Contact Graph Routing Enhancements for Delay Tolerant Space Communications

N. Bezirgiannidis*, C. Caini°, D. Padalino°, M. Ruggieri°, V. Tsaoussidis*
°DEI/ARCES, University of Bologna, Italy
*Space Internetworking Center, Democritus University of Thrace, Greece
ccaini@arces.unibo.it, {dmitrij.padalino, marco.ruggieri3}@studio.unibo.it {nbezirgi, vtsaousi} @ ee.duth.gr

Abstract— **In this paper we present two enhancements to Contact Graph Routing (CGR), a Delay-/ Disruption- Tolerant Networking routing algorithm developed by NASA JPL for space environments with predetermined connectivity schedules, such as Interplanetary communications and LEO satellite systems. The first enhancement, CGR-ETO, aims to improve the accuracy of predicted bundle delivery time by considering the available information on queueing delay. The second, the Overbooking Management, aims to proactively handle contact oversubscription, which occurs when high priority bundles are forwarded for transmission on a contact that is already fully subscribed by lower priority bundles. Both enhancements have been inserted as optional features in the Interplanetary Overlay Network CGR implementation in order to comparatively evaluate their performance on a GNU/Linux testbed running the full protocol stack. The results show that the two enhancements are complementary and can significantly improve routing decisions compared to standard CGR.**

Keywords-component; Delay-/Disruption- Tolerant Networking, CGR, Satellite communications, Space Communications, Performance evaluation.

I. INTRODUCTION

Delay-/Disruption- Tolerant Networking (DTN) emerged as a potential solution to cope with the special nature of space communications, including satellites [1], [2], [3], [4]. Its ability to handle long delays, communication disruptions, high error rates, asymmetric link rates, and lack of end-to-end connectivity has provided the community with an ideal network architecture to build the future Solar System Internet [5]. Since its first appearance, DTN has evolved and various protocols and services have been added, many of which are in the process of standardization by the Consultative Committee for Space Data Systems (CCSDS). In parallel, numerous studies have been presented to evaluate its effectiveness for challenged terrestrial networks (e.g., mobile ad-hoc networks, sensor networks, tactical networks, underwater communications, etc.) [6].

Due to the possible lack of continuous end-to-end connectivity, routing in DTN is a very complex problem and one of the most challenging research issues. In contrast to terrestrial DTNs, where transmission opportunities or "contacts" between nodes typically follow an opportunistic pattern, in space contacts are essentially deterministic. These transmission opportunities are possible when DTN nodes (space assets and earth ground stations) are in line-of-sight, and these periods can easily be predicted as all nodes follow deterministic motion. By using the "contact plans", space mission engineers prepare configurations for every mission, exploiting the communication opportunities between each pair of nodes. In this context, Contact Graph Routing algorithm (CGR) has been proposed to construct graphs dynamically using the contact plan and to take routing decisions accordingly [7]. CGR is part of the Interplanetary Overlay Network, the NASA JPL DTN implementation [8] and was tested in a deep-space mission, in the Deep Impact Network Experiment [9].

Ever since CGR first appeared, the research community has worked on improving its functionality and usage. In [10] the authors proposed Enhanced CGR, using path selection with Dijkstra's algorithm, and they have also inserted the new algorithm in ION-DTN implementation. In [11] the authors proposed the use of source routing, and suggested that information extracted at the source node be stored in a Bundle Extension Block [3]. The impact of queueing delay on routing was first examined in [12], where some of the present authors proposed CGR-ETO, a modification of CGR that incorporates available queue backlog information in routing decisions. Simulations showed improvements in both routing and estimation of end-to-end delivery delay. Stimulated by these encouraging results, in the present work we move one step forward and, after implementing CGR-ETO within ION code, we evaluate its performance in an emulated scenario, where nodes use real implementations of the full protocol stack.

