A Markov Decision Process for Routing in Space DTNs with Uncertain Contact Plans

Fernando D. Raverta*†, Ramiro Demasi*‡, Pablo G. Madoery*†, Juan A. Fraire*‡§,
Jorge M. Finochietto*†, Pedro R. D'Argenio*‡§
*Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Córdoba, Argentina
†Facultad de Ciencias Exactas, Físicas y Naturales (FCEFyN), UNC. Córdoba, Argentina
‡Facultad de Matemática, Astronomía, Física y Computación (FAMAF), UNC, Córdoba, Argentina
§Department of Computer Science, Saarland University, Saarbrücken, Germany

***Abstract*—Delay Tolerant Networking (DTN) has been proposed to provide efficient and autonomous store-carry-and-forward data transport for space-terrestrial networks. Since these networks relay on scheduled contact plans, Contact Graph Routing (CGR) can be used to optimize routing and data delivery performance. However, scheduling uncertainties and faults induced by the harsh space environment can provoke different network connectivity than the one assumed in the provisioned contact plan. In this work, we develop a theoretical model based on a Markov Decision Process (MDP) to determine the Best Routing Under Failures (BRUF). Existing routing solutions are thus compared with the analytical bound obtained from implementing BRUF in PRISM. Results over random networks prove that state-of-the-art CGR is close to the theoretical delivery ratio and that supervised data replication is mandatory to further improve the performance under uncertain contact plans.**

***Index Terms*—Delay Tolerant Networks, Space and Satellite Networks, Contact Graph Routing**

I. Introduction

Large-scale satellite networks are becoming increasingly popular as a means to provide high quality imagery, video and communication services around the globe [1]. Efficient space-terrestrial communication technologies, capable of successfully moving large volumes of data between space and ground networks, are a key element in these networks. In this context, Delay Tolerant Networking (DTN) has been identified as a novel approach which can meet this goal in a cost-effective way by relaxing communication requirements and network infrastructure usually assumed in traditional protocols. The DTN architecture, originated from deep-space and interplanetary networking, embraces the concept of occasionally-connected networks that may suffer from frequent partitions, high delay, and that may be comprised of more than one divergent set of protocols [2]. To this end, a *bundle layer* that exists at a layer above the transport (or other) layers of the network, employs a persistent storage on each DTN node to store-carry-and-forward data packets called *bundles* as transmission opportunities become available.

In the case of space-based networks, the forthcoming episodes of communications (a.k.a. *contacts*) and their properties can be determined in advance based on orbital dynamics. These types of deterministic DTNs are known as scheduled DTNs and can take advantage of a *contact plan* comprising the future network connectivity in order to optimize data forwarding. However, scheduled routing solutions such as Contact Graph Routing (CGR) assumes the estimation of the future topology status is highly accurate [3]. Indeed, CGR does not consider scheduling uncertainties such as transient or permanent faults of nodes, antenna pointing inaccuracies or unexpected interferences.

Authors in [4] studied satellite networks under opportunistic and probabilistic routing solutions [5]. Although useful to minimize calculation effort and to avoid relying on a timely contact plan distribution, neglecting topological predictability severely undermines overall performance in space-terrestrial networks. Instead, DTN nodes can take advantage of contact plans as it avoids training overhead and facilitates audition, control and troubleshooting. In this regard, other works focused on Opportunistic CGR (O-CGR) have sought to extend CGR to react when unplanned (opportunistic) contacts occur [6], but the topological information encoded and distributed in the contact plan was still assumed accurate. After analyzing CGR reactions to contact prediction inaccuracies and faults in [7], authors studied different replication strategies for space-terrestrial DTNs under uncertain contact plans [8]. Nonetheless, results proved that there is not an optimal routing scheme for all uncertainty ranges in all types of scenarios under all types of traffic. Indeed, deciding a single routing framework in space DTNs with potentially inaccurate contact plans is still an open research question.

