To link to this article: DOI :10.1002/sat.1159
URL : http://dx.doi.org/10.1002/sat.1159

To cite this version : Diana, Rémi and Lochin, Emmanuel and Franck, Laurent and Baudoin, Cédric and Dubois, Emmanuel and Gelard, Patrick DTN routing for quasi-deterministic networks with application to LEO constellations. (2015) International Journal of Satellite Communications and Networking. pp.1-18. ISSN 1542-0973

DTN routing for quasi-deterministic networks with application to LEO constellations

R. Diana[1,2], E. Lochin[1,2], L. Franck[3,2], C. Baudoin [4], E. Dubois[5], P. Gelard[5]

[1]ISAE, Université de Toulouse, Toulouse, France,
[2]TéSA, Toulouse, France,
[3] Telecom Bretagne - Institut Telecom, Toulouse, France,
[4]Thales Alenia Space,
[5]Centre National d'Etudes Spatiales (CNES), Toulouse, France

Abstract

We propose a novel DTN routing algorithm, called DQN, specifically designed for quasi-deterministic networks with an application to satellite constellations. We demonstrate that our proposal efficiently forwards the information over a satellite network derived from the Orbcomm topology while keeping a low replication overhead. We compare our algorithm against other well-known DTN routing schemes and show that we obtain the lowest replication ratio with a delivery ratio of the same order of magnitude than a reference theoretical optimal routing. We also analyse the impact of terrestrial gateways density and analyse DQN performances in heterogeneous cases.

1 Introduction

Satellite constellations have a dynamic character in terms of mobility and connectivity. However, their dynamics are mostly predictable and static routing, enabled by a precomputation of the network routes, can be applied to them [1][2]. The main drawback of pre-computed routing schemes is to be static and not tolerant to link outages. There are two design approaches for satellite constellations: with and without inter-satellite links (ISL). The present contribution focuses on the latter with the advantage of offering a simplified platform design. However, dynamic routing approaches, such as MANET routing, cannot be applied to these satellite constellations as most of the time, a continuous path between the source and destination may never be existing. The delay tolerant network (DTN) paradigm has been designed with that constraint in mind. However, common DTN routing algorithms are not optimized as they have been initially designed to perform over random topologies.

DTN was initially created for interplanetary networks [3] but received a great interest for opportunistic networks [4]. Several DTN routing schemes have been proposed for social and random networks [5][6][7][8], however the idea of using DTN routing algorithms over LEO satellite constellations is quite unexplored while is seen as an alternative solution for satellite applications [9]. The quasi-deterministic character of a satellite constellation makes current DTN routing proposals (specifically designed for random networks) not optimal over such specific topologies. We qualify a satellite topology as quasi-deterministic since the traffic nature and link outages might compromise the inherent deterministic character of the constellation. Furthermore, Section 4 shows that current implemented DTN routing schemes have a too high replication overhead for satellite networks where the cost of transmission must be taken into consideration. Existing proposals attempt to improve the delivery throughput based on the multi-commodity theorem [10]. The principle behind, which extends the definition of the max-cut min-flow theorem [11], is to build a graph by analysing a given quasi-deterministic topology in order to find the maximum achievable throughput that the network can handle [12]. Other schemes, such as Contact Graph Routing [13], propose to compute the optimal route for DTN messages forwarding. In 2012, its application in LEO satellite constellations was studied by C Caini and al in [14]. In this work, the authors analyse the interest brought by CGR algorithm in a highly predictable context. Indeed, topologies considered in this study are composed of one or two satellites and from 2 to 3 terrestrial nodes. These topologies are small sized and the traffic considered cannot be considered as a source of indeterminism. As a result, the system can be seen as purely deterministic and as it is small sized, computation cost is kept low. This is a major difference with the LEO topologies we are focused on, in which we consider more satellites and terrestrials nodes with an unknown traffic leading to a complex system with inherent indeterminism. All the previously mentioned approaches are based on pre-computed algorithms, requiring a computation effort closely dependant on system size and are not designed to be fault tolerant and to deal with indeterminism.

On the opposite of [15], we assume a LEO constellation without storage in the payload and no inter-satellite links either. The rationale here is to design low cost satellites (i.e., without complex on-board processing and storage) either to minimize the cost of the overall mission. In a context of scientific missions or data collection, this approach makes sense.

End-to-end communications are done in a multi-hop manner from gateways to gateways via satellites. This illustrates the intermittent character of the chosen topology since gateways are not always covered by a satellite. Over such a LEO constellation, pre-computed routing algorithms are often used. These algorithms compute routing tables of each gateway for a given duration $[t_0,\ t_0 + \Delta_t]$. Thus, routing tables are fixed during Δ_t and will not be recomputed even if an outage occurs during this period. Basically, a pre-computed algorithm corresponds to an implementation of an optimal algorithm if no failures occur and there is no buffer limitation. As a consequence, such a routing scheme is not resistant to link failure. If a link is down during a time before Δ_t, the delay to detect it, to compute new routing tables and to upload them in each node might result in several data losses.

In this paper, we propose to cope with this periodic computation scheme problems by designing an optimal DTN replication-like routing scheme for such quasi-deterministic networks. We focus on the optimal number of replication without too much delay compromise and we test our routing protocol over the Orbcomm topology [16]. The Orbcomm system offers M2M (machine to machine) global asset monitoring (data collection) and messaging services from its constellation of 29 LEO communications satellites orbiting at 775 km. The rationale of using Orbcomm is linked to its intermittent connectivity. This explains why this topology is best suited for users who send very small amounts of data and where services are much like email and messages based (as over DTN networks) [16]. In Section 2, we drive an analysis of the contacts and inter-contacts duration of Orbcomm nodes. Following this analysis, we propose a novel DTN routing algorithm called *DQN*. We first evaluate our proposal against other well-known DTN routing algorithms in Section 4. Then, we analyse our algorithm more deeply in Section 5 with three different studies. We first consider a variable gateways density to study the impact of gateways geographical distribution. Then, we study more realistic cases with the evaluation of DQN over network composed by an heterogeneous gateways density and finally we study the impact of network load on our algorithm. We conclude and discuss this work in Section 6.

This paper can be considered as an extension of the study presented in [17] in which some key ideas are established. However, we present here deeper analysis and some new ideas about both the topology used and the algorithm developed. The last part, focused on the network load impact on DQN is fully novel.

2 Analysis of the constellation

The objective of this section is to characterize the studied topology in terms of contacts and inter-contacts duration between gateways and their distribution. We also assess the inter-gateway availability. The study presented below is driven over a bipartite network: a constellation of satellites and a set of terrestrial gateways. For the sake of simplicity, we first consider that the terrestrial gateways are uniformly distributed (a more realistic scenario will be detailed Section 5.2). Based on Orbcomm, the constellation is formed by 35 satellites distributed over six orbital planes. Three orbital planes contain eight satellites and one contains seven satellites all uniformly spaced. A polar and a retrograde orbital plane, each holding two satellites are also present. In this study, the terrestrial gateways are uniformly spaced on Earth with an interval $d = 20°$. The whole network is presented in Table 1 and Fig.1.

We now present the hypotheses used to analyse this topology.

Table 1: Constellation characteristics

Orbital plane	Inclination	Altitude (km)	# of satellites
1	45°	≈ 827	8
2	45°	≈ 827	8
3	45°	≈ 827	8
4	45°	≈ 827	7
Polar	70°	≈ 744	2
Retrograde	108°	≈ 833	2

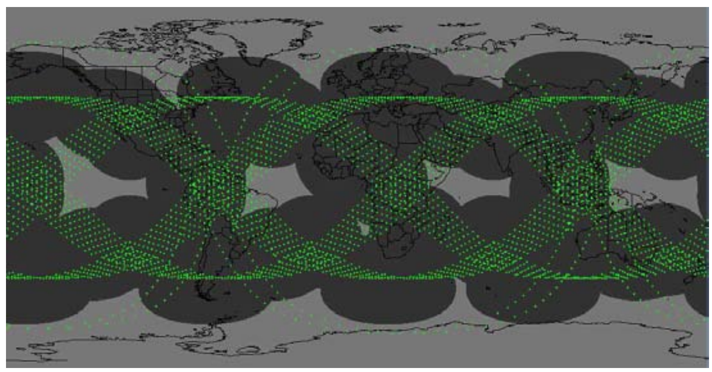


Figure 1: Orbital traces and footprints

2.1 Main hypothesis

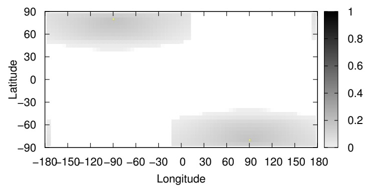
We consider that the satellites do not store any data and that there is no inter-satellite link (ISL). Thus, each satellite is considered as a simple data link between two terrestrial gateways. When a link is available, a contact between two terrestrial gateways is enabled and is defined by a contact duration. All data are carried between the terrestrial gateways where, in our experimental scenarios, some of them are senders while others are receivers. Note that each gateway has the knowledge of the geographical position of all other gateways.

