performance of RRA-DIST.

Figure 6 shows algorithm performance varying the simulated area, in case of 30 nodes at 1 m/s. At the increasing in area size, node density decreases. Therefore, the data availability in case of absence of replication decreases as well.

As far as the RRA update-cost is concerned, we argue two contrasting effects: on the one hand, the decrease in node density requires more replica-nodes and each replica-node costs more, since network paths are longer. Consequently, update-cost tends to increase. On the other hand, the decrease in node density leads replica-nodes to interact more rarely, so lower update processes occur and update-cost tends to decrease (as well as replica age increases). Such contrasting effects motivate the parabolic behavior of RRA update cost.

Regarding the flooding performance, we have an initial increase of update-cost, since networks paths become longer. This effect is then contrasted by a more rarely interaction among nodes.

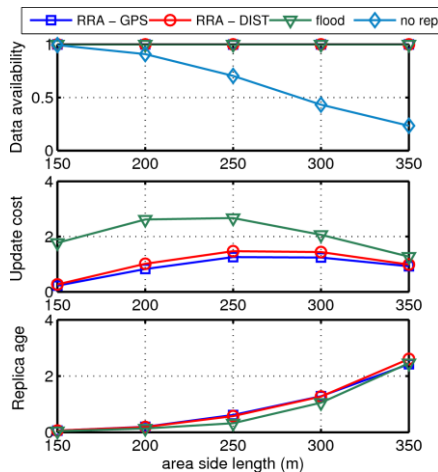


Figure 6 – Performance varying the simulation area.

Figure 7 reports performances by varying the number of nodes. The simulation area is $250 \times 250 \text{ m}^2$ and node speed is 1 m/s. At the increase of number of nodes, node density increases and the performance behavior is the same of the previous scenario, when we shrink the area size. The only considerable difference from previous case is that flooding needs to distribute replicas on more nodes, thus its update-cost linearly increases. On the contrary, the resource consumption of RRA scales with number of nodes.

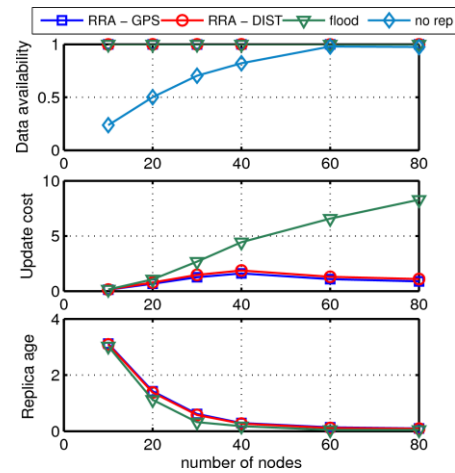


Figure 7 - Performance varying the number of nodes

CONCLUSIONS

We proposed and explored a new challenge in data replication research area: the presence of accidental impairments. To assure data persistency in such environments, we resort to a practical simplification of the problem: just to consider at most a single accidental impairment per refresh period. In this way, we succeed in formalizing the replica placement problem as a problem of minimum-cost-set-cover. RRA is an algorithm that implements our findings. Simulation results show that RRA provides a good trade-off between data persistence performance, bandwidth usage and complexity.

REFERENCES

- [1] P. Jacquet T. Clausen, "Optimized Link State Routing Protocol (OLSR)", *RFC 3626*, Oct 2003.
- [2] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri. "Real world Environment Models for Mobile Ad hoc Networks" *IEEE Journal on Selected Areas in Communications*, March 2005, Volume 23, No. 3
- [3] Ball "Computational Complexity of Network Reliability Analysis: An Overview", *IEEE Transaction on Reliability Vol. R-35 No. 3*, Aug 1986
- [4] K. Wang and B. Li, "Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks", *In Proc. IEEE Infocom 2002*
- [5] A. Derhab and N. Badache, "A pull-based service replication protocol in mobile ad hoc networks" *European Transactions on Telecommunications*, 18(1):1-11 2007
- [6] T. Hara, "Data Replication for Improving Data Accessibility in Ad Hoc Accessibility in Ad Hoc Networks", *IEEE Transaction on Mobile Computing Vol.5 N.11*, Nov. 2006
- [7] P. Padmanabhan, L. Gruenwald, A. Vallur, M. Atiquzzaman" A survey of data replication techniques for mobile ad hoc network databases", *The VLDB Journal, Volume 17, Issue 5 P. 1143 – 1164*, Aug. 2008
- [8] A. Derhab and N. Badache, "Data Replication Protocols for Mobile Ad-Hoc Networks: A Survey and Taxonomy", *IEEE Communications Surveys & Tutorial, Vol.11 No. 2*, Second Quarter 2009
- [9] A. S. Asratian, T. M. J. Denley, R. Häggkvist, "Bipartite Graphs and Their Applications" *Cambridge Tracts in Mathematics*
- [10] V.V. Vazirani, "Approximation Algorithms" *Springer*
- [11] A. Detti, L. Bracciale, "RRA simulation code and network animations" available at http://netgroup.uniroma2.it/Andrea_Detti/RRA