An additional problem, identified in [13], is the present late handling of contact "overbooking". Because of their finite length and transmission rate, contacts have a finite volume or "capacity". CGR checks the availability of enough residual capacity before forwarding a bundle (i.e., PDU of the Bundle Protocol [3]) to a contact. However, when it forwards a higher priority bundle it deliberately neglects lower priority ones in order to enforce priority. This can cause oversubscription of a contact, if the contact is already fully booked by lower priority bundles, a situation like seat overbooking on flights. Here we propose an additional modification of CGR, called Overbooking Management, which minimizes the consequences of overbooking by early handling. This enhancement has been implemented in ION and is presented and evaluated here for the first time.

To assess the validity of the enhancements proposed, we consider a simple network topology, which can represent both a LEO satellite and a deep space communications scenario, and we carry out a series of experiments on our DTN testbed based on GNU/Linux machines. The results highlight the benefits provided by the use of both; the proposals are complementary and their combined use can greatly improve CGR routing decisions in a wide variety of cases.

The remainder of the paper is organized as follows: In Section II we briefly present the CGR algorithm and in Sections III and IV we describe our enhancements, CGR-ETO and Overbooking Management. In Section V we present the emulation scenario and in Section VI we evaluate our proposals performance. Finally, conclusions are drawn in Section VII.

II. ROUTING ON INTERPLANETARY INTERNET

Space environments are characterized by intermittent scheduled connectivity. Due to the motion of space assets and planets, communication between DTN nodes is possible for only limited intervals of time, called "contacts". In turn, each contact offers the opportunity to transfer only a limited amount of data. Its maximum is the "contact volume" (or contact capacity) and is given by the product of the transmission rate and the contact duration. The connectivity is essentially deterministic, and minor random effects (due for example to failures) can be conveniently tackled by recovery mechanisms.

A. CGR Description

CGR is a dynamic routing algorithm designed by NASA-JPL for DTN networks characterized by intermittent, scheduled connectivity [7]. Therefore, it can be successfully applied not only to an Interplanetary Internet, but also to LEO satellite communications, as in both cases link availability is known a priori. In brief, CGR exploits the information about contacts contained in a "contact plan" and tries to find the most suitable path from source to destination, based on some routing metric (typically earliest delivery time). CGR is a composite algorithm, because dealing with scheduled intermittent links, instead of continuous links, is intrinsically complex. For an exhaustive explanation we refer to the CGR section of the ION manual [8]. Here we provide some key points of its functionality, necessary for a better understanding of our enhancements.

- Each node uses the contact plan information to build a "contact graph" and then a routing table with a list of plausible routes for each possible destination.
- Each route consists of the complete path to destination. More precisely, each route is the collection of contacts that can be used to reach the destination. Note that several series of Dijkstra searches are required to cope with link intermittency. Routes must be recomputed if there are changes in the contact plan.
- For each bundle, CGR checks the available routes and chooses the best. However, once route is selected, CGR uses only the information related to the first contact of the route, the final results being: 1) the proximate node to forward the bundle to, and 2) the "forfeit time", by which the bundle must have been sent to this node.
- CGR results are used only locally. The best route is recomputed at each node implementing CGR, through the path to destination.
- The criteria for optimal route selection may vary. In practice, in the latest ION releases the utilized routing objective is the shortest "expected delivery time".
- CGR takes bundle priorities into full account.
- If a bundle is not transmitted before the annotated "forfeit" time, it is reforwarded. This is a backoff mechanism to cope with any sort of link impairments.
- In the route selection process, CGR takes account of data already scheduled for transmission to proximate nodes (like seats booked on a flight) by carrying out a "residual volume" check on the first contact of every candidate route. If the bundle's estimated capacity consumption is larger than this residual volume, the route is skipped.

III. FISRT CGR ENHANCEMENT: ETO

In route computation, CGR assumes that bundles can be sent at the start of the contact, or, if the contact is already open, immediately. In other words, it does not consider queueing delay, i.e., the time necessary to transmit bundles already in the transmission buffer. The ETO enhancement aims to use queue knowledge to provide a better estimate of the actual transmission time of the bundle. Note that while it may be difficult to obtain information on the queue for other nodes, this is easy for the local node.