In order to deal with routing in space DTNs under uncertain contact plans, we propose a first theoretical model to determine the optimal routing solution in any possible scenario. In particular, given a space DTN described by some traffic to be delivered to destination, and a contact plan where each contact has a probability of failure, we seek to determine the routing decisions which maximizes the probability of delivering that traffic. We model the problem using a Markov Decision Process (MDP) and implement it in PRISM [9]. The model serves as an upper theoretical bound not only to compare existing routing schemes but to configure optimal static routes in medium-sized space DTNs. We finally compare the optimal model decisions with those made by CGR and its variations as well as other routing solutions applicable in uncertain space DTNs.

This paper is organized as follows. Section II provides an overview of the problem, introduces the theoretical model, and describes its implementation. Then, results are analyzed in Section III and conclusions are drawn in Section IV.

II. System Model

A. Uncertainty Model

In this work, we study the impact that uncertain events may have on space-terrestrial DTNs that use scheduled contact plans. These include faults in the nodes, unwanted interference hindering the proper utilization of a link, antenna pointing inaccuracies, unexpected power outages, equipment rests, or even last-minute mission commands modifying the topology issued after provisioning the contact plan.

In order to model unplanned events, a simple uncertainty model is considered. In particular, a contact can be suppressed (from the start to the end of the contact) from the contact topology with some given probability Pf. In this model, faulty contacts are independent of each other, meaning that a node might still implement some contacts while others fail at the same time. As a consequence of these uncertainties, the contact plan assumed by the routing algorithm, in this case CGR, may not always represent the actual topology of the network. This means that the optimal route table calculation might no longer hold and data might need to be rerouted thus bounding the performance of the space-terrestrial network.

Figure 1 illustrates a simple example of the proposed model. A network topology of 3 DTN satellite nodes is captured by means of 3 states (k_1, k_2, k_3) of 30s duration in which three contacts labeled $c_{ki,j,k}$ are established in state ki between nodes j and k. In state k_1, a directed contact $c_{k1,0,1}$ between nodes 0 and 1 has a capacity to transfer 30 traffic data units which can later take advantage of $c_{k2,1,2}$ to deliver data to node 2. An alternative and direct path from node 0 to 2 can be used at k_3 via contact $c_{k3,0,2}$. In this example, node 0 generates 5 traffic data units destined to node 2 at the beginning of state k_1. If a CGR algorithm is used, node 0 can use the contact plan with contacts $c_{k1,0,1}$, $c_{k2,1,2}$ and $c_{k3,0,2}$ to compute two possible routes, namely R_1 and R_2. R_1 contains two hops (two transmissions) and can deliver the traffic with an earliest delivery time metric of 60s whereas R_2 can deliver the same traffic at a later time of 200s but using only one hop (one transmission).

If R_1 is chosen and the uncertainty model injects a fault in contact $c_{k2,1,2}$ (with $Pf = 0.5$), node 1 will not be able to deliver the traffic to node 2 in this topology. Moreover, the traffic will remain stuck in node 1 storage since no more routes exist to reach destination. However, if node 0 knew that route through contacts $c_{k1,0,1}$ and $c_{k2,1,2}$ has a higher probability of failure than the route through $c_{k3,0,2}$, then choosing the latter would provide a higher averaged delivery ratio. Indeed, under uncertain contact plans, the optimality criteria for a route shall also consider the overall delivery probability. The objective of this study is to obtain a theoretical model to study the upper bound delivery ratio and transmission efficiency in space-terrestrial DTNs under such uncertain contact plans.