2.2 Contacts and inter-contacts duration characteristics

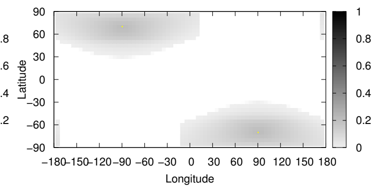
We first illustrate the gateway service availability which corresponds to the distribution of the immediately reachable locations around a gateway. Fig. 2 illustrates this repartition. In each subfigure, the white dot (in the middle of each the dark surface) corresponds to the reference gateway and the dark surface represents the reachable area. Each subfigure gives two reference gateways (one in the northern hemisphere and one in the opposite southern hemisphere) placed at the same longitude but at differeny latitudes. We define the availability as the ratio between the average contact duration and the total duration. An availability of 1 means that the gateway and the remote location are permanently connected. In other words, the darker the surface is, the more contact opportunities the gateways get. Note that this figure does not provide the exact distribution of contact and inter-contact duration according to the localisation of the gateways. Indeed, this is only an average. Nevertheless, a low availability corresponds generally to a low average contact duration and a long average inter-contact duration.

2.3 Routes characterisation and analysis

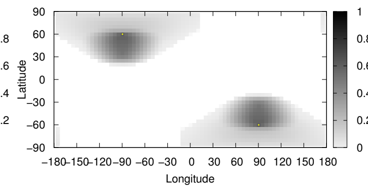
We now focus on the routing aspects and evaluate the possible existing paths. We have chosen two representative gateways, one sender that continuously sends data while the other receives them. Each



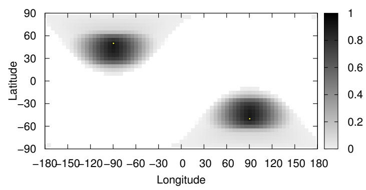
(a) $|Lat| = 80$



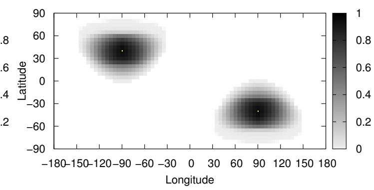
(b) $|Lat| = 70$



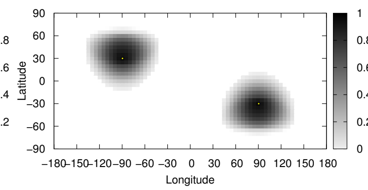
(c) $|Lat| = 60$



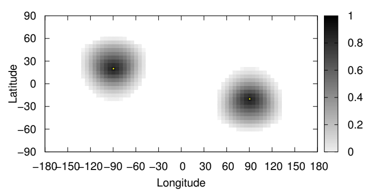
(d) $|Lat| = 50$



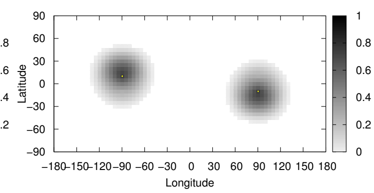
(e) $|Lat| = 40$



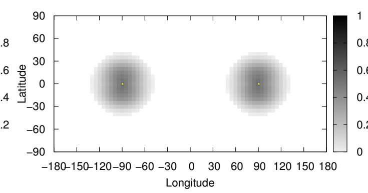
(f) $|Lat| = 30$



(g) $|Lat| = 20$



(h) $|Lat| = 10$



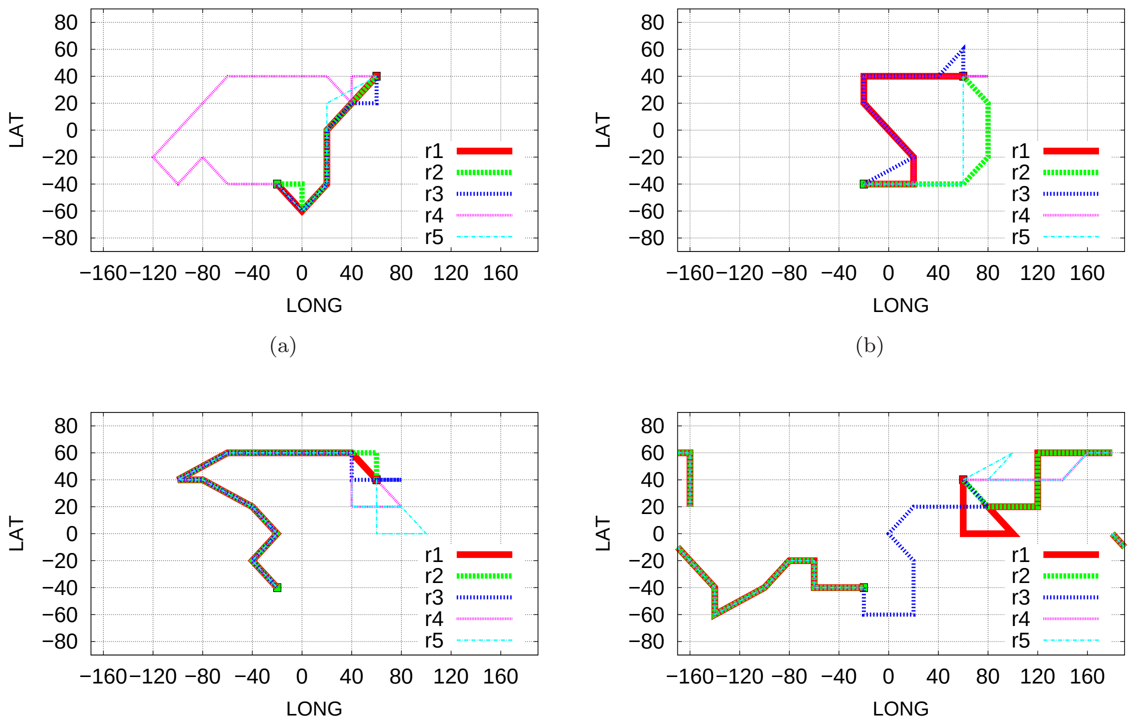
(i) $|Lat| = 0$

Figure 2: Spatial and temporal distribution of the contacts around a gateway (two separate gateways are represented in each figure).

of the 162 gateways replicates in an epidemic manner the data sent. This epidemic process results in a huge number of existing routes allowing to carry bundles from one source to several destinations. So, we choose to select the routes above a given maximum delay threshold. This therefore limits the number of routes considered. After applying this threshold, we classify the routes with the same order of magnitude of delay inside five groups denoted r_1 to r_5 in Fig. 3. Within the same group, there may be up to tens of routes with a close delivery time. The groups are sorted in ascending order. This means that copy of a message sent over a route from the group r_1 has a shorter delivery time than a copy following a route belonging to r_2. The remaining routes are not clustered because their delay was considered too high to be relevant.

This experiment allows to conclude that the fastest routes are not static and time dependent. Another aspect underlined by Fig. 3 and particularly by Fig. 3(d) is the fact that there is no obvious spatial and temporal correlation in routes shape. Because this is not the purpose of the paper and that it does not provide any valuable metric, we do not present results that analyse the resilience time (i.e. lifetime) of the fastest route which is anyway very low.