CGR with Earliest Transmission Opportunity (CGR-ETO) was introduced in [12], along with the Contact Plan Update Protocol (CPUP), which disseminates available local queue information to other network nodes. In this paper we move one step forward: we implement the algorithm in ION and evaluate its deployment in our testbed.

The Earliest Transmission Opportunity parameter represents the expected queuing delay, with different values for each level of priority. Thus, based on BP priorities (bulk, standard, expedited) [3] and on ION implementation [8], the local node's contact plan stores different ETO values for each contact: ETO for bulk priority, ETO for standard priority, and 255 values of ETO for expedited priority, based on the 255 ordinal extended class-of-service levels [15] present in ION. The default and starting value of ETO, for all priorities, is the contact start time. Whenever a bundle is queued for transmission in a local outduct to a neighboring node, CGR-ETO (in contrast to CGR) extracts the contact during which the bundle is expected to be transmitted, as additional information. It calculates the Estimated Capacity Consumption ("eccc" [7], [8]) of the bundle, which includes the BP header and an estimate of the underlying protocols overhead, and converts this capacity into transmission time, dividing it by the link transmission rate. The estimated transmission time for this bundle is then added as queueing delay to either the previously estimated ETO of the contact, or the current time, whichever is later. ETO update occurs for all priorities equal to or lower than the specific bundle's priority.

Available ETO information is then used in the CGR-ETO contact plan traversal to decide on optimal bundle routing. Start time, used in CGR, is replaced by ETO of the bundle's priority. This also provides CGR with a more accurate check if the transmission of a bundle can be completed before the end of a contact.

In ION implementation, calculation of Dijkstra is typically done only when the contact plan changes. In principle, the addition of queueing delay in CGR-ETO should modify the contact plan every time a bundle is forwarded, thus imposing extra calculation costs. To avoid this computational overhead, we have inserted a contact plan update threshold, expressed as a percentage of the contact duration. For example, if contact duration is 1000s and the update threshold is set to 10%, recalculation of optimal routes is not triggered until ETO has increased more than 100s since the previous calculation. For this recalculation check we use the bulk priority ETO, since it is the lowest priority and is updated for every forwarded bundle.

IV. Second CGR Enhancement: "Overbooking Management"

During the "residual volume" computation, candidate routes with an earlier delivery time are not discarded when "fully booked" by lower priority bundles. In other words, recalling the analogy with airlines tickets, contact "overbooking" (or more formally, "oversubscription") is allowed in CGR. Our proposal aims to improve the way overbooking is handled by CGR.

For the sake of simplicity we consider the case of a future contact, and continue with the airline analogy. As a result of overbooking, some low priority bundles forwarded to the neighboring node x (i.e. put in the queue to node x) "miss" their contact, because some higher priority bundles have taken their "seats". Once the forfeit time of these bundles expires (typically at contact end-time), these bundles are reforwarded by CGR. This, although robust, is suboptimal, because overbooking is handled only a posteriori, when the bundles have already missed their contact. In order to improve CGR, we suggest that it be handled a priori, by reforwarding the bundles that will miss the contact, as soon as possible, *i.e.* immediately after forwarding the higher priority bundle that has caused the overbooking.

A. Overbooking Check and Handling

Our enhancement consists of two distinct logical phases, namely overbooking check and overbooking handling. The former can be easily performed by introducing computation of the "total residual capacity" (see Figure 1), i.e. the residual backlog volume of all bundles regardless of priority ("totalBacklog"). Note that the "residual capacity" calculation in CGR considers only bundles of the same or of higher priority ("backlog"). Upon forwarding a bundle, overbooking occurs only if i) the residual capacity is higher than bundle dimension, so the bundle can be sent during the contact; and ii) the total residual capacity is not, which means that the bundle can actually be sent only due to its priority. Note that ii) implies that bulk bundles (i.e., of the lowest priority) can never cause overbooking. In the case of overbooking, the handling procedure immediately reforwards the lower priority bundles that can no longer be transmitted during this contact. For a more detailed explanation we have to distinguish between partial and full overbooking, depending on the amount of total residual capacity.