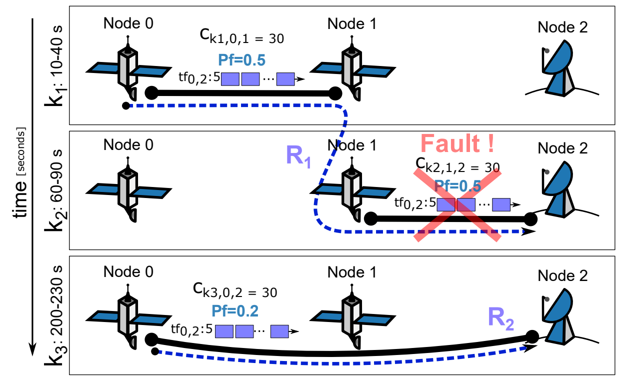


Fig. 1. Routing in scheduled scenario with faults

B. Markov Decision Processes

A Markov Decision Process (MDP) is an appealing mathematical framework for modeling systems that use both probabilistic and non-deterministic behavior such as the data flow over uncertain contact plans. These models offer an effective means for describing processes in which sequential decision making is involved. Many applications can be found across an increasing number of fields including economics, biology and engineering [10]. In general, the "solution" to a MDP is a control policy which either minimizes or maximizes a particular cost defined with respect to the states in the MDP.

Formally, a MDP $\mathcal{M}$ is a tuple $(S, Act, \mathbf{P}, l_{init}, AP, L)$ where

- ❒ S is a countable set of states with initial distribution $l_{init} : S \to [0,1]$
- ❒ Act is a finite set of actions
- ❒ $\mathbf{P} : S \times Act \times S \to [0,1]$ is a transition probability function such that for all $s \in S$ and $\alpha \in Act$: $\sum_{s' \in S} \mathbf{P}(s, \alpha, s') \in \{0, 1\}$
- ❒ AP is the set of atomic propositions and labeling $L : S \to 2^{AP}$

In order to solve MDPs, non-determinism is resolved by the so called *policy*. A *policy* for MDP $\mathcal{M}$ is a function $\mathcal{G}$ that for a given finite sequence of states through $\mathcal{M}$ yields an action to take next. In particular, we need a probability space over infinite paths to formally reason about MDPs. However, a probability space can only be constructed once all the nondeterminism has been resolved. In MDP, each possible resolution of nondeterminism is represented by a policy, which is responsible for selecting an action in each state of the model, based on the history of its execution.

In practice, MDP are attractive to minimize or maximize either the probability of a specific set of paths, or the expected value of some random variable. In fact, when using an MDP to model and verify quantitative properties of a system, this corresponds to evaluate the *best* or *worst-case* behavior that can arise. For instance, we are interested in "the maximum probability of a bundle being delivered". Commonly, Probabilistic Computation Tree Logic (PCTL) [11] serves as a temporal logic to specify important properties of MDPs. Then, model checking algorithms [12] are used to calculate the probability that the MDP satisfies the given property.

Among available probabilistic model checkers, PRISM has been widely used to model and analyze formal models of systems that exhibit random or probabilistic behavior [9]. Indeed, PRISM has been used to analyze systems from many different application domains including communication and multimedia protocols, randomized distributed algorithms, security protocols, biological systems, and many others. PRISM can solve several types of probabilistic models including MDPs. To this end, PRISM takes as input a description of a system written in the *PRISM language*, a simple, state-based language. Initially, the tool constructs the model from this description and computes the set of reachable states to then execute automated analysis of a wide range of quantitative properties of this model. In the PRISM semantic, each module is considered a process and they are composed in parallel using a handshaking semantic (process synchronized actions which have the same name). Each state of the MDP is described by the union of the states of each integrating node.

C. Best Routing Under Failures

In this work, we transform the problem of giving the best routing decisions for each bundle in a DTN under uncertainties, into the problem of choosing a policy in a MDP. Therefore, we build a MDP which encodes all possible routing decision for a given bundle, and then we compute a policy which maximizes the probability that the bundle reaches its destination. The model is coined Best Routing Under Failures (BRUF) and is implemented and analyzed using the aforementioned PRISM framework[1].

In order to introduce BRUF, let us consider the example network depicted in Figure 1 comprising 3 nodes and 3 contacts. Assume the origin node 0 (i.e., traffic source) has a bundle destined to node 2 at time 0. The corresponding PRISM source code for such scenario is illustrated in Figure 2. Note that we have introduced the keyword *mdp* at the beginning of the file to indicate the model is a MDP. We define two global integer variables *NUM_OF_NODES* and *NUM_OF_TS*, which represent the number of nodes and time stamps (states), respectively. For any given network, we define one module in PRISM called *bundle* to describe all possible routing decisions for the bundle in this network. The definition of the *bundle module* contains two parts: variables and commands.

