Effect of Buffer Size on Anycast Routing in Delay Tolerant Networks

Éderson Rosa da Silva & Paulo Roberto Guardieiro

***Abstract*—DTN (Delay/Disruption Tolerant Networking) have grown as a research area that is focused on addressing the communication requirements to challenged networks, characterized by frequent disconnections and high delays. There are different types of DTNs, depending on the nature of the network environment. Hence, different routing schemes are proposed. In this paper we treat the anycast routing where hosts wish to delivery messages to at least one, and preferably only one, of the members in an anycast destination group. To do this, we analyze two routing approaches: a routing algorithm that takes in account only the number of hops, and other based on GAs (Genetic Algorithms) for route decision. Then we implement a GA-based anycast routing algorithm and analyze the effect of buffer size on performance. Our simulation results have shown that the routing using GA produces good results in the simulated scenarios.**

***Index Terms*—anycast routing, delay tolerant networking, genetic algorithms.**

I. INTRODUCTION

Along with the increasing penetration of computing and communications technologies into our world and our lives, we have seen the arising of emerging networks, however, in certain networking scenarios, important Internet protocols are not usable. DTN (Delay/Disruption Tolerant Networking) as a research area is focused on addressing the communication requirements specific to these challenged networks. These networks may suffer frequent disconnection, high delay, high data rates, asymmetric data rates between source and destination, with the possibility of never having end-to-end connectivity between the source and the destination over a given period of time. Therefore, the design of protocols for those networks becomes a unique challenge.

The IRTF (Internet Research Task Force) created a new research group to examine the more general area of DTN. That group is called the DTNRG (DTN Research Group), and it is currently the main open venue for work on the DTN architecture and protocols. The DTNRG is documenting these protocols as so-called experimental RFCs (Request for Comments).

RFC 4838 [1] describes an architecture for DTNs, defining an end-to-end message-oriented overlay, called bundle layer (Figure 1). RFC 5050 [2] describes the end-to-end bundle protocol, block formats, and abstract service description for the exchange of messages (bundles) in DTNs.

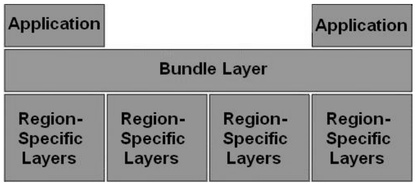

Fig. 1. Bundle layer.

Many of the principles of DTN architecture are reviewed by Fall and Farrell [3], being highlighted design decisions that have persevered through repeated analyses.

A common challenge in DTNs is the routing. In store-carry-forward operation, a next hop may not be immediately available for the current node for forward the data (bundle). The node will need to buffer the data, and maybe carry it, until the node gets an opportunity to forward the data and must be capable of buffering the data for a considerable duration.

Basically, we have two categories of routing protocols in DTNs: deterministic and stochastic. Zhang [4] reviewed a few routing protocols for both cases. Deterministic protocols use information about network topology or network conditions to make forwarding decisions. Stochastic protocols deal with the way that several copies of the message can be disseminated in the network to increase the chance that it would reach the destination. This way, many different routing schemes have been proposed, depending on the environments in which the node may find itself and the information used by the routing algorithms.

In our work, we use a multi-graph to represent the DTN, i.e. we consider that the network topology may be known ahead of time. Moreover, we treat the routing for anycast delivery, which allows a node to send a message to at least one, and preferably only one, of the members in a group. The idea behind anycast is that a client wants to send messages to any one of the several possible servers offering a particular service, but does not really care any specific one.

There are various applications of anycast in DTN such as disaster rescue field (people may want to find a doctor or fireman without knowing their locations and specific IDs), battle fields (e.g. a command center may want to deliver a particular message to any soldier among a group - squad), long

---

E. R. Silva (ederson@eel.ufu.br) and P. R. Guardieiro (prguardieiro@ufu.br) are with Faculty of Electrical Engineering, Federal University of Uberlandia, Uberlandia – MG, Brazil.

distance education (e.g. send a message to any one of the members in a group), and many other applications.

DTNs are characterized by long transfer delays. Under these situations, the group membership may change during a message transfer, being necessary to define the intended receivers of a message. In Section III-A, we will see a situation when from the perspective of traditional anycasting, it is not clear which nodes should receive a message and three anycast semantic models.

The most of works found in the literature treat the unicast delivery. However, in this case, the destination is unique and determined when the message is generated, while in anycast, the destination can be any one of a node group and the group membership may change during a message transfer.

The anycast routing algorithm makes route and destination decisions. We can have several anycast sessions in the network. This way, the routing algorithm decisions will influence many parameters simultaneously, like delay, delivery probability, and messages distribution.