In summary, we have concluded that the lifetime of the routes and the intermittent character of the links prevent the use of standard mobile network routing such as AODV [18] or OLSR [19] that would allow the topology to be fault tolerant. In a general manner, MANET routing protocols are not designed to perform over such a topology making a DTN approach a good candidate. Following this analysis, in the next section we design a new DTN routing scheme that takes advantage of the topology, comparing it to well-known DTN routing proposals.



(a) (b) (c) (d)

Figure 3: Five best routes groups between two gateways at time t=t_0, $t_0 + 1000$, $t_0 + 2000$, $t_0 + 3000$

3 DQN algorithm

Algorithm 1 DQN Routing Algorithm

function forward
 # A router R have to decide whether or not to forward the bundle
if R == Bundle.source() **then**
  # R is the source of the bundle
  for each Contact C_j of R **do**
    # R checks all its contacts
    if ($Nb_{sentcopies} < L$) AND (alternation between East/West fulfilled) **then**
      bundle is forwarded to C_j and marked with the appropriate direction (EAST or WEST)
      EAST/WEST repartition updated
      $Nb_{sentcopies}$ ++
    end if
  end for
else
  # R is not the source of the bundle
  for each Contact C_j of R **do**
    R requests from C_j its list of contacts $L_{C_j} = \{A_i^j\}$
  end for
  # R has a list of contacts composed by all the contacts list denoted: $\Lambda = \{L_{C_j}\} \cup \{C_j\}$
  R seeks in Λ the closest contact to the destination denoted K where K is closest to the destination than R
  # The distance computation takes into consideration the direction of the bundle
  if K exists **then**
    # An interesting contact can be exploited
    R forwards the bundle to one of its own contact C_j that is in contact with K
  end if
end if

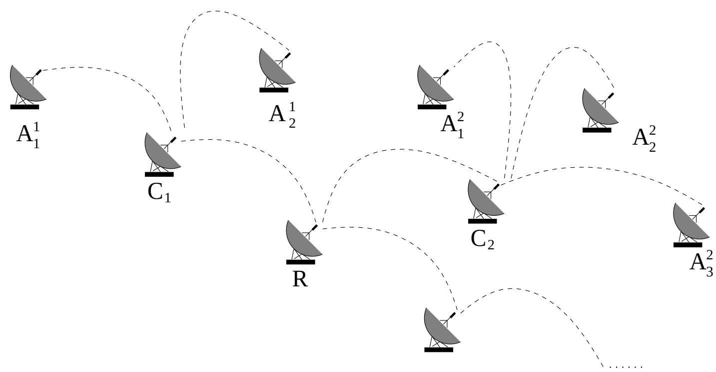


Figure 4: Illustration of the forwarding decision

We rely on the previous charaterisation analysis in Section 2 to design an efficient dynamic routing algorithm without pre-computing. The developed algorithm has to be able to determine every kind of routes, from the shortest to particular routes such as the ones illustrated in Fig. 3(d). Note that we seek to be failure resistant, this means that even if a connection between two gateways fails, data must be delivered over an alternative path at the expense of an increase of the delivery time. We note that each route goes either eastbound or westbound. In the example of Fig. 3(d), the fastest routes are all westbound (note that this property is linked to the Orbcomm constellation topology and can be changed). Taking this property into account, we develop a replication-like algorithm. We recall that end-to-end communications are done in a multi-hop manner from gateways to gateways via a satellite.

The proposed algorithm, DQN, is detailed in Algorithm 1. We wish to point out that some recent work highlights the benefit of the vicinity knowledge in terms of routing over opportunistic networks [20]. In particular, Tiphaine Phe-Neau et al. [21, 22] demonstrate the benefit of knowing neighbor nodes located within k hops from a source and call it $k-$vicinity knowledge. In a certain manner, DQN lays on the same property by limiting the spreading of its number of copies within a number of hops as explained in the following algorithm.

The beginning of this pseudo code (line #3) deals with the case where the gateway R is the source. In this case, the source sprays L copies (also called bundle messages [3]) while trying to achieve a balance between West and East in order to increase the chance to discover one of the five fastest routes. Then, no further replication is done. Otherwise, if R is an intermediate gateway having received one bundle message, it takes a forwarding decision as follows:

1. As illustrated in Fig. 4, R builds a list of reachable gateways (denoted A_i^j in Fig. 4) in two phases. First it determines predictable contacts (denoted C_1 and C_2 in Fig. 4). Then, a signalisation process, consisting in a message exchange, upgrades this list in case of unavailable connexion;

2. after retrieving a set of reachable gateways $\{A_i^j\}$ (where i is the number of the gateway and j the number of the contact C_j which connects this gateway to R), the intermediate gateway decides to forward to one of its one-hop neighbour which is also in contact with the A_i^j gateway the closest to the destination (line #19);

3. if the intermediate gateway does not find any consistent forwarding gateway, it keeps the copy and wait a fixed period in order to obtain a better list of contacts.

4 Performance evaluation

In this section, we give a first evaluation of DQN algorithm. In order to provide a consistent analysis, we evaluate the performance of our proposal against other similar DTN routing algorithms. For all simulations, we used the ONE simulator [23] which is considered as the reference tool for DTN simulations. DQN was implemented in the ONE and all others algorithms were already implemented in this tool.

There are two classes of DTN routing algorithms. Those that only forward copies (e.g. hot-potato routing) and those that replicate a set of infinite or limited copies. As our algorithm is a replication-like algorithm, we compare our proposal to Epidemic, Binary Spray and Wait (BSW) [5], PRoPHET [6] and RAPID [7] algorithms.

Epidemic was chosen to give a lower bound of the delay. This scheme has the biggest overhead in terms of number of disseminated copies but is the fastest in terms of time delivery considering that the buffer of the topology are infinite. We use the Epidemic scheme as the reference scheme for delay. Binary Spray and Wait was chosen because it is a popular and efficient DTN routing algorithm which limits the number of replicates. Despite of being optimal for purely random networks, we may expect poor results from this scheme over a quasi-deterministic topology. In our context, due to the per-hop communication scheme, the relative diversity of each node is low and the average route length is high. Moreover, BSW requires at least a 2^n replication factor to follow a route with n intermediate nodes. As the average route length is high, BSW will need a high replication factor to correctly perform and obtain good delivery ratio performance. For instance, to be able to follow an eight intermediary hops path, the replication factor must be greater than 256, yielding a high overhead, likely not adapted to a satellite constellation.

The advantage of PRoPHET algorithm is to detect particular contact schemes. This makes it a strong candidate to our proposal. However, the performance of this protocol is quite sensitive to its initial parameters α, β, γ [24].

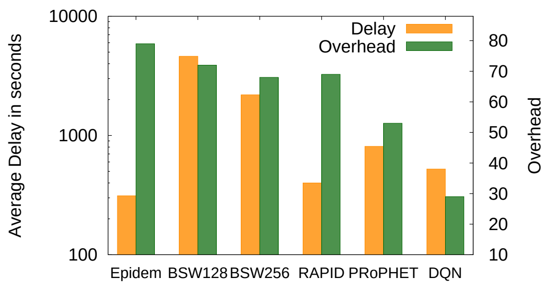
Finally, we choose RAPID which is based on a per-message and a global utility computation, because this scheme is known to obtain satisfying performance in terms of delivery delay.

In order to drive a representative simulation, we select four different pairs of nodes. Each pair corresponds to a given source and destination and generates a traffic of 220 bundles spaced by 1000 seconds. We recall that all gateways are able to receive and send messages, acting as message relays. We fix L to 5 in DQN. Increasing this value increases a bit the average overhead since more copies can be disseminated in the system. It also lowers the average delivery delay since more copies implies more chance for one of them to follow one of the fastest routes. Some experiments, not presented here, confirmed that this value is a good compromise. Results are presented in Fig. 5. For each routing algorithm, we plot the end-to-end delay, defined as the time between the creation of the bundle and the reception of the first replicate by the destination. We also represent the average overhead in terms of number of replicates. The per-bundle overhead, O_b is defined for each bundle b as follows:

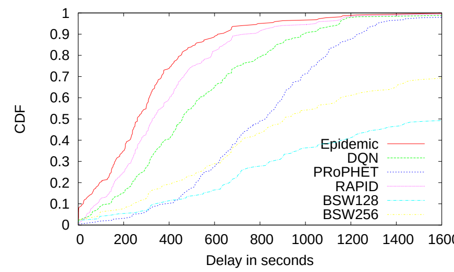
$$O_b = \frac{n_{t,algo} - n_{t,opt}}{n_{t,opt}}$$

with $n_{t,algo}$ the number of bundle transmissions for the selected routing algorithm and $n_{t,opt}$ the number of bundle transmissions in the optimal case. Note that the Epidemic scheme is the best case in terms of delay but the worst in terms of replicates overhead. The direct competitor to our proposal seems to be RAPID. However, the most important result is that in terms of overhead, our proposal obtains the best score compared to all other schemes.