The bundle module implements two integer variables: *node* and *ts*, describing which node is currently carrying the bundle and the time in which such event takes place respectively. The *node* variable has a range $[0 \ldots NUM\_OF\_NODES - 1]$ and is initialized in 0 (node 0 is the traffic source in this example). The *ts* variable has a range $[0 \ldots NUM\_OF\_TS]$ and is initialized in 0 (the first state in the topology) and represents the current time-stamp value.

```
mdp

const int NUM_OF_NODES = 3;
const int NUM_OF_TS = 3;

module bundle
  node: [0..NUM_OF_NODES - 1] init 0;
  ts: [0..NUM_OF_TS] init 0;

  [send_n0_n1_0] ts=0 & node=0 → 0.50: (ts'=ts+1) +
                                 1 - 0.50: (node'=1);
  [send_n1_n2_1] ts=1 & node=1 → 0.50: (ts'=ts+1) +
                                 1 - 0.50: (node'=1);
  [send_n0_n2_2] ts=2 & node=0 → 0.20: (ts'=ts+1) +
                                 1 - 0.20: (node'=2);
  [next] ts < NUM_OF_TS → (ts'= ts + 1);
endmodule
```

Fig. 2. BRUF model in PRISM for the network in Figure 1

The behavior of the bundle module in BRUF is described by commands, comprising a guard and one or more updates. In Figure 2, the left part of the right arrow is the guard of each command, and the update is defined by the following part. Each update describes a transition that the module can make if the guard is true. A transition is specified by giving the new values of the variables in the module, possibly as a function of other variables. Moreover, each update is also assigned a probability which will be assigned to the corresponding transition. We define a command for each contact in the Contact Plan and the transition probability matches the contact failure probability (Pf). In other words, this command models the case where the bundle is routed toward the neighbor on the other end of the contact.

Two possible outcomes are possible in each command depending on the failure probability: ***(i)*** the contact fails and the bundle stays in the sending node or ***(ii)*** the bundle is successfully transmitted and now it is carried by the receiving node[2].

For example, considering the contact $c_{k1,0,1}$ a command with label *send_n0_n1_0* is generated. The command guard is ts=0 & node=0 which means the bundle can be sent using this contact at time 0 and if it is carried by node 0. There are two possible updates in this command:

1) $0.50 : (ts' : ts + 1)$ - It means that the value of ts is increased with probability 0.50. This behavior describes the situation when the contact fails and then the bundle will remain at the source node 0. The time-stamp is increased since the source node will not be aware of the contact failure until it finished.
2) $1 - 0.50 : (node' = 1)$ - The case where the bundle is successfully transmitted to node 1 and it happens with

[1]Interested readers can access publicly available code in the following repository: https://bitbucket.org/rdemasi/bruf_stint_2018.

[2]For the sake of correctness, the BRUF model also includes auxiliary binary variables on each *send* command to refrain from generating transmission updates that might derive in routing loops. In other words, when two nodes have the same probability of choosing each other as next hops, auxiliary variables force the model to consider a unique data path among two of equal value.

probability 0.5.

Similarly, we define the commands [send_n1_n2_1] and [send_n0_n2_2] to model the behavior of the rest of the contacts in the network. Additionally, we have a special command ([next]) to increase the variable *ts* when it is less than the number of total time stamps defined in the model. This command models the case in which the bundle is not sent using any other command in current time stamp.

