Modeling Routing in Scheduled Delay Tolerant Networks Under Uncertainties

Fernando D. Raverta
fdraverta
@gmail.com

Pablo G. Madoery
pablo.madoery
@alumnos.unc.edu.ar

Juan A. Fraire
juan.fraire
@unc.edu.ar

Jorge M. Finochietto
jorge.finochietto
@unc.edu.ar

CONICET, Instituto de Estudios Avanzados en Ingeniería y Tecnología (IDIT)
Universidad Nacional de Córdoba (UNC), Argentina

Abstract

Collaborative Earth observation satellite constellations are arising as a new paradigm with important advantages in comparison with traditional monolithic systems. In this context, Disruptions Tolerant Networking (DTN) protocols were proposed to provide efficient and autonomous store-carry-and-forward data transport. Although a scheduled contact plan can be used to optimize routing and data delivery metrics, significant challenges remain in studying the ability to recover from unplanned events. In order to evaluate different routing schemes under failures, we propose a framework based on Markov Chains and Probabilistic Computation Tree Logic which allows its comparison in terms of either the expected throughput or the probability of fulfilling a given mission. This approach provides considerable advantages with respect to traditional simulations since it allows the computation of expected values of network metrics instead of approximations without any error estimation.

1 Introduction

Earth observation through small satellites [1], as well as its use to transport data from and toward isolated planet areas [2], have been attracting lot of attention from government agencies as well as from private companies. This new paradigm offers an incremental access to space with gradual costs, in a flexible way, and with better adaptability to new mission requirements. In particular, the segmented architecture paradigm [1], which is being proposed by the Argentinean Space Agency (CONAE), suggests to decompose a large satellite with multiple functions into separate autonomous modules called segments with the capacity of sharing resources to fulfill a given mission.

However, the consequences of dividing a satellite into a distributed system imply important technology challenges which need to be tackled before any mission can be deployed in space. Particularly, in this work we will put the focus on the communications between segments and with the ground stations. Regarding this aspect, there exists a large number of technologies and protocols which have been developed for the Internet, but in general they are inefficient or unsuitable when used on satellite networks, because orbiting trajectories do not allow satellites to establish continuous and stable end to end paths among them [3].

As a consequence, many satellite networks do not satisfy the continuous connectivity principle between their nodes and have remained outside of the Internet paradigm because they have used incompatible specialized protocols which usually require several protocol adapters. However, NASA and other space agencies have recently decided to develop and standardize protocols which allow communication between these kind of networks which are called Delay and Disruption Tolerant Networks (DTN) [4]. At first, they were studied to implement interplanetary networking (IPN) [5], but they have been also recognized as a valid solution for satellite application since they can deal with intermittent channels, which are very common in low-orbit satellite networks (LEO) [6]. Since the beginning of DTN, there had been some significant advances like the architecture specification [7], the Bundle Proto-

---

In: Proceedings of the IV School of Systems and Networks (SSN 2018), Valdivia, Chile, October 29-31, 2018. Published at http://ceur-ws.org

col and the Contact Graph Routing Algorithm (CGR) definition in [8,9].

The Bundle Protocol works as a new protocol layer on top of the transport layer. Its function consists in overcoming the limited connectivity by applying a store-carry-and-forward approach. Under this paradigm, data is grouped into packets called *bundles* and they are transmitted as communication opportunities are available. In the case of satellite networks, the forthcoming episodes of communications 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 [10]. Indeed, CGR does not consider scheduling uncertainties such as transient or permanent faults of nodes, antenna pointing inaccuracies, unexpected interferences, or even last-minute mission commands modifying the topology issued after provisioning the contact plan.

Regarding scheduling under uncertainties, the authors have evaluated the behavior of the state-of-the-art routing algorithms by means of simulations [11,12]. From those experiences the authors have noticed the necessity of developing a model which enable a higher level of understanding of the network. As a first step on this way, we present here a method to build a Markov Chain which potentially encodes all possible network status for a given traffic, routing algorithm and link failure probability.

This paper is structured as follow. In Section 2 we provide some background on routing in scheduled DTNs and the failures model considered here. Then, in Section 3 we describe the proposed framework. Afterwards, in in Section 4 we use our method in a case study. Finally, we discuss and summarize this work in Section 5.