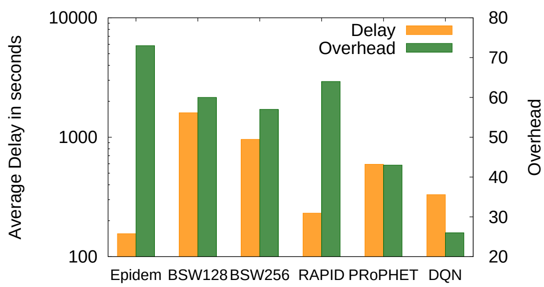
Previous results provide the average delay. However, it is important to also compare these algorithms with respect to their delay repartition. An algorithm might display similar performance than another on average but can present either a very short or very long delay. To drive this comparison and to complement these results, Fig.5 gives the CDF of delays. If we consider a compromise between delay and overhead in terms of number of replicated messages, our proposal displays the best trade-off. It highly decreases the overhead, at least by 35%, which is the most important aspect in a satellite context and obtains a delivery delay only about 20% higher compared to RAPID. Finally and to better assess the performance of our proposal, a percentage gain balance is plotted in Fig. 6. This figure sums up the whole results provided in Fig. 5. A positive value means that our proposition offers better performances than a given scheme (the gain is relative, $-1 \leqslant G \leqslant 1$).



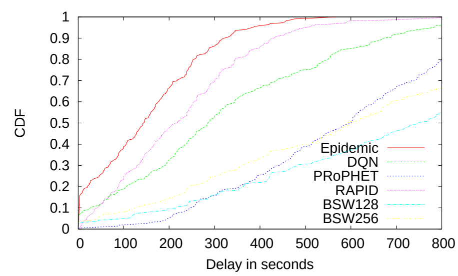
(a) From Ushuaia to Hudson Bay



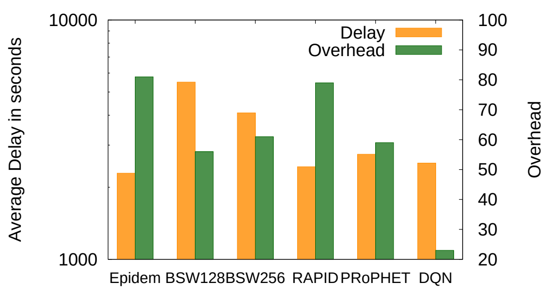
(b) From Ushuaia to Hudson Bay



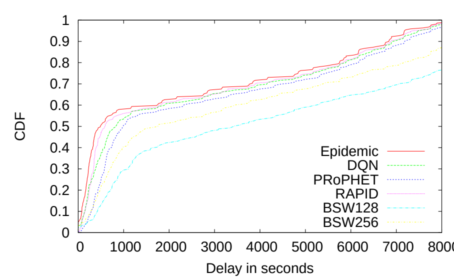
(c) From Santiago to Rome



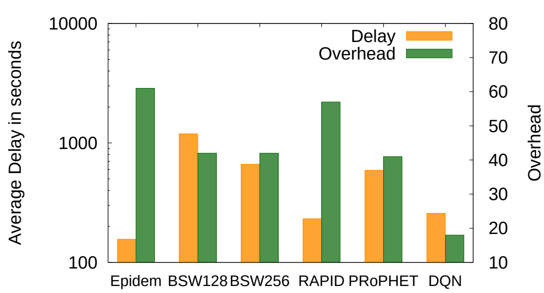
(d) From Santiago to Rome



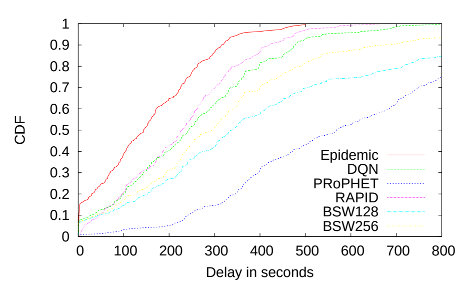
(e) From Leticia to Mokoku



(f) From Leticia to Mokoku



(g) From Istanbul to São Paulo



(h) From Istanbul to São Paulo

Figure 5: Evaluation of delivery delays (average values and distribution functions) and data overhead obtained by DQN and other common DTN routing algorithms.

5 Extensive evaluation of DQN

5.1 Impact of terrestrial gateways density on routing performances

In this section we study the impact of terrestrial gateways (GW) density on delivery delay for DQN against an optimal routing. To keep a general aspect and obtain unbiased results, we still consider a uniform distribution of terrestrial gateways. Our analysis are driven on three different distributions in

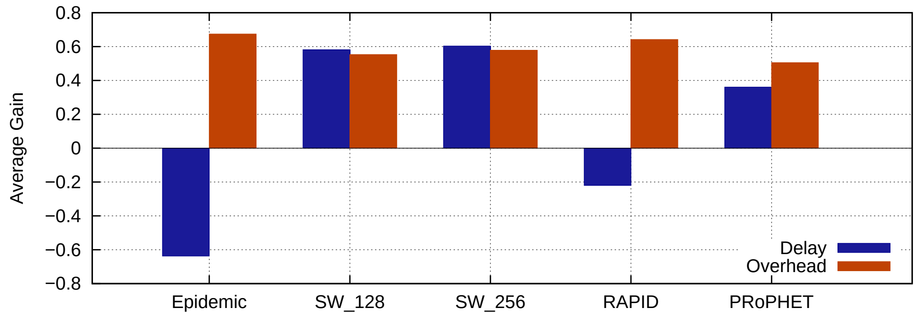
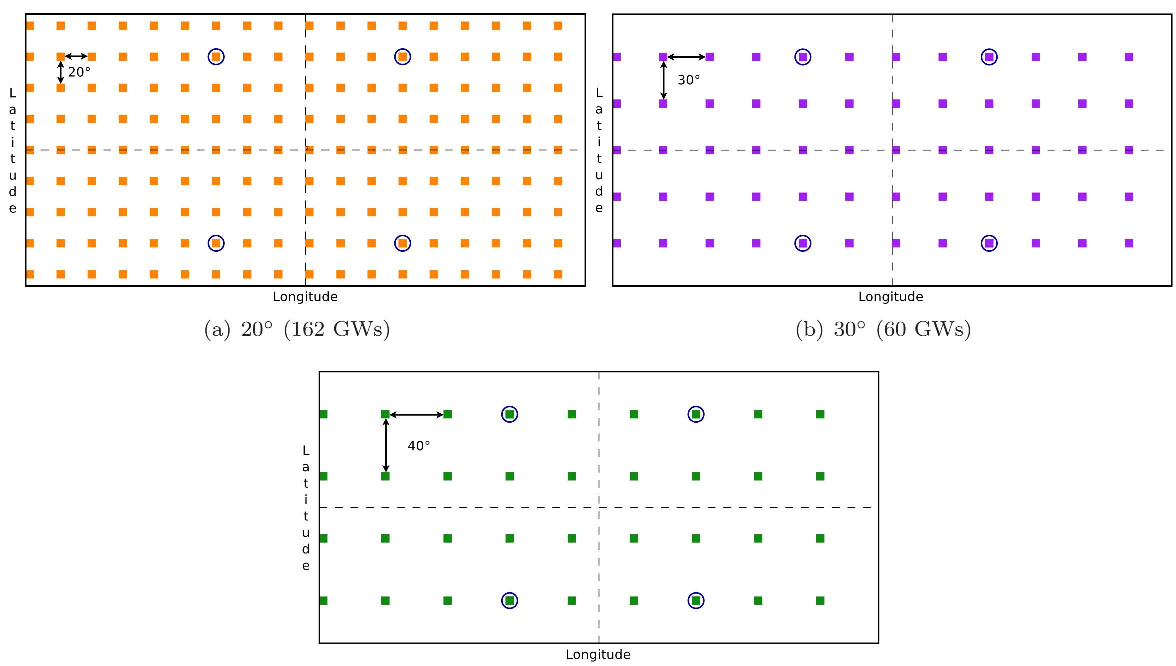


Figure 6: Gains results against DQN

which GW are respectively separated by a corresponding distance between GWs of 20°, 30° and 40° of latitude and longitude. Distributions with a shorter or longer separation distance are not considered since it imply either too many GW to be realistic or too few to ensure a fully connected network.