In order to perform model checking, PRISM first needs to construct the corresponding probabilistic model. During this process, PRISM computes the set of states which are reachable from the initial state and the transition matrix which represents the model. The transition matrix generated by PRISM for our example is depicted in Figure 3. It comprises all possible situations or *states* a given bundle can be found along its lifetime. Each state contains the value of the variables *node* and *ts* described as a pair *(node value, ts value)*. For instance, the initial state labeled by 0 contains the pair $(0, 0)$ which characterizes the state in which node 0 has the bundle at time 0. Moreover, each state contains (nondeterministic) choices, each of which is essentially a probability distribution over successor states that we can view as a set of transitions. For the initial state, we can observe the nondeterministic choice between actions *next* and *send_n0_n1_0*. If the former, *ts* is increased by one and we move to state 1, i.e., $(0, 1)$. If the latter, with probability 0.5 send the bundle successfully to node 1 reaching the state 4, i.e., $(1, 0)$; and with probability 0.5 message sending fails and we move to state 1, i.e., $(0, 1)$.

When model checking some properties of MDPs, PRISM can generate an optimal policy, i.e. one which corresponds to either the minimum or maximum values of the probabilities or rewards computed during verification. Recall that, for MDPs, PRISM quantifies over all possible policies, i.e. all possible resolutions of nondeterminism in the model. PRISM allows formulas of the form $P^{max}_{=?}\ [\psi]$, i.e., "what is the maximum probability that path formula ψ is true?". Note that a path property is a formula that evaluates to either true or false for a single path in a model. Although the model allows for others queries, we are interested in the following property: $P^{max}_{=?}\ [F \text{ node} = 2]$[3], that is, "what is the maximum probability that a bundle sent from source node 0 at time 0 is delivered to node 2?". The result computed by PRISM is that $P^{max}_{=?}\ [F \text{ node} = 2]$ is equals to 0.8.

Furthermore, PRISM also generates the optimal policy for this property, from which a route decision can be obtained. In Figure 3 we highlight the transitions from state 0 to state 10 which corresponds to the policy that maximizes the probability that the bundle reaches node 2. This police can be interpreted as sending the bundle toward the $c_{k2,0,2}$, because the probability of reaching the destination toward this direct route is higher than sending the bundle using the 2 hops route through contacts $c_{k1,0,1}$ and $c_{k2,1,2}$. However, if the $c_{k2,0,2}$ had a probability of failure greater than 0.5, the resulting

[3]The property F prop is true for a path if prop eventually becomes true at some point along the path.

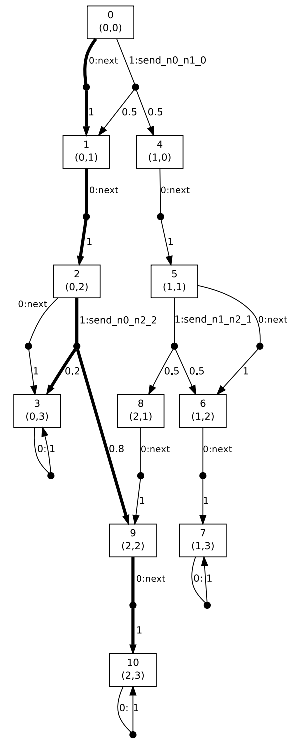


Fig. 3. Transition matrix for the network in Figure 1. Optimal routing policy highlighted as thick bold line.

best routing would be via $c_{k1,0,1}$. If forwarding on $c_{k1,0,1}$ is successful, the bundle will use $c_{k2,1,2}$ as next hop contact. Otherwise, node 0 should transmit the bundle toward $c_{k2,0,2}$. In general, one BRUF model can be solved for each source-destination pair in the traffic matrix in order to derive optimal route tables under uncertain contact plans.