1) Partial Overbooking

If the total residual capacity is positive but smaller than the incoming bundle Estimated Capacity Consumption ("eccc") we have a "partial" overbooking (as in Figure 1). In response to partial overbooking the handling immediately reforwards as many bundles as necessary to empty a contact capacity greater than or equal to the "overbooking" length, which is defined as *overbooking = eccc – totalResidualCapacity*. The overbooking handling mechanism starts the reforwarding procedure from the last bundle of the lowest priority ("bulk") queue. Note that multiple bundles can be reforwarded if their dimension is lower than the overbooking length.

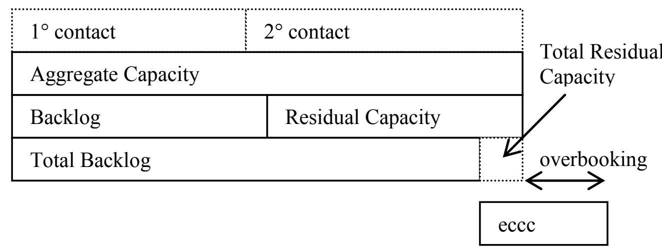

Figure 1: Scheme of a partial overbooking case. The route under consideration starts with the 2nd contact, whose total residual capacity is positive but lower than the bundle dimension (eccc).

2) Total Overbooking

If the total residual capacity is negative (i.e. the contact volume is already fully subscribed), we have "total" overbooking, with *overbooking = eccc*. Unlike the previous case, reforwarding does not start from the last bundle of the queue, but from the last bundle scheduled for the considered contact. In fact, the last bundles of the queue have actually been scheduled for next contacts towards the same proximate node, since the 2nd contact was already fully booked by bundles of the same or higher priority. These bundles are not reforwarded by our overbooking handling mechanism, in order to avoid a cascade effect. Referring to our analogy once again, we "reprotect" passengers of the same flight, not passengers scheduled on successive flights to the same destination.

V. Scenario

In order to evaluate the proposed enhancements we consider the simple four-node topology depicted in Figure 2. Node 1 (the "ipn" scheme [8] uses numbers to identify DTN nodes) represents a space asset, such as a LEO satellite, and is the data source. Node 4 is the Mission Operations Center (MOC) and is the data sink, while nodes 2 and 3 are two terrestrial gateway stations, acting as data relays. In a satellite scenario the first relay could be a terrestrial ground station, while the second could be the control centre of a GEO constellation acting as relay for the LEO sat. The same topology could apply to space communications as well. In our contact plan, space links are intermittent whereas terrestrial links are continuous. On the former we use LTP [14] and TCP [16] is used on the latter. Propagation delays and losses are considered negligible on all the links, as they are irrelevant to routing. The default link characteristics as well as the contact plan information are depicted in Table I. Different settings will be given when used. For convenience, in Table I, the contact volume is also expressed in bundles, considering a bundle payload of 100kB, as used in our experiments (eccc = 107235B, including overhead).

TABLE I. Characteristics of the links (default values).

Link	*Contact#*	*Start-stop time (s)*	*Tx rate*	*Contact Volume*
1-2	1	60-80	512kbit/s	1.28 MB (11.9 bundles)
1-3	1	30-90	128kbit/s	960 kB (8.9 bundles)
1-3	2	105-135	128kbit/s	480 kB (4.4 bundles)
3-4 & 2-4	Dummy (cont.)	1-200	10Mbit/s	

Upon data generation in Node 1, the task of CGR is to find the optimal path to 4 in the presence of intermittent links. Note that because of this intermittency, the best route may vary from bundle to bundle.