2 Routing and Failures

2.1 Network Model

In general, the DTN topology is time varying. This evolutionary nature can be captured by means of states, where each state is represented by a graph whose vertex are the network nodes and the links are the transmission opportunities available in that state. Each state is valid for a finite period of time, which is given by an initial time (t_{ini}) and a final time (t_{end}). Figure 1 shows a scenario with 3 nodes and a duration of 30 seconds divided into 3 states of 10 seconds each one. Other important contact attribute is the capacity, which represents the traffic volume that can be transmitted through that contact. The capacity of a contact, between nodes i and j at state k, is symbolized here as $C_{i,j,k}$, and we will measure it in bytes.

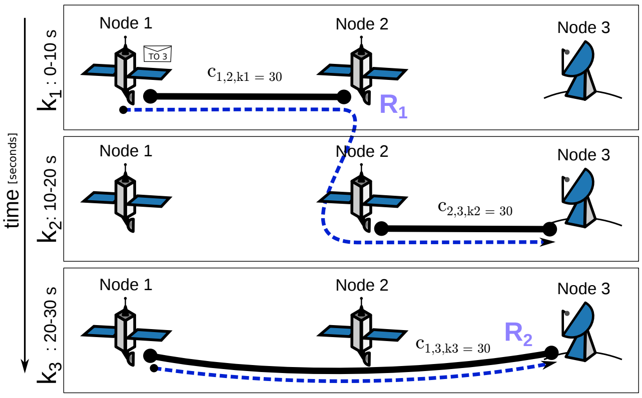

Figure 1: Routing in scheduled scenario

2.2 Scheduled Routing

Thanks to orbital mechanics, it is possible to codify the future transmission opportunities in a *contact plan*, which can then be used to make efficient routing decisions either in a distributed way (by providing the contact plan to all nodes and executing then the routing algorithm in each node) or in a centralized way (by directly providing a route table for each traffic). Particularly, the state-of-the-art algorithm for routing in this kind of networks is CGR and it is mostly thought to be executed in a distributed manner. Thus, when a node needs to send data, CGR first computes a route table considering many neighbors, and then it chooses the route which delivers the bundle as early as possible in time. Finally, the bundle is sent through the first contact of that route. If the receiver node is not the final destination, it will repeat the same procedure in a hop by hop basis until the bundle reaches the destination.

2.3 Failures

Unexpected events like antenna pointing inaccuracies, interferences, energy outages or even equipment rest, could cause that a contact which is present in the contact plan actually does not happen. Then, the routing algorithm which had planned to send a bundle through that failing contact, will need to make another decision to react to the unexpected event.

In order to explain the fault model adopted in this work, let's consider the example scenario in Figure 1. If it is considered that Satellite 1 has data to send to ground station, then there are two routes available R1 which consists in two transmissions (from satellite 1 to 2 at time 0 and from satellite 2 to ground station at time 10) and R2 consisting in 1 transmission (from satellite 1 to ground station at time 20). If CGR is the routing algorithm, then it will choose R1 since this provides the best delivery time to reach the destination. However, if a failure is encountered in $link_{2,3,k2}$, the traffic will not be able to be delivered because there will not be any extra routes from Node 2 to Node 3 in state 2. However, if a failure happens in $link_{1,2,k1}$, the situation is different. In that case, we assume that Node 1 can realize that the transmission has failed and that a route R2 can still deliver the bundle to Node 3. Finally, $link_{1,3,k3}$ can fail or succeed and considering all these possibilities allows us to compute the exact values of the quantitative properties of the network under failure.

3 Model

3.1 Framework overview

We propose a framework consisting of 3 stages as showed in Figure 2. A Markov Chain model is generated in order to potentially encode all possible states of the network for a given traffic and routing algorithm. Therefore the following components are needed as inputs for the Model Builder stage:

1. **The network contact plan**, which must be annotated with the probability of failure of each link.
2. **The network traffic**, which consists in a list of bundles. Each bundle must have a source and destination node ID, a size and a time of generation.
3. **The routing algorithm**, which is provisioned by implementing a programming interface consisting of 2 methods used by the model generator engine.