III. Analysis

In order to evaluate the theoretical model we compare the routing results with existing schemes. To this end, 10 random topologies comprised of 8 nodes over a total duration of 100 seconds are fragmented in 10 seconds episodes. In each episode, the connectivity between nodes (i.e., presence of contacts) is based on a contact density parameter which can take values between 0.0 and 1.0. Similarly to the analysis in [8], such density parameter was set to 0.2. Furthermore, an all-to-all traffic pattern was assumed and bundles sizes are set small enough to avoid congestion biases. Besides congestion-free, channels are also configured as error-free (i.e., no packet drop) in order to focus the analysis only on the uncertainty phenomena as follows. Each routing algorithm is simulated over each of the 10 networks for different contact failure probabilities which ranges between [0,1] using DtnSim [13], a publicly available simulator for scheduled DTNs. In particular, the contact failure probability is the probability for every contact in the topology of not occurring and thus of impeding the course of data between the connected nodes pair. For each contact failure probability we perform 100 simulations (repetitions) totaling 1000 simulations per evaluated routing solutions detailed below:

- BRUF: We send specific routing rules (i.e., static routes) to each node in the network which were computed using the MDP model implemented in PRISM.
- CGR: Current implementation of CGR which forward a bundle using the first contact of the route which has the *best delivery time* among all to the given destination.
- CGR-Hops: A different implementation of CGR presented in [14] which forwards a bundle on the first contact of the route which has the *least hop count* among all to the given destination.
- CGR-2Copies: A version of CGR with replication presented in [14], which sends the traffic via both the best delivery time and the least hop count routes, when different.
- CGR-FaultsAware: The current implementation of CGR based on a contact plan where future failures are encoded (i.e., the algorithm is able to know where and when faults will occur). Since error and congestion are not present in this analysis, this is the best achievable performance on each scenario.

A. Performance

To analyze and compare the performance of BRUF we plot the average delivery ratio for all simulations in Figure 4. For contact plans without failures (probability of uncertainty 0) all schemes provides perfect delivery ratio. On the other hand, for fully disconnected contact plans (probability of uncertainty 1) no scheme can deliver any data. As stated, CGR-FaultAware is able to exploit the knowledge of forthcoming faults to take optimal decisions to route data. However, given that faults are uncertain and cannot be scheduled in advance, this curve is plotted as a reference. The best realistic routing solution without data replication is BRUF, which is able to use the contact failure probability and the probabilistic MDP model to decide highly reliable routes. Indeed, if the hypothesis proposed in this work holds, BRUF is the theoretical upper bound for copy-less routing schemes with uncertain contact plans. Classical CGR based on delivery time follows BRUF quite closely and CGR-Hops shows the worst delivery ratio as it tends to choose single-hop routes limiting it reaction to unpredicted faults [8]. It is interesting to notice that copy-based CGR-2Copies outperform copy-less BRUF in terms of delivery ratio. This is explained by the fact that CGR-2Copies replicates data and is able to increase the chances of delivering data by choosing divergent paths, but more research is needed to determine theoretical bounds for copy-based solutions. Indeed, we claim that data replication is a suitable strategy to overcome uncertainties in scheduled DTNs and approach the hypothetical CGR-FaultAware performance. Nonetheless, data replication comes at the expense of more congestion (here disregarded) and more transmission effort hindering energy efficiency as discussed below.

Figure 5 illustrates the average energy efficiency metric for each of the simulated contact failure probabilities. We define energy efficiency as the number of delivered bundles over the number of transmissions made to achieve such goal (such metric is undefined for failure probability = 1.0 since no transmission can happen in a fully disconnected network). In other words, this metric gives an indication on how much effort the network made (in average) to deliver the data, but normalized for each data unit. When probability of uncertainty is 1, the contact plan is fully disconnected and no transmission can be made resulting in null efficiency. Evidently, CGR-Hops is the most efficient one because it honor routes with less hops rendering very low transmission effort. CGR-FaultAware is second in terms of effort ratio. The increase in efficiency ratio of CGR-FaultAware as faults becomes larger is explained because this scheme knows which contact will fail in advance and will refrain to transmit data from the very beginning if the route will fail in the future. Indeed, this is not the case of realistic routing whose efficiency decreases with the probability of uncertainty (i.e., attempts to deliver data are less successful in the presence of more faults). BRUF provides quite a stable efficiency between 0.45 and 0.4 for all failure rates, being particularly flat in the range of 0.0 up to 0.5. CGR-DeliveryTime efficiency drift from BRUF at very low