In this paper we analyze two routing approaches: a very simple that considers only the application of the Dijkstra algorithm to compute the path with the lesser number of hops, and at the other extreme, a routing algorithm that uses GAs (Genetic Algorithms) to perform the anycast routing in DTNs.

We study the effect of buffer size (storage capacity available for the nodes) in the network on the routing algorithm decisions, i.e. in which situations the use of a complex routing algorithm (GA-based routing algorithm) is appropriate. To do this, we use simulation and compare the algorithms performance under different scenarios in DTNs modeled by graphs.

The remainder of this paper is organized as follows: Section II presents related works. Section III describes the system model. Section IV shows two anycast routing algorithms, a simple that considers only the number of hops and other that uses GAs. Section V describes the simulation while Section VI shows and discusses the results. The last Section presents the conclusion and future works.

II. RELATED WORKS

Anycast routing has been studied extensively in Internet and MANETs (Mobile Ad Hoc Networks). More specifically, anycast routing using GA in MANETs has been studied [5, 6], but routing in DTN is more challenging due to the frequent partitions and long end-to-end delay.

Multicasting is analyzed in DTNs [7, 8] using several multicast routing schemes. It is important to note that when the multicast service is used, mobile nodes responsible for assisting in the delivery of messages, store the messages until it is confirmed that all members of the destination group already have received. In our anycast case, mobile nodes responsible for bringing the message to a member of the anycast group need to store them until delivery to only one member of the anycast group, which leads to a substantial saving in storage of mobile devices that relayed the messages to a destination group.

Gong et al [9] analyzed the anycast semantic for DTN and presented a metric named EMDDA (Expected Multi-Destination Delay for Anycast). The authors assumed that nodes in the network were stationary. The connectivity among the nodes was the mobile devices that act as carrier to deliver messages for the nodes. Also the moving patterns of these mobile devices can be obtained.

We used some ideas from [9], but this work is based on estimation, i.e. decision about forwarding packets are based on the likelihood of the delivery of each neighbor. The decisions considered the average end-to-end delay. The routing performed by Gong et al [9] is categorized as stochastic case and our routing can be considered deterministic case.

Though Gong et al [9] presented three types of anycast semantics that allow the source explicitly specify the destination of a message through the CM (Current Membership), TIM (Time Interval Membership) and TPM (Temporal Point Membership) models, the network traffic during the selection of routing is not considered. Our GA-based anycast routing scheme incorporates both node storage constraint and network traffic dynamics.

GA-based approaches have been used to address the problem of SP (Shortest Path) routing with different chromosome representations: in [10] is used chromosome with constant length and in [11] is used chromosome with variable length. In our work we use chromosomes with constant length. The problem of multicast using GA is addressed in [12, 13, 14]. The main differences in these works are in the different chromosome representations, routing objectives, problems based on constraints, characteristic of the networks and methods to improve the algorithm convergence.

Although we use some ideas of the works above, we are interested in the problem of anycast routing in DTNs. The initial idea of using GA for routing in DTNs was introduced in [15]. In this paper, we analyze the effect of buffer size on GA-based anycast routing algorithm using different network topologies and the results are computed based on the average over 10 runs.

III. SYSTEM MODEL

For semantics of anycasting in traditional networks such as the Internet and MANETs, the receiver of an anycast packet is well defined, since data transfer delay in these networks is short. This, however, due to the large transfer delays is no longer valid in DTNs, because the memberships can change during data transfer. This way it is necessary to define new semantic models for anycast in DTNs.

We represent the DTN as a directed multi-graph. In a tutorial paper [16], it is described a simple combinatorial reference model that captures most characteristics of time-varying networks.

A. Anycast semantics

In the following, we consider the simple example in Figure 2, where a source sends a message to a group at time *t*. Let *t’* be the earliest time that other nodes could possibly receive this message according to network topology limitations. Suppose that node A joins the group at time $t_1 < t$ and leaves at time t_2, $t < t_2 < t'$. Node B joins at time t_3, $t < t_3 < t'$ and never leaves. From the perspective of traditional anycasting, it is not clear which nodes should receive this message, whether A, B or neither of them.

Fig. 2. Example of anycast semantic in DTNs.

Consequently, new anycast semantic models should be developed. In [9] is described that the intended receiver should be clearly defined for a message as group membership changes when nodes join and leave the group. It is showed three anycast semantic models that allow message sender to explicitly specify the intended receivers of a message:

– CM (Current Membership) model: the receiver of the message should be destination group member at the time of message delivery;

– TIM (Temporal Interval Membership) model: a message includes a temporal interval that specifies the period during which the intended receiver must be a member of destination group member;

– TPM (Temporal Point Membership) model: its intended receiver at least should be a member of destination group at some time during membership interval.