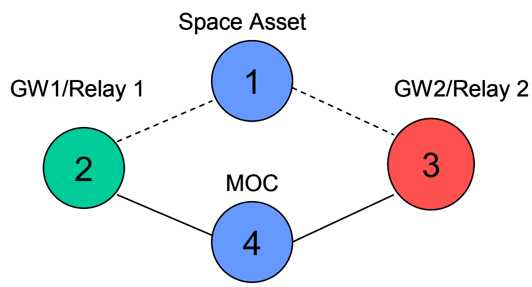

Figure 2: Topology of the scenario considered. Dotted lines: space intermittent links with LTP; terrestrial continuous lines: continuous links with TCP.

VI. EXPERIMENTS

All tests are carried out on our testbed [17] consisting of four GNU/Linux machines running the latest version of ION (3.2.0), reproducing the layout depicted in Figure 2. The experiments, presented in order of ascending complexity, will gradually clarify the two proposed enhancements and highlight their impact on routing decisions. Analysis is made bundle by bundle (micro-analysis).

A. *Parallel equivalent routes; delivery time and load balancing in standard CGR and CGR-ETO.*

We start by considering the deliberately extreme case of two equivalent parallel routes, via 2 and 3. Thus, by contrast to the default case, here we assume two equivalent contacts (length = 110s, Tx rate = 128 kbit/s, contact volume = 1.76 MB, equivalent to 16.5 bundles). The contact to 2 starts just 1s before the contact to 3, at 29 s. Although clearly unrealistic, this case allows us to point out the improvements introduced by ETO. Node 1 generates 16 bundles of 100kB each, all of the same priority (bulk, but this not influential here). In Figure 3 we depict the routing decisions of the standard CGR. The routing algorithm forwards these bundles as soon as they are generated. As standard CGR does not consider the queueing delay caused by the previously forwarded bundles, and since the contact to 2 assures a delivery time one second shorter than its competitor, all bundles are forwarded to 2. When the contact starts, bundles are delivered one-by-one to 2, which relays them to 4 (“Delivered” series in Figure 3). The last bundle is delivered at the end of the contact (contacts are shown at the bottom of the chart), in accordance with the estimated contact volume of 16.5 bundles.

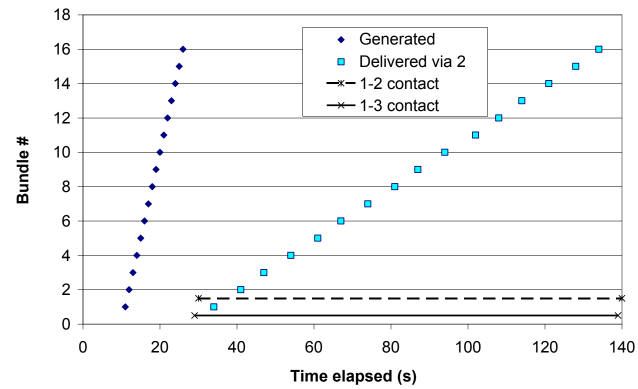

Figure 3: CGR standard with parallel equivalent routes. Bulk bundles from node 1 to 4, all delivered via 2.

The same experiment is conducted with CGR-ETO, using a low threshold in such a way the routes are always re-calculated after each bundle is forwarded. The first bundle is forwarded to 2, as before; then, thanks to ETO’s consideration of queueing delay, the two contacts are used alternately. Results (Figure 4) highlight two advantages: first, the delivery time of the last bundle is now in the middle of the contact, instead of at the end as in standard CGR, leading to a 50% reduction in total data delivery delay; second, the load balancing is perfect, which is as important as the former, because it leaves some capacity on both contacts for subsequent traffic. The drawback is the processing overhead due to route recalculations. However, the threshold mechanism provided by CGR-ETO allows the planner to choose the best trade-off between performance and computation processing.