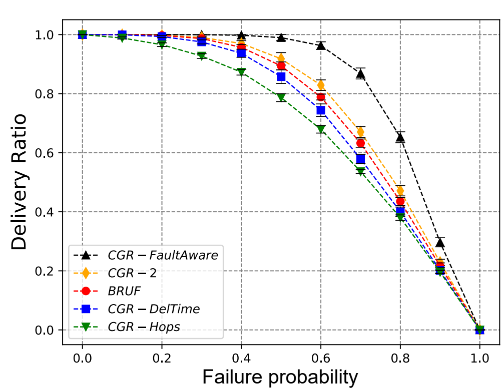


Fig. 4. Delivery ratio given the failure probability

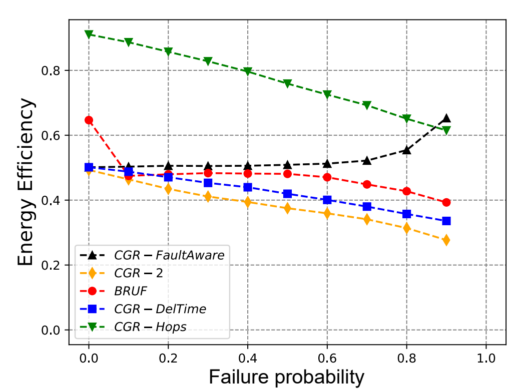


Fig. 5. Energy efficiency given the failure probability

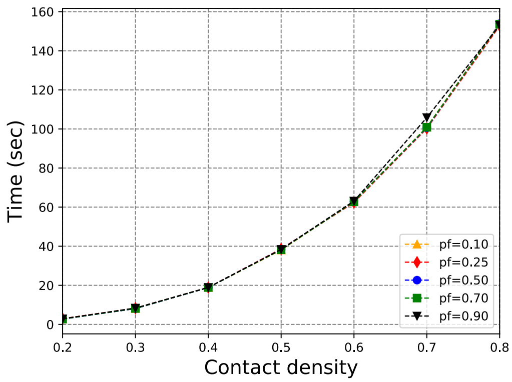
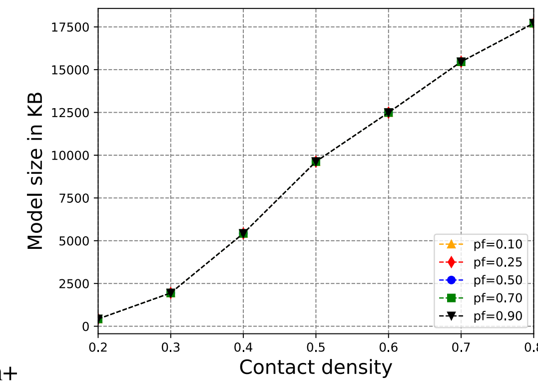


Fig. 6. Time for model construction and property checking



n n+

Fig. 7. Model size given a contact density

uncertainty probabilities but not as much as CGR-2Copies which provides the less energy efficiency ratio due to its data replication approach.

B. Scalability

BRUF scalability was also analyzed for similar random networks of 12 nodes and varying contact density parameter (i.e., the more contacts present, the more complex the decision matrix in BRUF). The following metrics were obtained from an Intel® Core™ i7-5820K CPU @3.30GHz × 4 processors with 9.8Gib running an Ubuntu 16.04 LTS 64-bit OS and PRISM version 4.4.

Figure 6 plots the processing time for PRISM to a) construct the logical model of the network and b) check the property to determine the best routing under uncertainties. It is worth mentioning the former is, in all cases, responsible for more than 98% of the measured time. Figure 7 summarizes the memory required to store the PRISM data model. Although small instances of the model can be solved limited recourses, the curve suggest that larger models might require significant processing time. As expected, results verifies that the processing time and memory utilization are rather insensitive towards the probability of failure.