The models described above can be used according to the needs of the application. In our anycast routing we define the intended receiver when the message is generated. This is a particular case of the TIM model, whereas that the temporal interval is the instant of the message generation.

B. *Network model*

We use a directed multi-graph to represent the DTN and consider that the topology may be known ahead of time. In DTN graph more than one edge may exist between a pair of nodes. Besides, the link capacities (storage capacity, propagation delay and departure time) are time-dependent.

The edge representation used is showed in Figure 3. An edge between node 1 and node 2 means that there exist some mobile devices moving from the initial node 1 (source) to the terminal node 2 (destination). The storage capacity ($c(1,2)$) on all mobile devices is limited and we will study its influence in our results.

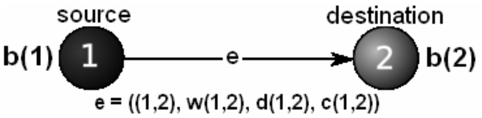

Fig. 3. Edge in a DTN graph.

We assume that every mobile device that moves between the same initial node, 1, and terminal node, 2, has the same moving speed, thus having the same moving delay, $d(1,2)$, from the node 1 to the node 2. The departure time $w(1,2)$ of mobile devices on each edge is represented by randomly generated numbers with Poisson distribution. $b(1)$ and $b(2)$ are the storage capacity of node 1 and node 2, respectively.

Besides, the nodes in the network are stationary and generate messages. On the other hand, mobile devices move from one node to another and do not generate messages themselves.

IV. ANYCAST ROUTING ALGORITHMS

In this section, we describe the routing algorithms studied for anycast delivery: the SP routing algorithm and the GA-based routing algorithm.

A. *Shortest path routing algorithm*

Routing algorithms in the constructed space-time graph can be developed using SP algorithm. Dijkstra’s algorithm [17] is a search algorithm that solves the SP problem for a graph. For a given source node in the graph, the algorithm finds the path with lowest cost between that node and every other node. This way, we applied the Dijkstra’s algorithm to search the path with the lesser number of hops between the source and the intended receivers. This algorithm does not consider any information about the network, and it takes in account only the number of hops.

B. *Role of genetic algorithm in routing*

The objective of GA in our work is to assist in the anycast routing for route and destination decisions. GAs are defined as search algorithms based on the mechanics of natural selection and natural genetics [18]. They combine survival of the fittest among setting structures (in our case, routes) with a structured yet randomized information exchange to form a search algorithm. They efficiently exploit historical information to speculate on new search points with expected improved performance.

Most of the real world problems are nonlinear, where nonlinearity is the norm, where changing one component may have ripple effects on the entire system, and where multiple changes that individually detrimental may lead to much greater improvements in fitness when combined. GAs are appropriated to solve these problems.

Before examining the mechanisms and the power of a simple GA, we must be clearer about our goals when we say we want to optimize a function or a process.

We defined each population individual being represented as a set of possible routes for each session. For each anycast session we use SP algorithm to obtain potential solutions in isolation, i.e. we consider many solutions for the same session to be combined by the GA-based algorithm. The GA keeps these routes created and allows filter routes to combine and produce offspring with new characteristics, which may replace low fitness old routes. Fitness function is a particular type of objective function that quantifies the optimality of a solution.

The GAs copy routes with some bias toward the best, mate and partially swap (sub) individuals, and mutate occasional possible paths for good measure. We set the crossover probability to 0.8 and the mutation probability to 0.03. We use these values to increase the diversity in the individual population. Our GA is controlled by the number of generations and the population has 50 individuals.

We want to search the route and consequently the anycast destination that better serve the routing objective (enhance some performance metric) using GA. Figure 4 shows an example of population representation. We have two individuals (X1 and X2) representing four anycast sessions and the points P1, P2, and P3 are representing the possible crossover points. The points P1, P2, and P3 are separating each possible route and they are used to produce only regular individuals. Each number in each square represents the nodes. We can see that the route for the first anycast session is 21-17-08, i.e. the source node is the node 21 and the destination chosen by the algorithm is the node 08 passing by node 17. Moreover, some squares are empty. This is because we are using chromosomes with constant length.

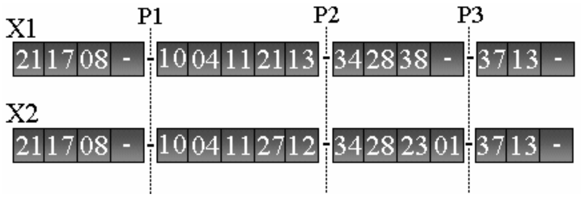

Fig. 4. Population representation.