The output of the model builder is a tree in which each branch is a possible network execution. This model is then translated to a PRISM model [13], saving in each state only the required information to compute the desired properties. For instance, only the number of delivered bundles is needed to be saved in order to compute the expected network throughput. In this case, we call *"delivered"* to this attribute. The model is then loaded using the PRISM Model Checker tool and from there it is possible to compute the expected delivery ratio by using the following formula:

$$E(Delivery\_Ratio) = \frac{\sum_{b=0}^{N} b * P =?[F(delivered = i)]}{N}$$

Where N is the number of bundles in Traffic, and P=?[F (delivered=i)] is a Probabilistic Computation Tree Logic Formula (PCTL) which measures the probability of delivering i bundles in all the possible realization of the network.

Also, this model allows to compute the probability of fulfilling a given mission. For the sake of simplicity, we will consider that we fulfill a mission by delivering at least a given fraction f of the total traffic. Then, the next PCTL formula computes the desired probability:

$$Pr(successful\_mission) = P =?[F(delivered >= f{*}N)]$$

Furthermore, this model supports the specification of more complex mission properties like for example, those ones involving the delivery time.

3.2 Model Builder Description

The model is generated by the process described in Algorithm 1. It systematically explores all possible network states, generated by calling the *genNextState* method which interacts with the routing algorithm by means of the following interface:

1. **route routing(state, bundle)**: This method must return a list of contacts (route) which encodes the made routing decisions.
2. **void update(state, route)**: This method receives a route and it has to update the current state according to this routing decision. For example, in those algorithm (like CGR) which consider contact capacity, this method should update the residual capacity of each contact.

A detailed description of the *genNextState* routine is provisioned in Algorithm 2. Indeed, the main idea of this algorithm is to compute the routing decisions for the incoming traffic at a given state. Then, using this information it generates the state's children by considering any set of contact failures. In other words, given a set of contacts named as *contacts_to_use* that were chosen by the routing algorithm to be used at that moment, it takes any subset $failure\_set \in \mathcal{P}(contacts\_to\_use)$ and computes a new child state by considering: 1) The bundles for which the routing algorithm choose a route which contains a contact belonging to failure set, stays in the sender node. 2) The bundles whose contacts does not belong to the failure set are now in the receiver node. After that, each child state is linked with the parent in the Markov Chain by an edge whose probability *pl* is given by:

$$pl = \prod_{c \in failure\_set} pf(c) \times \prod_{c \in failure\_set^{c}} (1 - pf(c))$$

4 Case Study

In order to show the application of the proposed framework, we analyze the linear formation topology depicted in Figure 3, which is composed of 3 satellites

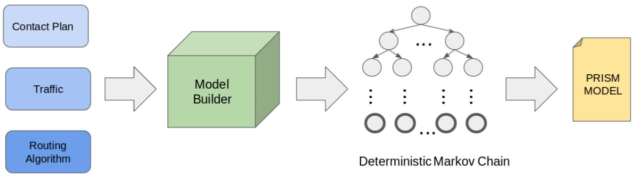

Figure 2: The proposed framework

Algorithm 1: Generate Model Algorithm

```
$states \leftarrow [initialState]$;
while not $states.empty()$ do
    $current \leftarrow states.dequeue()$;
    $states.enqueue(genNextStates(current))$;
return $initialState$;
```

Algorithm 2: gen_next_states

```
input  : Network status containing all
         information required by routing
         algorithm
output: Set of new network status generates
         from input status by making routing
         decisions
$contacts\_to\_use \leftarrow []$;
$routing\_decision \leftarrow []$;
for $bundle$ in $current.incomming\_bundles$ do
    $selected\_route \leftarrow routing(state, bundle)$;
    $routing\_decision.append((bundle, selected\_route))$;
    $update(state, selected\_route)$;
    $contacts\_to\_use.append(selected\_route.contacts())$;
for $n\_of\_faults \in range(len(contacts\_to\_use))$
  do
    $comb \leftarrow$
      $combinations(contacts\_to\_use, n\_of\_faults)$;
    for $fault\_set \in comb$ do
        $new\_state \leftarrow state.copy()$;
        for $bundle, contactinrouting\_decision$ do
            if $contactinfault\_set$ then
                $new\_state.update(bundle, FAILURE)$;
            else
                $new\_state.update(bundle, SUCCESS)$;
    $p \leftarrow compute\_prob(fail\_set, contacts\_to\_use)$;
    $current.add\_child(new\_state, p)$;
return $current.get\_childs()$;
```

that generate traffic for a unique ground station. In this formation satellites are equally spaced and follow very similar orbital trajectories. Among the many benefits of this disposition, satellites do not require complex transfer maneuvers if launched from the same vector. Also, since satellites perceive similar gravitational perturbations, significant savings in propellant for formation-keeping can be made. From a communications perspective, the topological stability of this formation also favors the simplicity of fixed antennas against complex gimbal mounts or electronically steered antennas for inter satellite links (ISLs). Similar topologies have been used in previous satellite DTN studies [14] and result particularly appropriate for Earth observation missions.