5.1.1 Three different distributions of gateways

We decide to compare delivery delay performances of optimal routing and DQN on three different distributions. To enable a fair comparison, these three distributions must have several GWs identically positioned. They are presented in Figure 7. We can see on these figures that four Gws are at the same position in all distributions.



(a) 20° (162 GWs)

(b) 30° (60 GWs)

(c) 40° (36 GWs)

Figure 7: Three different homogeneous gateways density

Using these four nodes, we determine 4 couples of source/destination nodes as presented in Figure 8. The first three pairs correspond to connections between node positioned in different hemispheres. The fourth one gives an example of connection between two nodes localized in the same hemisphere. The rotational symmetry of the system allows to limit the testing to these four pairs.

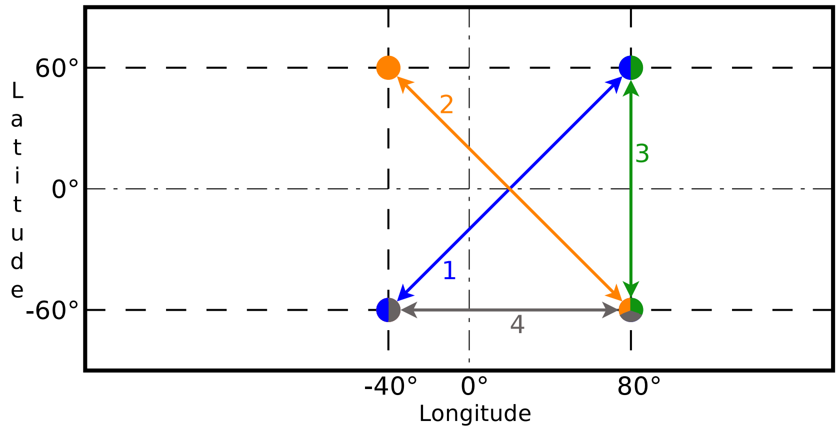


Figure 8: Geographical position of analysed pairs.

5.1.2 Results of average delivery delay as a function of GWs density

As in the previous section, we evaluate and compare the average delivery delay of DQN and an optimal routing for each distribution. Figure 9 presents the results obtained for pairs 1, 2 and 3. For each pair, simulations have been done in both ways, by switching the source and destination.

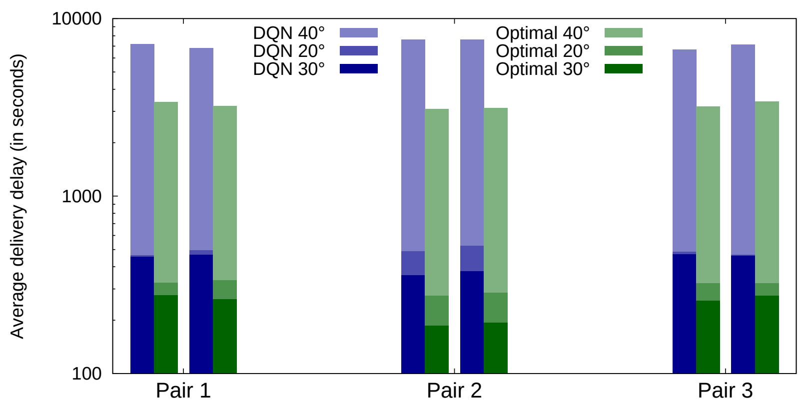


Figure 9: Average delivery delay for the pairs 1, 2 and 3.

First, we note that the delivery delay presents a break between 30° and 40°. This is explained by the distance between GWs in the equatorial area. In fact, the covering area of considered LEO satellites corresponds to a circular area of 44° in this zone. Thus, the connection availability between two nodes of this area is very low. This explains why we observe a strong increase of the delivery delay between two nodes positioned in different hemispheres as data messages have to cross the equatorial area. If we consider a distribution in which GWs are separated by a corresponding distance of 50°, delivering data between two nodes positioned in different hemispheres becomes impossible. The distance threshold is given by the minimal size of satellites covering area. In the case of 40°, the network can be seen as a bipartite network in which connections between its two parts are rare. This break is visible for the three first pairs of nodes. In Figure 10, we only present results for 20° and 30° to ease the reading.

Figure 10 underlines a counter-intuitive result. Indeed, the average delivery delay for all pairs is lower on the 30° distribution than on the 20° distribution. This means that the average delivery delays are better with fewer terrestrial gateways, even in the optimal case. Before explaining the reason for this surprising result in the next Section, it is important to refine the obversation. Let's count the number of cases where delivery delays are better with a 20° distribution than with a 30°. Figure 11 show these results for pair 1. In this experiment, 726 data messages are sent. In half cases, messages

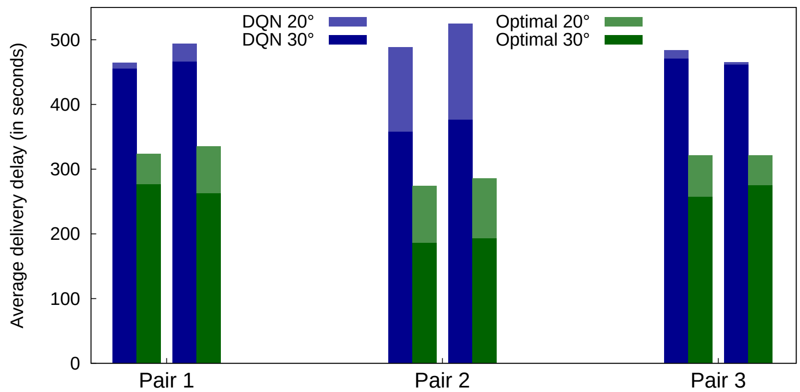


Figure 10: Average delivery delay for 20 and 30 repartitions.

are delivered faster in the 30° distribution. For 25% of cases, delivery delays are identical for both gateway densities. Finally, the 20° distribution gives better results only in the last 25% of the cases despite three times more GWs. The results are similar for the three first pairs.

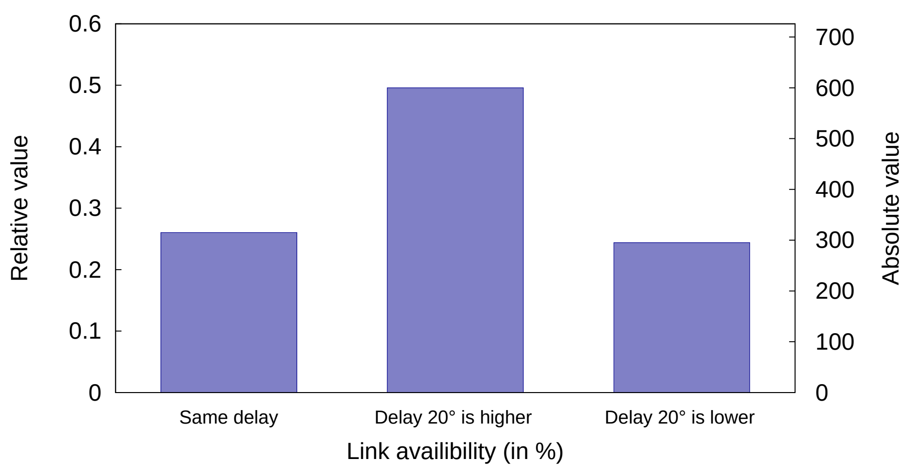


Figure 11: Delivery delays comparison of 726 messages for pair 1.