Y1 and Y2 (Figure 5) represent the individuals generated by the crossover between X1 and X2 at point P2. Figure 6 shows a mutation of the second route in the individual X1.

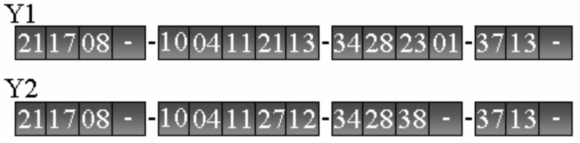

Fig. 5. Crossover example.

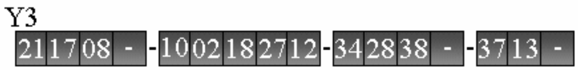

Fig. 6. Mutation example.

We define two performance metrics for optimization. First, we choose a DP_{min} (minimum Delivery Probability). We define that the GA must search routes with DP (Delivery Probability) above a threshold (DP_{min}) and a route (optimal or very good) that satisfying this probability and having the lesser delay.

V. MODELING AND SIMULATION

We use simulation to compare the performance of the GA-based anycast routing algorithm and the SP algorithm in different scenarios.

Depending on the available knowledge about the network, in [19] are defined four knowledge oracles. Our GA-based anycast routing algorithm has a partial knowledge (more practical assumption from an implementation perspective than complete or zero knowledge): it uses information about node and edge queuing, storage capacity, moving delay, and departure time of the mobile devices. We use these informations to define the better destination and compute the route. On the other hand, the SP algorithm considers only the number of hops.

In our simulation, we employ the Waxman Network Topology Generator [20] to generate a random graph of 40 nodes. In the Waxman generator, the nodes follow a Poisson process in the plane. The probability to have an edge between nodes u and v is given by

$$P(u,v) = \alpha . e^{-dist/(\beta . L)} \tag{1}$$

where $\alpha > 0$, $\beta \leq 1$, $dist$ is the distance from u to v, and L is the maximum distance between any two nodes. We set α to 0.4 (we chose a density of short edges relative to longer ones middle) and β to 0.25 (graphs with lower edge densities). These parameters can be adjusted to obtain the desired characteristics in the resulting graph. Figure 7 shows an example of network topology generated using the parameters above in the square area 1300 m x 1300 m.

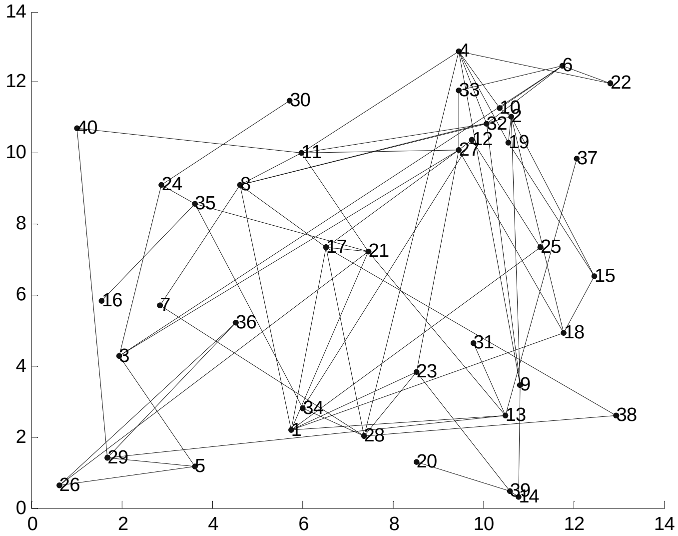

Fig. 7. Network topology.

We assume the communication between nodes is carried out by mobile devices (to simulate the behaviors of DTNs). For each edge generated by Waxman generator, we replace it with a mobile device acting as ferries (mobile nodes that exploit the device mobility to enable or improve the communication). We generate random numbers from the Poisson distribution to represent the leaving time ($w(u,v)$) of mobile devices on each edge with mean interval time selected randomly from 600 to 6000 seconds. The moving delay ($d(u,v)$) on each edge is a number selected randomly between 60 and 600 seconds, which is multiplied by the distance between the nodes.

We assume that the storage capacities (*c*(*u*,*v*)) of each mobile device are random numbers selected randomly between an interval, and we study its influence in routing algorithms performance. The storage capacities of each node (*b*(*node*)) may vary from 600 to 1000 messages. We showed an example of edge between node *u* (1) and *v* (2) in Figure 2.