We evaluate two different routing algorithms in terms of its probability of fulfilling a given mission. On the one hand, *Direct Routing* is a simple scheme where satellite nodes can only deliver traffic to destination by using direct communication contacts. This scheme is very appealing when satellites have constrained onboard computers like those available on cubeSats. On the other hand, we also consider *CGR*, which is more demanding in terms of computing effort but with the advantage of being able to use routes with multiple hops to destination.

In this scenario we consider that each satellite generates one bundle directed to ground station at time 0 and we set the failure probability in 0.1 for the ISLs and 0.5 for the earth to satellite links (ESLs) . Regarding contact capacity, it is assumed that all traffic can be sent using any single contact and it will be available in the receiver node in the following time stamp (if a failure does not happen). Under this scenario, we evaluate the probability of fulfill 3 different missions for each routing algorithm: ***Mission A*** is fulfilled if at least one bundle is delivered to ground station. ***Mission B*** requires that more than 1 bundle be delivered. While ***Mission C*** is fulfilled if only if all the generated traffic is delivered to the ground station.

The results of the performed evaluation are showed in Figure 4. There we can see that direct routing presents the highest probability of fulfilling the *Mission A* since it always tries to send 1 bundle for each contact with the ground station, and this maximizes the chances of delivering at least 1 bundle. On the other hand, CGR always tries to send all bundles through the route with the best delivery time and in its eagerness to arrive before it wastes (in very unlikely cases) chances of sending at least 1 traffic for each contact with the ground station (which is a necessary condition in order to maximize the delivery of at least 1 traffic). However, as the mission requirements become more demanding CGR, which tries to download all the traffic for each of the 3 contacts, gets performances almost as good as the one gotten in the case of the least demanding mission. On the contrary, the behavior of direct routing is dramatically affected

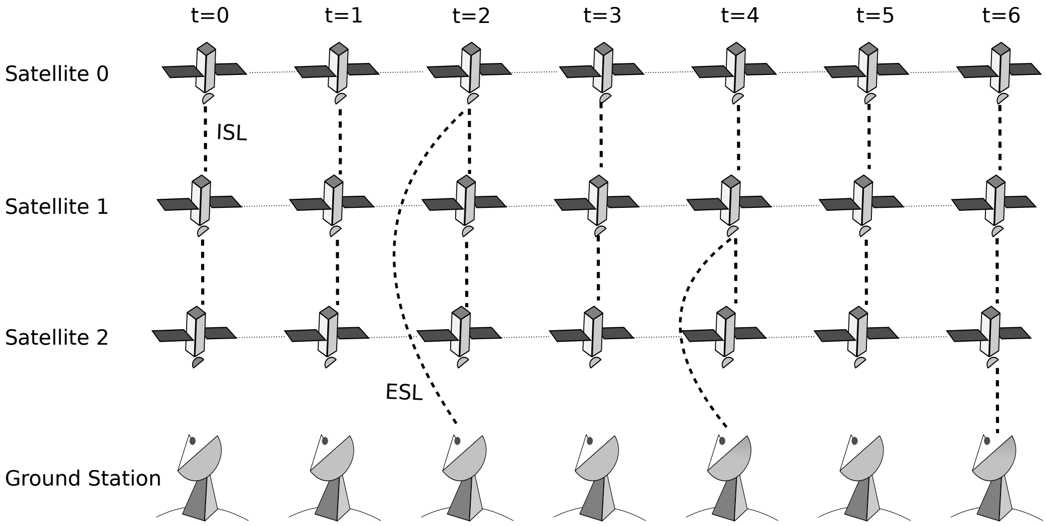

Figure 3: Case Study Contact Plan

when the mission requirements become more demanding.