5.1.3 Analysis of difference between 20° and 30° distribution

To understand this difference and why a larger number of GWs yields better results only in one case over four, let's focus on the routes followed by messages in the optimal case. The fact that 30° is not a multiple of 20° means that routes followed by a message on the 30° distribution does not have necessarily a corresponding route in the 20° distribution. Indeed, in the 20° distribution routes present jumps of 20° or 40° but never 30°. However, the maximum covered distance by a satellite in the equatorial area is about 44°. As a result, a jump of 30° has a higher probability of occurrence than a jump of 40° because 40 is closer to the threshold. That means that in the 20° distribution and in the equatorial area, jumps are often limited to 20° whereas in the same conditions it is often possible to have a 30° jump on the 30° distribution. As a consequence, data messages often move faster in the 30° distribution because they faster cross the equatorial area. This qualitative analysis explain previous observations.

This phenomenon is particularly true in the area where satellite coverage is the thinnest. This means that for two nodes situated in the same hemisphere the delivery delay difference should be smaller. To fully validate our explanation, we focus on the delivery delay of pair 4 formed by two nodes positioned in the same hemisphere. Results are presented in Figure 12.

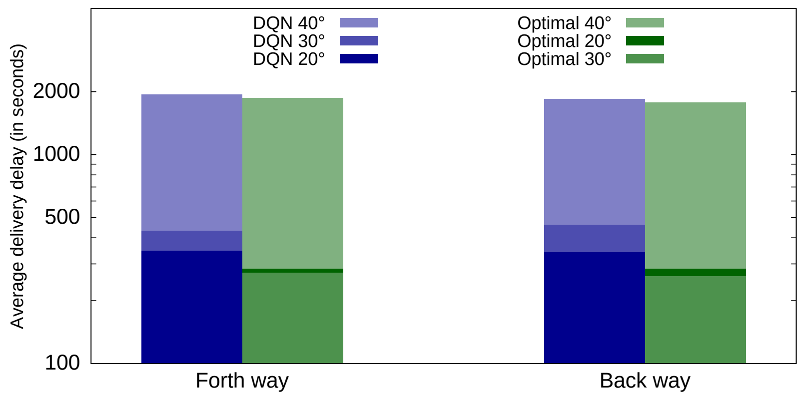


Figure 12: Results for pair 4.

The results shows that the performance difference between the two distribution is quasi null. This confirms our previous explanations and analysis. However, there is still a small difference in the advantage of 30° distribution. In other words, the previous explanation is still valid for area of smaller latitude. The difference is in the appearing frequency of the cited phenomenon. This frequency is maximal for equatorial area and is lower for area of smaller latitude.

This study also demonstrates that it is not necessarily useful to uniformly increase the amount of GWs. To optimize delivery delay, it would be more efficient to have a higher density of GWs in the equatorial area and a lower one in the proximity of poles. However, it is very complicated to clearly model this problem and to obtain an optimal function.

5.2 Experiment in a realistic scenario

We propose in this section to evaluate and compare our algorithm over a realistic scenario. Thus, we define a topology of 33 gateways placed near major towns. Figure 13 presents their localization. The main characteristic of this topology is to present isolated gateways as the one in Sydney, Australia for example. We propose to further optimize the message delivery by analysing the new contacts distribution resulting from this sparse repartition and to enable a contact learning phase in order to discover these isolated gateways. The objective is to prevent spurious replications in areas where the destination would never be reachable.

Figure 14 presents the results obtained and shows that our algorithm outperforms RAPID in this context. The overhead observed is low while the delay is in the worst case, 35% higher than the optimal delay. This experiment is a simple example that will be further studied in a future work.

5.3 Evaluation of DQN on a loaded network

5.3.1 Load of the network

Previous evaluations and comparisons of DQN have been done on unloaded networks. DQN was designed to handle unexpected disconnections. This ability is a real advantage in comparison of pre-computed routing algorithms. Indeed, DQN does not directly use the network determinism and follows a replication scheme that enables reliability in real conditions or in case of failures. A precomputed routing algorithm bases all decisions on network determinism. In case of connection failures, packets might get directed to a dead end and performance will be impaired

Connection unavailability can be caused by failures or severe load increase. On the one hand, there are rare hardware failures and link budget impairments due to weather conditions (e.g. strong rain [25]). On the other hand, unavailability can be a consequence of an increasing load in the network.

Figure 13: Topology used and source/destination localization

In fact, a contact between two nodes has a limited duration and so is the amount of data that can be exchanged. Thus, in a loaded network, a message can be stored longer since it cannot exploit all contacts. We can express the exchangeable amount of data C_c as follows:

$$C_c = D_c.T_h \tag{1}$$

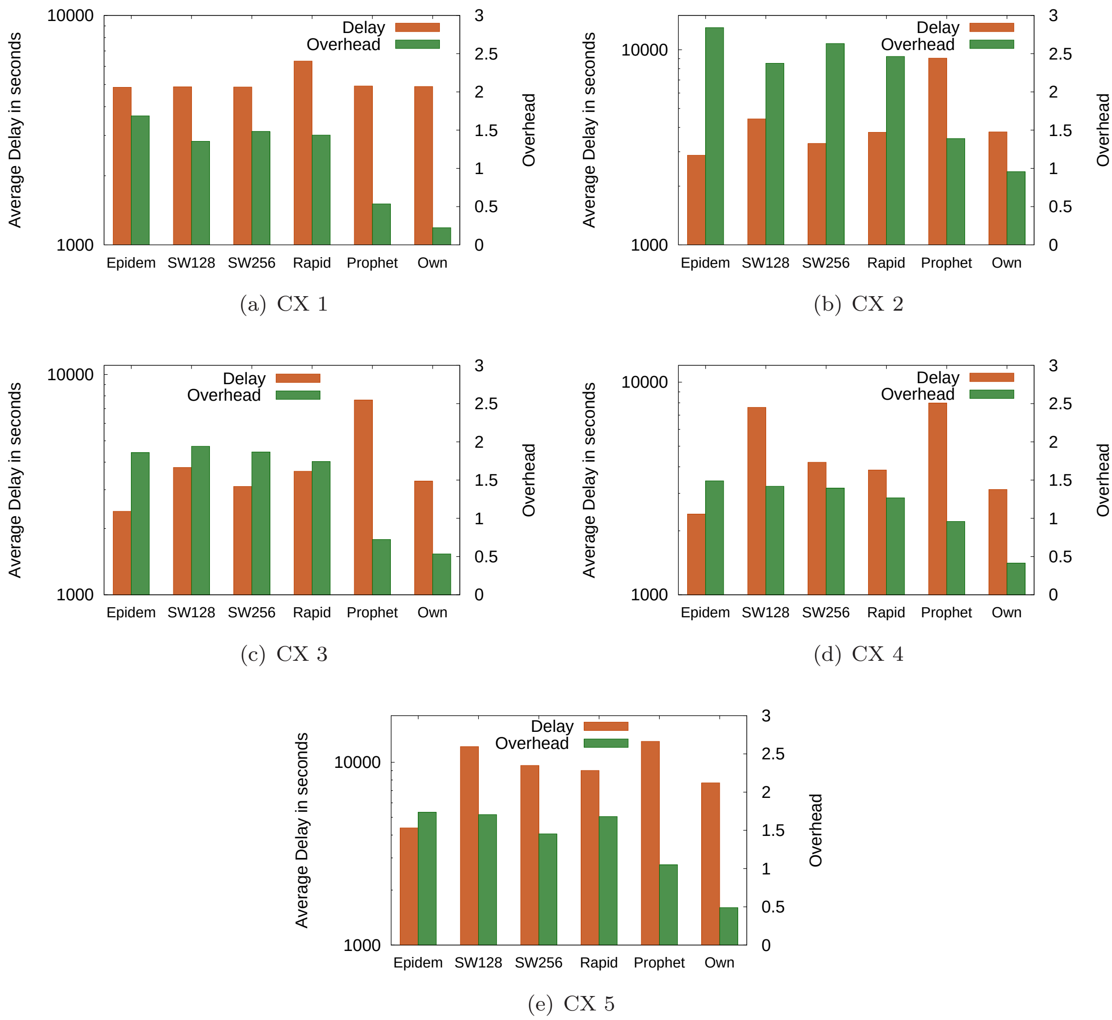
with D_c the contact duration between the two considered nodes and T_h the throughput.