For each anycast session (we consider only anycast traffic in our simulations) we generate a random number between 2 and 5 to represent the number of possible destinations. We randomly pick a node as the anycast source. The first destination member is selected randomly from the possible nodes except the source node. The rest of destination group is the node in sequence if this node is different from source node until completes the number of desired destinations. The messages to send to the destination group for each source can vary randomly between two numbers (we will vary this number of messages in Section VI-A). We generate random numbers between 0.04 and 0.06 to represent the inter-arrival times (messages generated per second). Table I has an example of initial traffic for four sessions.

TABLE I
INITIAL TRAFFIC EXAMPLE.

	SESSION 1	SESSION 2	SESSION 3	SESSION 4
Source	21	10	34	37
Destination group	[8; 9]	[12; 13]	[38; 39; 40; 1; 2]	[12; 13; 14]
Messages to send	380	380	370	300
Beginning of the session (s)	1700	2150	2700	4300

A message is split only at source and different parts (fragments) are routed along same paths. To compare the algorithms performance we collected statistics about *DP* (total number of unique anycast messages received by any anycast group member to the total number of messages transmitted by the anycast source) and delay *D* (weighted mean of delay, the weights are the number of delivered messages). We evaluate the SP algorithm and the GA-based anycast routing algorithm under different mobile devices storage capacities. Table II presents an example of the *DP*, delay *D*, and the routes found by the algorithms. We illustrated these routes achieved by GA-based and SP algorithms in Figure 4, and showed an example of crossover (Figure 5) and mutation (Figure 6).

TABLE II
PERFORMANCE METRIC EXAMPLE.

	DP (%)	*D* (s)	ROUTE SESSION 1	ROUTE SESSION 2	ROUTE SESSION 3	ROUTE SESSION 4
SP	77	11398	21-17-8	10-4-11-21-13	34-28-38	37-13
GA	91	10450	21-17-8	10-4-11-27-12	34-28-23-1	37-13

Analyzing Table II we see that the GA-based routing algorithm achieves a *DP* higher and delay *D* lesser than the SP algorithm. A reason for this behavior is that the route obtained by SP algorithm uses the same node (node 13) in sessions 2 and 4.

VI. PERFORMANCE EVALUATION

We simulate a network with 12 anycast sessions. The results are the average over 10 runs with different random seeds and network topologies. This way, we study the effect of buffer size on algorithms performance (SP and GA-based algorithms) under different network topologies generated by Waxman Network Topology Generator. To do this, we vary the number of messages sent by source nodes for each anycast session and the storage capacity of each mobile device. The GA-based algorithm is controlled by number of generations (200). Our simulations run for 43000 seconds (≈ 12 hours) in simulation time.

A. *Varying the number of messages sent by source nodes for each anycast session*

In this scenario we fixed the storage capacity of each mobile device (400 to 700 messages). We vary the number of messages sent by source nodes for each session to analyze the algorithms under different traffics. The results are the average over 10 runs with different random seeds and network topologies, varying this number of messages from 200 to 500, 300 to 500, and from 400 to 500. When we increase the number of messages and maintain the storage capacity of nodes and mobile devices constant, the routing becomes more challenging.

Figure 8 shows that for all scenarios both SP and GA-based algorithms achieve good rates of *DP*. It is considered a DP_{min} equal to 90%. It is important to remember that the GA-based routing algorithm searches routes above DP_{min} (90%). As the number of messages increases, i.e. the routing becomes more challenging, the *DP* decreases for both algorithms.

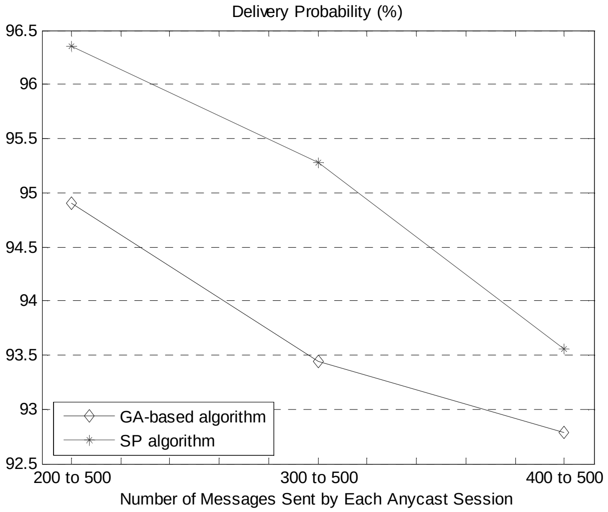

Fig. 8. *DP* under different number of messages sent by source nodes for each anycast session.