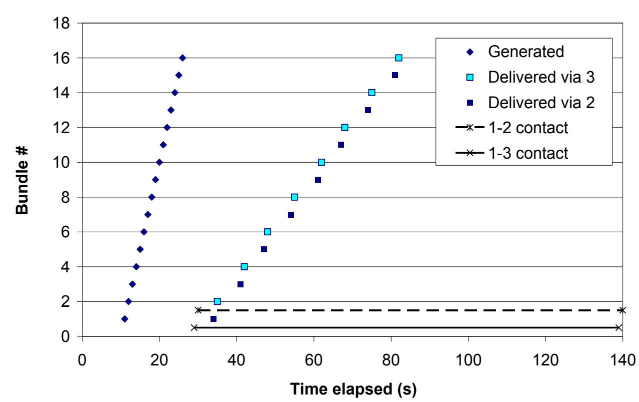

Figure 4: CGR-ETO (low threshold) with parallel equivalent routes. Bulk bundles from node 1 to 4, via either 2 or 3.

B. *Routing bundle traffic of the same priority: CGR-ETO improvements.*

Next we consider a more realistic scenario, similar to the LEO scenario in [13]. Here all contacts have the characteristics given in Table I. As in the previous case, we consider the generation of bulk bundles of 100 kB each; here, however, we produce 15 bundles and the choice is no longer between two equivalent contacts. The sole contact to 2 is nested in the first contact to 3, and also has a faster transmission rate and a larger contact volume. When standard CGR is used (Figure 5), we observe that the first 8 bundles are routed via 3 and the others via 2. The selection of intermediate node 3 is a consequence of the fact that the first contact to 3 starts before contact to 2, as in the previous case. However, once the residual capacity of this contact is exhausted (after the 8th bundle), the corresponding route is discarded and bundles are forwarded to 2, which is the best of the residual choices (the second contact to 3 starts much later). Although CGR delivers all bundles in a reasonable time, we observe, as in [13], three sub-optimal results: First, the order of delivery is scrambled; bundles 1 and 4 are delivered first, then 5-7 in parallel with 9-15, and 8 is delivered last. Although this is compliant with BP RFCs, it is not desirable. Second, bundle 8 could have been delivered earlier, if routed via 2. Third, the first contact to 3 is no longer available for subsequent traffic.

By introducing CGR-ETO (Figure 6), all aforementioned sub-optimalities are resolved. Comparing the two algorithms, the advantages provided by CGR-ETO are evident: i) no more large scale disordered delivery, ii) shorter delivery time, as all bundles are delivered before the end of contact to node 2, and iii) there is residual capacity left for contact to node 3. Note that this last advantage is a direct consequence of load balancing between parallel contacts, provided by CGR-ETO.

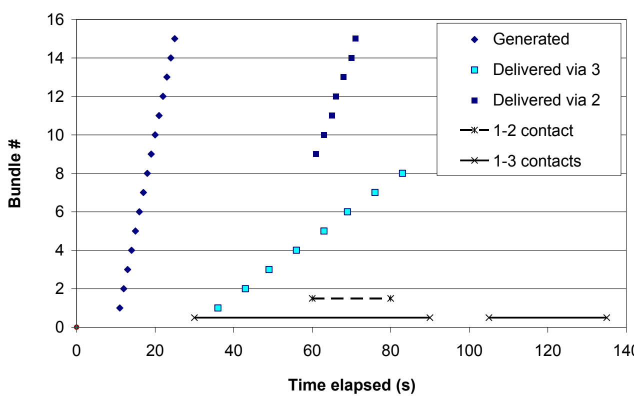

Figure 5: Standard CGR. Bulk bundles from node 1 to 4, via either 2 or 3.

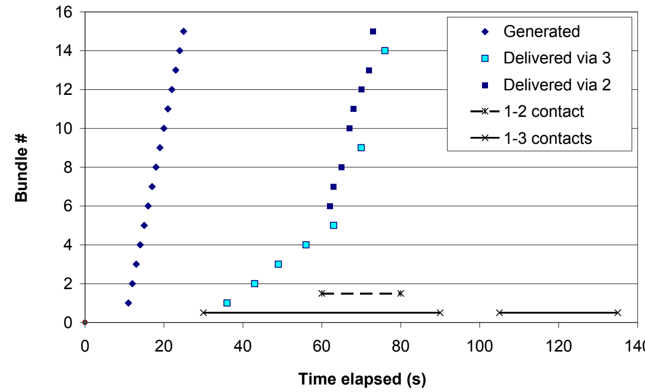

Figure 6: CGR-ETO (low threshold). Bulk bundles from node 1 to 4, via either 2 or 3.