IV. Conclusion

In this work we studied space-terrestrial Delay-Tolerant Networks (DTNs) where the contact plan encoding the future connectivity can result inaccurate due to scheduling uncertainties or unplanned events. In this context, the routing problem was for the first time tackled with a theoretical model based on a Markov Decision Process (MDP) to obtain Best Routing Under Failures (BRUF). BRUF was implemented in the PRISM tool and serves as an upper bound framework to compare the performance of existing DTN routing schemes such as CGR and its variations. Results showed that, although there is room for improvement, state-of-the-art CGR behaves quite well in random DTN with uncertainties. However, replication based solutions can improve delivery metrics at the expense of lower efficiency. Indeed, we propose as a future work to validate this conclusions in more realistic DTN topologies and to develop a N-copy BRUF model to correctly evaluate and improve contact-plan and replication based DTN routing solutions.

References

[1] J. Alvarez and B. Walls, "Constellations, clusters, and communication technology: Expanding small satellite access to space," in *2016 IEEE Aerospace Conference*, March 2016, pp. 1–11.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant networking architecture," Internet Requests for Comments, RFC Editor, RFC 4838, April 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4838.txt

[3] G. Araniti, N. Bezirgiannidis, E. Birrane, I. Bisio, S. Burleigh, C. Caini, M. Feldmann, M. Marchese, J. Segui, and K. Suzuki, "Contact graph routing in dtn space networks: overview, enhancements and performance," *IEEE Comms. Magazine*, vol. 53, no. 3, pp. 38–46, March 2015.

[4] M. Feldmann and F. Walter, "Routing in ring road networks with limited topological knowledge," in *2017 Int. Conf. on Wireless for Space and Extreme Environments (WiSEE)*, October 2017.

[5] S. Pathak, N. Gondaliya, and N. Raja, "A survey on prophet based routing protocol in delay tolerant network," in *2017 Int. Conf. on Emerging Trends Innovation in ICT (ICEI)*, Feb 2017, pp. 110–115.

[6] S. Burleigh, C. Caini, J. Messina, and M. Rodolfi, "Toward a unified routing framework for dtn," in *2016 IEEE Int. Conf. on Wireless for Space and Extreme Environments (WiSEE)*, Sept 2016.

[7] J. A. Fraire, P. Madoery, S. Burleigh, M. Feldmann, J. Finochietto, A. Charif, N. Zergainoh, and R. Velazco, "Assessing contact graph routing performance and reliability in distributed satellite constellations," *Hindawi Journal of Computer Networks and Communicationse*, in Press.

[8] P. G. Madoery, F. D. Raverta, J. A. Fraire, and J. M. Finochietto, "Routing in space delay tolerant networks under uncertain contact plans," in *2018 IEEE Int. Conf. on Comms. (ICC)*, May 2018, pp. 1–6.

[9] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, ser. LNCS, G. Gopalakrishnan and S. Qadeer, Eds., vol. 6806. Springer, 2011, pp. 585–591.

[10] M. N. Moghadasi, A. T. Haghighat, and S. S. Ghidary, "Evaluating markov decision process as a model for decision making under uncertainty environment," in *2007 International Conference on Machine Learning and Cybernetics*, vol. 5, Aug 2007, pp. 2446–2450.

[11] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," *Formal Asp. Comput.*, vol. 6, no. 5, pp. 512–535, 1994. [Online]. Available: https://doi.org/10.1007/BF01211866

[12] C. Baier and J. Katoen, *Principles of model checking*. MIT Press, 2008.

[13] J. A. Fraire, P. G. Madoery, F. Raverta, J. M. Finochietto, and R. Velazco, "Dtnsim: Bridging the gap between simulation and implementation of space-terrestrial dtns," in *Space Mission Challenges for Information Technology (SMC-IT), 2017 IEEE Int. Conference on*, Sept 2017.

[14] P. G. Madoery, J. A. Fraire, and J. M. Finochietto, "Congestion management techniques for disruption-tolerant satellite networks," *Int. Journal of Satellite Communications and Networking*, pp. n/a–n/a, sat.1210.