Analyzing the results for delay (Figure 9), we can see that the GA-based routing algorithm always obtains better results than the SP algorithm. We can see that the GA-based routing algorithm is working correctly, i.e. it finds routes with a *DP* above the DP_{min} (90%) and with the lesser delay. The reason for delay *D* increases is that when the number of messages sent by source nodes for each session increases, the competition and the delay waiting for an opportunity to transmit increases too.

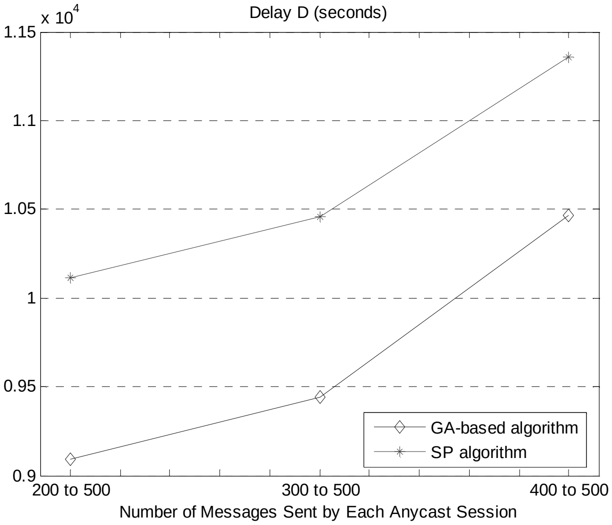

Fig. 9. Delay *D* under different number of messages sent by source nodes for each anycast session.

Table III presents the average number of hops used by each routing algorithm. As SP algorithm considers only the number of hops to route decision, it maintains this number constant. On the other hand, the GA-based algorithm uses more hops when the routing complexity increases. This is because it searches alternative routes to avoid that a large number of messages passing through the same edge, consequently the delay decreases.

TABLE III
AVERAGE NUMBER OF HOPS UNDER DIFFERENT NUMBER OF MESSAGES SENT BY SOURCE NODES FOR EACH ANYCAST SESSION.

Algorithm	Hops for messages = [200-500]	Hops for messages = [300-500]	Hops for messages = [400-500]
SP	23.4	23.4	23.4
GA	26.8	27.3	27.4

B. Varying the storage capacity of mobile devices

For performance evaluation, we test the algorithms for anycasting under different buffer sizes $c(u,v)$ on the mobile devices. In this scenario we fixed the number of messages sent by source nodes for each anycast session (400 to 500 messages) and we consider three values for $c(u,v)$: from 400 to 700, from 400 to 500 and from 300 to 500. Figure 10 shows the *DP* achieved by each algorithm. We can see that the *DP* for both algorithms decreases when the storage capacity of mobile devices is lower. This can be explained by the fact that when we decrease the storage capacity of mobile devices, the competition for an opportunity to transmit increases.

For $c(u,v)$ varying from 300 to 500, i.e. the scenario more challenging, the *DP* obtained by the SP algorithm decreases a lot. On the other hand, the results obtained by the GA algorithm present little variation if compared with SP algorithm, and it achieves *DP* above the DP_{min} (90%). This indicates that the GA-based algorithm is more robust when the network resources are lower.

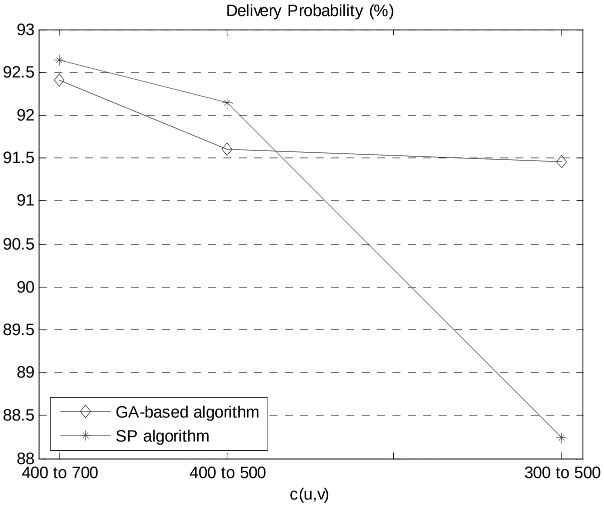

Fig. 10. *DP* under different storage capacity of mobile devices.

The results for delay *D* are showed in Figure 11. Again, when the routing becomes more challenging the delay *D* increases and the GA-based algorithm always gets better results than the SP algorithm.