C. *Routing bundle traffic of different priorities: The "Overbooking" problem.*

To evaluate the Overbooking Management mechanism we introduce a variant in the previous case, by increasing the priority ("expedited") of the last 4 bundles. Although seemingly minor, this modification is very challenging. Since CGR forwards the bundles in the order they are generated, the routing decisions are the same for the first 11 bulk bundles, but vary for the next four.

In Figure 7 we depict the results obtained with standard CGR. The first 11 bundles are initially forwarded as before (see Figure 5), i.e. bundles 1-8 via node 3 and bundles 9-11 via 2. For the 4 expedited bundles, CGR enforces priority-based forwarding. Thus it allocates them to the best possible contact, the first contact to node 3, regardless of the fact that 3's volume was already assigned to the first 8 low priority bundles. This results in overbooking 4 bundles. When the contact opens, the four expedited bundles are transmitted first, adding extra queueing delay for the following ones, and thus causing a temporal shift. Bundles 1-4 are delayed but sent regularly during the first contact to node 3. Bundle 5 is transmitted at the end of the contact and is successfully delivered to destination. However, the LTP acknowledgement (Report Segment) [14] does not arrive in time before contact closure (at 90s). Bundles not sent (6-8) are reforwarded immediately after it ends, along with the unacknowledged bundle 5. At this time the only possible choice is the second contact to node 3, and these bundles will be delivered on this, albeit with a very long delay. The remaining bundles, i.e. 9-11, are unaffected and thus forwarded to 2 without any delay. The sub-optimality here is that when CGR "overbooks" the first contact to 3, the contact to 2 has not yet started and still has a lot of residual capacity, therefore bundles 6-8 could have been conveniently re-allocated to it.

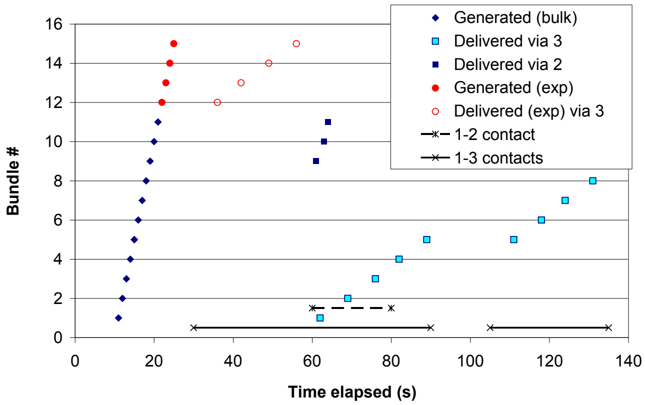

Figure 7: Overbooking with standard CGR. Bulk and expedited bundles from node 1 to 4, via either 2 or 3. Bundle 1-4 are just delayed, bundle 5 is duplicated and reforwarded, bundles 6-8 are reforwarded. Reforwarding is carried out at the end of 1-3 contact.

We continue by examining CGR-ETO (Figure 8). Similarly to the first 11 bundles of the bulk traffic case (Figure 6), bundles 1-5 and 9 are forwarded to node 3, and the rest (6-8, 10-11) to 2. Expedited bundles have higher priority and CGR-ETO correctly does not consider any queueing delay due to the previously forwarded bulk bundles. As a result, they are forwarded to 3, causing overbooking. However, since the volume of the first contact to 3 still had some capacity available (2 bundles), the overbooking is limited to bundles 5 and 9, which are reforwarded upon contact closure to 3 as in standard CGR. In general we can state that CGR-ETO, due to superior load balancing, helps prevent or limit overbooking but cannot reactively tackle it when it happens.