Additional buffering delay in GWs is directly correlated to network load and traffic on the satellite uplinks. Generally, traffic distribution is hard to predict and an important source of indeterminism. In our system, up-downlink throughput is limited (some kB/s) and the amount of data messages to circulate is potentially important. A static pre-computed routing would suffer from this aspect. Introducing traffic dynamics in a pre-computed routing algorithm requires signaling traffic so to adapt routing decision. Considering the specifics of the system considered, we believe that this very aspect of dynamic routing is complex to implement. As a consequence, pre-computed algorithms performance should be quite degraded over a loaded network.

5.3.2 How to simulate network load in a network

To simulate network load, a first solution consists in generating an important number of data flows in the whole network to form a background traffic. Thus, we can focus on a particular flow and extract additional buffering delays or missed connections for example. However, to ensure a general aspect, the background traffic needs to be randomly built and the experiments must be repeated several times. This method requires several time-consuming simulations partly dependant on the amount of data messages exchanged. As a consequence, the simulation time needed to develop this method is not tractable.

A consequence of network load is that data messages cannot use all connections between nodes. To simulate this load, we choose to randomly remove connections from the initial trace. Experiments are done on a degraded connections trace directly extracted form the original connections trace. This is a first method to simulate network load consequences. The obvious advantage is to greatly decrease the simulation time. However, it is impossible to clearly express the relationship between a given network load and the probability of disabling a given connection. Such a function exists but cannot be literally determined. Section 5.4 proposes an alternate scheme which considers partial disabling of connections.



(a) CX 1 (b) CX 2 (c) CX 3 (d) CX 4 (e) CX 5

Figure 14: Delay and overhead distribution, each connexion (CX) is represented in Fig. 13

5.3.3 Simulation assumptions and considered routing protocols

In our context, no congestion information is exchanged in the network. Pre-computed routing algorithms cannot rely on this metric to optimize routing decisions. To build degraded connexions traces, we apply a connection loss probability on the original trace corresponding to a unavailability rate of contacts. We take different seeds to generate several degraded traces with the same connection loss probability. Then, simulations are played in the same way as previously. We choose a low message generation frequency to prevent additional and uncontrolled load. We evaluate DQN delivery delay performance as a function of unavailability rate and compare it with the performance of an Oracle and a basic version of pre-computed algorithm.

Oracle

The Oracle corresponds to an omniscient protocol. It fully knows current and future network conditions. Thanks to this complete knowledge, it can take optimal routing decisions. As a result, this routing algorithm does not suffer from the indeterministic aspect of traffic load. This type of protocol has no real existence, however it indicates a lower bound on the delivery delay, comparable to epidemic routing.

Basic precomputed routing algorithm

Pre-computed routing protocols rely on a periodic update of routing tables. This computation is done without any consideration of unavailability and is applied by relays for a given period. During this period, if some connections are not usable, data messages can be blocked and will not take the optimal routes. If routing tables updates happen with in the right timing, the message will be blocked for a short while but can follow a different route theoretically optimal. If not, the message will follow the initial route and will suffer from additional buffering delays. In our system, we know that optimal routes localization is strongly time dependant. Thus, if routing tables are updated during the progress of a message, the risk is to direct it to the wrong region and to miss the optimal routes. As a consequence, the message risks to do some zigzags or loops and will highly increase its delivery delay. Even if a computation, based on expected future availability, is done at every hop, there is no guarantee that the followed route will be the actual optimal one. Indeed, indeterminism makes that the concatenation of presupposed optimal decision cannot be guaranteed to be optimal. This aspect fully makes sense in our system since we consider the unpredictable unavailability brought by the indeterministic character of the traffic. This point is the central topic of a future study in which we will try to highlight these phenomenons and the limit of precomputed algorithm application in uncontrolled systems.

Links unavailability can have a strong impact on such routing algorithm.

To prevent this kind of phenomenons that are difficult to quantify, we choose to focus on a basic precomputed routing. We propose that a message simply follows the initial computed route. This route corresponds to the optimal one without connection failure. The performance of this routing protocol will be degraded with the increase of unavailability.

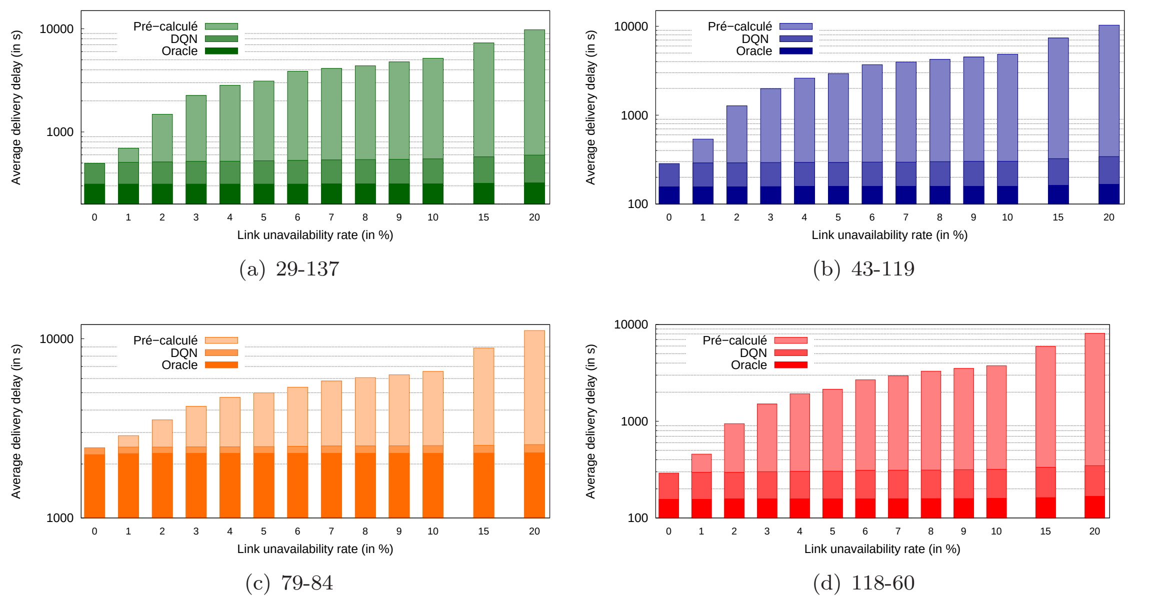
5.3.4 Results

The evaluation of DQN performance in case of loaded networks is done with the same four pairs source/destination considered in the unloaded case. We sum up delivery delay results obtained for DQN, Oracle and basic precomputed routing algorithm in Figure 15. Results obtained for an unavailability of 0% correspond to results obtained in an unloaded network. We can notice on this figure that the lower bound of delivery delay which is given by the oracle results, increases almost linearly until 10% of unavailability. Results for DQN increase in the same order of magnitude. Thus, the relative difference between DQN and Oracle seems to remain constant. This means that DQN is robust and handles well connection failures. Delay for pre-computed routing rapidly increases. This result confirms the high impact of unavailability on pre-computed routing.

5.4 Refinement of load simulation

First simulations are done considering that the load induces a random connection unavailability. Thus, for a given message, only a subset of the initial connections can be used. This means that connection failure probability is the same and does not depend on connection length. To cope with this issue, we propose to consider the connections duration. Indeed, a longer connection is more likely to be usable by a given message since its transmission capacity is higher. Thus, instead of randomly disabling some connections, we shorten them by a fixed amount of time. If the shortening delay is higher than the connection duration, the connection is disabled. This modification translates the increasing probability of disabling a connection as a function of its length. We consider different shortening delays that correspond to different load. Practically, we disable too short connections and shorten the others to obtain new simulation traces. Simulations are played exactly in the same way as in the previous case. We present results obtain by DQN and the Oracle in Figure 16. Shifting delays evolve from $5s$ to $100s$.