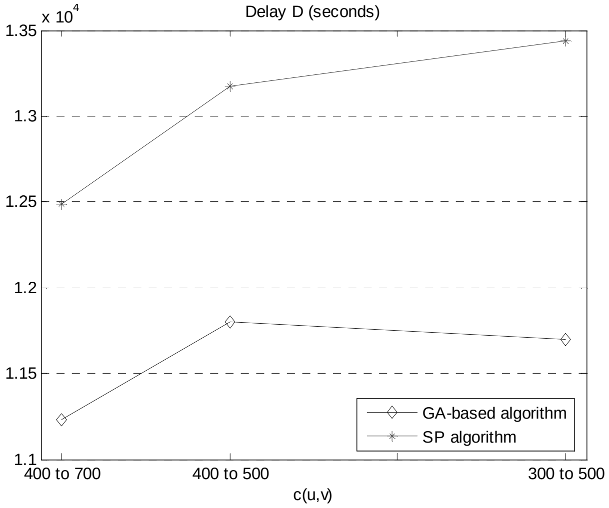

Fig. 11. Delay *D* under different storage capacity of mobile devices.

It is important to note that the results obtained in this section for $c(u,v)$ varying from 400 to 700 and a number of messages between 400 and 500 are not the same from the previous section because the scenarios are simulated with different random seeds and network topologies. However, for both sections the results have the same behavior.

Table IV presents the average number of hops used by each routing algorithm. Again, as SP algorithm considers only the number of hops to route decision, it maintains this number constant. On the other hand, the GA-based algorithm uses more hops when the routing complexity increases. This is because it searches alternative routes to avoid that a large number of messages passing through the same edge.

TABLE IV
AVERAGE NUMBER OF HOPS UNDER DIFFERENT STORAGE CAPACITY OF MOBILE DEVICES.

Algorithm	Hops for $c(u,v)$ = [400-700]	Hops for $c(u,v)$ = [400-500]	Hops for $c(u,v)$= [300-500]
SP	24.9	24.9	24.9
GA	28.8	28.8	28.9

C. Discussion

The results above suggest that when the network resources are scarce the improvement obtained by GA-based algorithm is higher if compared with SP algorithm. This is because the SP algorithm considers only the number of hops for route decision. This suggests that with little resources, routing algorithms more “intelligent” are necessary to achieve a good performance, and available information can be used to optimize the performance. Moreover, the good results obtained by the GA-based algorithms can be explained by the main characteristic of the algorithm: it searches the combination of routes above a minimum delivery probability and having the lesser delay. This DP_{min} can be adjusted according to the application needs.

At last, we compute the mean simulation time and the mean number of generations used by the GA-based routing algorithm. It takes 364.75 seconds in average to converge, for the simulated scenarios in this section. The mean number of generations necessary to find the routes is 117.83. This means that the GA-based routing algorithm takes a time (364.75 seconds in average) to find routes lesser than the leaving time ($w(u,v)$) of mobile devices on each edge, i.e. it takes an acceptable time. This time is for 200 generations, and we see that the GA-based routing algorithm spends on average 117.83 generations until get the routes. This way, both mean simulation time and the number of generations are appropriate for the simulated scenarios.

These measures can vary if we increase the number of sessions. For example, when the number of sessions is higher than the values used in the simulated scenarios (12 sessions), the GA-based routing algorithm will take more time and more generations until converge. On the other hand, if we increase the number of sessions, the network complexity increases too, and consequently, the improvement on performance obtained by the GA-based algorithm is higher when compared with simple approaches as the SP algorithm that takes in account only the number of hops.

VII. CONCLUSION AND FUTURE WORKS

Future DTN nodes will likely have to support a number of different routing strategies and protocols in order to operate efficiently in the vast diversity of environments in which the node may find itself. So we analyze two anycast routing approaches: a very simple that considers only the number of hops (SP algorithm), and a routing algorithm that uses GAs to perform the anycast routing in DTNs. Simulation results showed that the GA-based routing algorithm can reduce the average delay when compared with the SP algorithm and it maintains the delivery probability above a threshold (DP_{min}) in the simulated scenarios.

These improvements obtained by GA-based algorithm are emphasized when the networks conditions are more challenging, i.e. when the storage capacity of mobile devices is lower and/or the number of messages to be sent by source nodes for each anycast session is higher. This means that when the buffer size in DTNs is scarce, routing algorithms more complex are required to achieve a good performance.

As future works, new schemes can be studied and developed to the GA-based routing algorithm converges faster. Moreover, we can analyze anycast applications in DTNs to define the most appropriate value for DP_{min}.

ACKNOWLEDGMENT

The authors gratefully acknowledge financial support from the FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais) under Project TEC APQ-4089-6.01/07.