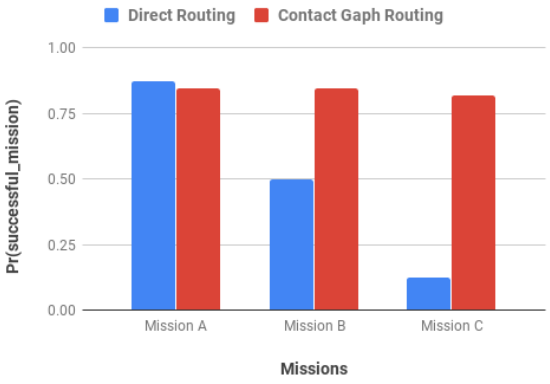

Figure 4: Successful probability results

5 Summary and Outlook

In this work we proposed a framework to systematically build a Markov Chain and encode all possible executions of a satellite network. Then, we gave a formula to compute the expected network throughput and the probability of fulfilling a mission as a way of showing the kind of information we can get from the model. After that, we use the introduced method in a case study.

Regarding the comparison between this method and the simulation approach, we showed that this one allows to compute the expected values of the network metrics (like throughput and latency) instead of approximations. This constitutes a significant step since we are working in the definition of a model to compute the best routing decisions under uncertainties, that is, the election of routes which maximize the expected throughput for a given network and traffic. Therefore, this framework will allow us to compare current routing solution with the optimal ones. From that analysis we expect to draw conclusion to improve the behavior of scheduled routing under uncertainties. Even more, we plan to contribute with the development of a fully-capable probabilistic CGR with enhanced uncertainty modeling, replication and recalculation features, which is very important for the DTN community since CGR is expected to be the de facto routing algorithm for this kind of networks.

References

[1] C. N. de Actividades Espaciales (CONAE), “Plan espacial nacional (PEN) 2004-2015, actualizacion 2010,” http://www.conae.gov.ar/prensa/PlanEspacial2004-2015.pdf, 2015.

[2] S. C. Burleigh and E. J. Birrane, “Toward a communications satellite network for humanitarian relief,” in *Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief*, ser. ACWR ’11. New York, NY, USA: ACM, 2011, pp. 219–224.

[3] C. Caini and R. Firrincieli, “Dtn for leo satellite communications,” in *Personal Satellite Services*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, G. Giambene and C. Sacchi, Eds. Springer Berlin Heidelberg, 2011, vol. 71, pp. 186–198.

[4] K. Fall, “A delay-tolerant network architecture for challenged internets,” in *Proceedings of the 2003 Conference on Applications, Technologies,*

Architectures, and Protocols for Computer Communications, ser. SIGCOMM '03. New York, NY, USA: ACM, 2003, pp. 27–34.

[5] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: An approach to interplanetary internet," *Comm. Mag.*, vol. 41, no. 6, pp. 128–136, Jun. 2003. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2003.1204759

[6] M. G. ONeill, H. Yue, S. Nag, P. Grogan, and O. L. de Weck, "Comparing and optimizing the darpa system f6 program value-centric design methodologies," AIAA Paper 2010-8828, 2010.

[7] 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

[8] K. Scott and S. Burleigh, "Bundle protocol specification," Internet Requests for Comments, RFC Editor, RFC 5050, November 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5050.txt

[9] S. Burleigh, "Contact graph routing, IETF-Draft draft-burleigh-dtnrg-cgr-01," Jul 2009.

[10] 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.

[11] P. Madoery, F. Raverta, J. Fraire, and J. Finochietto, "On the performance analysis of disruption tolerant satellite networks under uncertainties," in *Proceedings of the 2017 XVII RPIC Workshop*, September 2017.

[12] P. Madoery and F. Raverta and J. Fraire and J. Finochietto, "Routing in space delay tolerant networks under uncertain contact plan," in *International Conference on Communication 2018*, May 2018.

[13] M. Kwiatkowska, G. Norman, and D. Parker, "Prism: Probabilistic symbolic model checker." Springer, 2002, pp. 200–204.

[14] J. A. Fraire, P. G. Madoery, J. M. Finochietto, P. A. Ferreyra, and R. Velazco, "Internetworking approaches towards along-track segmented satellite architectures," in *2016 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, Sept 2016, pp. 123–128.