Dotted lines in Figure 16 represent the ratio between delivery delays obtained by DQN and Oracle. We can note that this ratio is either stable or decreasing. Indeed, for three pairs over four, this ratio is decreasing. Thanks to its replication algorithm, DQN performance is closer to the optimal performance when network loaded increases.



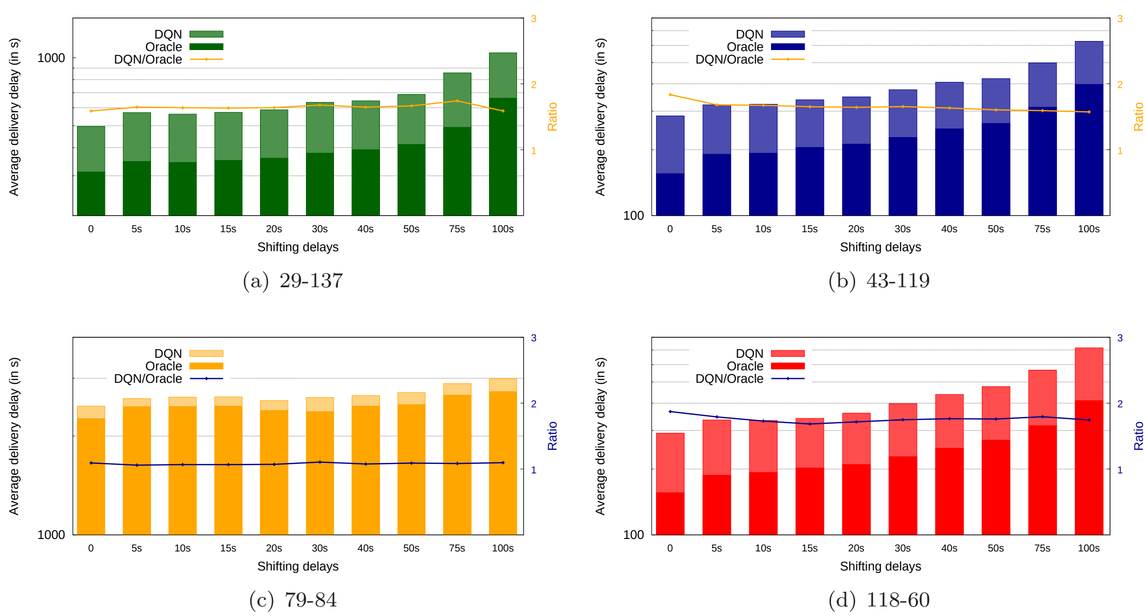
(a) 29-137

(b) 43-119

(c) 79-84

(d) 118-60

Figure 15: Comparison of *DQN*, pre-computed and Oracle algorithms as a function of links unavailability(Average results for 10 simulations).



(a) 29-137

(b) 43-119

(c) 79-84

(d) 118-60

Figure 16: Comparison of *DQN* and Oracle algorithms as a function of shifting delays (Average results for 10 simulations).

6 Conclusion

We propose in this paper a novel DTN routing algorithm specifically designed for quasi-deterministic networks. The main objective is to propose an alternative to pre-computed routing scheme commonly used over satellite constellations. We have tested our solution against well-known DTN routing protocols over a topology similar to the Orbcomm constellation. Unlike the Orbcomm system, the satellites we consider have no storage capacity to keep a low cost topology. We show that our proposal outperforms in terms of replication overhead most of current DTN routing scheme without too much delay compromise. Additional evaluations shows that our solution is a potential candidate to enable fault-tolerant routing over satellite constellation. Indeed, we demonstrate that DQN has good performance in realistic contexts: with an heterogeneous distribution of GWs or over a loaded network. In this last context, simulations underline the robustness of DQN. In fact, its results remain the same with the increase of network load. In a future work, we expect to deeper investigate this scheme in several other scenarios and to investigate further optimizations in the case of isolated gateways.

References

[1] Pratt S, Raines R, Fossa C, Temple M. An operational and performance overview of the iridium low earth orbit satellite system. *Communications Surveys Tutorials, IEEE* 1999; **2**(2):2–10.

[2] Cruz-Sanchez H, Franck L, Beylot AL. Precomputed routing in a store and forward satellite constellation. *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, 2007; 240–243.

[3] et al VC. Interplanetary internet (IPN): Architectural definition 2001. Http://www.ipnsig.org/reports/memo-ipnrg-arch-00.pdf.

[4] Crowcroft J, Yoneki E, Hui P, Henderson T. Promoting tolerance for delay tolerant network research. *SIGCOMM Comput. Commun. Rev.* Sep 2008; **38**(5):63–68.

[5] Spyropoulos T, Psounis K, Raghavendra CS. Spray and wait: An efficient routing scheme for intermittently connected mobile networks 2005; :252–259.

[6] Pelusi L, Passarella A, Conti M. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE* Nov 2006; **44**(11):134–141.

[7] Balasubramanian A, Levine B, Venkataramani A. DTN routing as a ressource allocation problem. *SIGCOMM Comput. Commun. Rev.* Aug 2007; **37**(4):373–384.

[8] Hui P, Crowcroft J, Yoneki E. Bubble rap: Social-based forwarding in delay-tolerant networks. 2011; 1576–1589.

[9] Caini C, Cruickshank H, Farrell S, Marchese M. Delay- and disruption-tolerant networking (dtn): An alternative solution for future satellite networking applications. *Proceedings of the IEEE* Nov 2011; **99**(11):1980–1997.

[10] Leighton T, Rao S. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM* Nov 1999; **46**(6):787–832.

[11] Ford L, Fulkerson D. *Maximal Flow Through a Network*. P (Rand Corporation), Rand Corporation, 1954. URL `https://books.google.fr/books?id=TkiWtgAACAAJ`.

[12] Hay D, Giaccone P. Optimal routing and scheduling for deterministic delay tolerant networks. *Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Conference on*, 2009; 27–34.

[13] Burleigh S. Contact graph routing. *Technical Report draft-burleigh-dtnrg-cgr-01*, IETF 2010. (Work in progress).

[14] Caini C, Firrincieli R. Application of contact graph routing to LEO satellite DTN communications. *Proceedings of IEEE International Conference on Communications, ICC 2012, Ottawa, ON, Canada, June 10-15, 2012*, 2012; 3301–3305.

[15] Taleb T, Kato N, Nemoto Y. Recent trends in ip/ngeo satellite communication systems: transport, routing, and mobility management concerns. *Wireless Communications, IEEE* Oct 2005; **12**(5):63–69.

[16] 2009. Orbcomm website, http://www.orbcomm.org/.

[17] Diana R, Lochin E, Franck L, Baudoin C, Dubois E, Gelard P. A dtn routing scheme for quasi-deterministic networks with application to leo satellites topology. *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, 2012; 1–5.

[18] Perkins C, Royer E. Ad-hoc on-demand distance vector routing. *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, 1999; 90–100.

[19] Clausen T, Jacquet P. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental) Oct 2003. URL `http://www.ietf.org/rfc/rfc3626.txt`.

[20] Phe-Neau T, Dias de Amorim M, Conan V. Vicinity-based dtn characterization. *Proceedings of the Third ACM International Workshop on Mobile Opportunistic Networks*, MobiOpp '12, ACM: New York, NY, USA, 2012; 37–44.

[21] Phe-Neau T, de Amorim M, Conan V. The strength of vicinity annexation in opportunistic networking. *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, 2013; 217–222.

[22] Phe-Neau T, Dias de Amorim M, Campista MEM, Conan V. Examining vicinity dynamics in opportunistic networks. *Proceedings of the 8th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, PM2HW2N '13, ACM: New York, NY, USA, 2013; 153–160.

[23] Keränen A, Ott J, Kärkkäinen T. The one simulator for dtn protocol evaluation. *Proceedings of the 2Nd International Conference on Simulation Tools and Techniques*, Simutools '09, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering): ICST, Brussels, Belgium, Belgium, 2009; 55:1–55:10.

[24] Lindgren A, Doria A, Schelén O. Probabilistic routing in intermittently connected networks. ACM: New York, NY, USA, 2003; 19–20.

[25] Castro M, Fernandez D. Voip cross-layer load control for hybrid satellite-wimax networks. *Wireless Communications, IEEE* Jun 2008; **15**(3):32–39.