REFERENCES

[1] V. Cerf et al., “Delay-tolerant network architecture,” *IETF RFC 4838*, informational, April 2007. Available: http://www.ietf.org/rfc/rfc4838.txt

[2] K. Scott and S. Burleigh, “Bundle protocol specification,” *IETF RFC 5050*, experimental, November 2007. Available: http://www.ietf.org/rfc/rfc5050.txt

[3] K. Fall and S. Farrell, “DTN: an architectural retrospective,” *IEEE Journal on selected areas in communications*, vol. 26, no 5, pp. 828-836, June 2008.

[4] Z. Zhang, “Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges,” *Communications Surveys & Tutorials*, vol. 8, no 1, pp. 24-37, first quarter 2006.

[5] K. Kotecha and S. Popat, “Multiobjective genetic algorithm based adaptive QoS routing in MANET,” *IEEE Congress on Evolutionary Computation (CEC)*, pp. 1423-1428, September 2007.

[6] M. Al-Ghazal, A. El-Sayed, and H. Kelash, “Routing optimization using genetic algorithm in ad hoc networks,” *IEEE International Symposium on Signal Processing and Information Technology*, pp. 497-503, December 2007.

[7] M. Abdulla and R. Simon, “A simulation analysis of multicasting in delay tolerant networks,” in *Proc. of the Winter Simulation Conference – WSC*, pp. 2234-2241, 2006.

[8] W. Zhao, M. Ammar and E. Zegura, "Multicast routing in delay tolerant networks: semantic models and routing algorithms," in *Proc. of the SIGCOMM DTN Workshop*, August 2005.

[9] Y. Gong et al., “Anycast routing in delay tolerant networks,” *Global Telecommunications Conference (GLOBECOM)*, pp. 1-5, November 2006.

[10] J. Inagaki, M. Haseyama, and H. Kitajima, “A genetic algorithm for determining multiple routes and its applications,” *IEEE International Symposium on Circuits and Systems*, vol. 6, 1999.

[11] M. Munemoto, Y. Takai, and Y. Sato, “ A migration scheme for the genetic adaptive routing algorithm,” *IEEE Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 2774-2779, 1998.

[12] Y. Leung, G. Li, and Z. B. Xu, “A genetic algorithm for the multiple destination routing problems,” *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 4, pp. 150-161, November 1998.

[13] H. Chen and B. Sun, “Multicast routing optimization algorithm with bandwidth delay constraint based on GA,” *Journal of Communications and Computer*, vol. 2, pp. 63-67, May 2005.

[14] L. S. Randaccio and L. Atzori, “Group multicast routing problem: A genetic algorithms based approach,” *Computer Networks*, vol. 51, no. 4, pp. 3989-4004, 2007.

[15] E. R. Silva and P. R. Guardieiro, “Anycast routing in delay tolerant networks using genetic algorithms for route decision,” *11th International Conference on Computer and Information Technology*, pp. 65-71, 2008.

[16] A. Ferreira, “Building a reference combinatorial model for MANETs,” *IEEE Network*, vol. 18, no. 5, pp. 24-29, September-October 2004.

[17] E. W. Dijkstra, “A note on two problem in connexion with graphs,” Numerische Mathematik 1, pp. 269-271, 1959.

[18] David E. Goldberg, “Genetic algorithms in search, optimization and machine Learning,” Massachusets, Addison-Wesley, 1989.

[19] S. Jain, K. Fall, and R. Patra, “Routing in a delay tolerantnNetwork,” *Special Interest Group on Data Communications - SIGCOMM*. August/September 2004.

[20] I. Kaj and R. Gaigalas, “Stochastic simulation: Waxman random network topology generator,” Available: http://www.math.uu.se/research/telecom/software/.

Ederson Rosa da Silva was born in São Paulo, Brazil, in 1984. He received his B.E. degree in Electrical Engineering from Federal University of Uberlândia in 2007. He is currently working toward the Ph.D. degree in Electrical Engineering from Federal University of Uberlândia.

His research interests include performance analyses of communication networks, delay tolerant networking, and genetic algorithms.

Paulo Roberto Guardieiro is a Full Professor at Universidade Federal de Uberlândia, Brazil, where he works since 1978. He received his B.E. degree in Electrical Engineering from Universidade Federal de Uberlândia (UFU) in 1978, the degree of M.E.E. from the Instituto Tecnológico de Aeronáutica (ITA), in 1984 and the Ph.D. degree in Electrical Engineering from the UNICAMP in 1991.

He is coordinator of the Computer Networks Laboratory of the UFU and member of the Brazilian Society of Telecommunications (SBrT). His research interests include mobile communications, multicast, QoS guarantees and DTN’s.