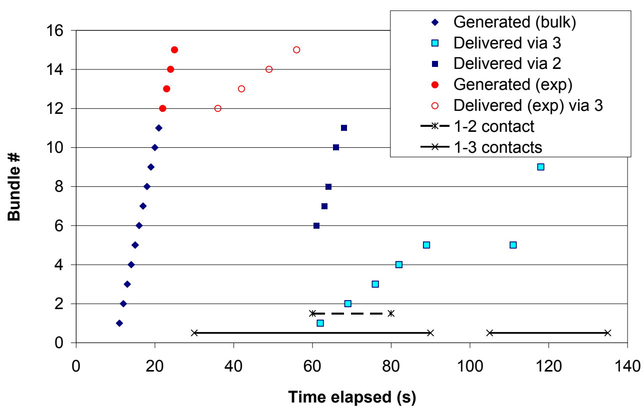

Figure 8: Overbooking with CGR-ETO (low threshold). Bulk and expedited bundles from node 1 to 4, via either 2 or 3. As before bundles 1-4 are just delayed, bundle 5 is duplicated and reforwarded. Thanks to ETO, here only one additional buindle (#9) is reforwarded instead of three. Reforwarding is carried out at the end of 1-3 contact.

Finally, we consider CGR and Overbooking Management (Figure 9). This time the overbooking is managed “a priori”, without waiting for contact end-time. Consequently, as soon as each of the expedited bundles is forwarded to node 3, the overbooking handling function reforwards a lower priority bundle from the end of the queue. One-to-one correspondence is due to the fact that all bundles have the same dimension. The upcoming contact to node 2 still has enough residual capacity left to accommodate the 4 reforwarded bundles; therefore no bundles are reforwarded to the second contact to node 3. The overbooking problem is solved.

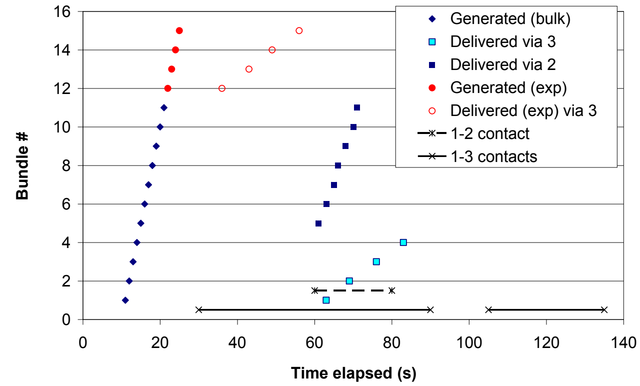

Figure 9: Overbooking with CGR standard+Overbooking management. Bulk and expedited bundles from node 1 to 4, via either 2 or 3. As before, bundles 1-4 are just delayed. Now bundles 5-8, first forwarded to 3, are reforwarded to 2 immediately after the insertion of expedited bundles. No bundles are delayed.

VII. Conclusions

In this paper we have presented two complementary enhancements to the CGR algorithm, CGR-ETO and Overbooking Management. They have been inserted as optional features in the original ION CGR code, in order to compare their performance on a testbed running the full protocol stack. CGR-ETO incorporates the available queue backlog information in routing decisions and, as observed in our experiments, offers a more efficient exploitation of future contacts with a better load balancing and a shorter delivery time. Overbooking Management proactively handles contact oversubscription and thus it improves performance in the presence of different priority traffic. The paper shows that the two enhancements complement each other and that their use can significantly improve CGR performance. We intend to share the implemented enhancements with the community in the open source framework of ION.

References

[1] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst and K. Scott, “Delay-tolerant networking: An approach to interplanetary internet", IEEE Commun. Mag., vol. 41, no. 6, pp. 128–136, Jun. 2003.

[2] V. Cerf , A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss “Delay-Tolerant Networking Architecture”, Internet RFC 4838, Apr. 2007.

[3] K. Scott, S. Burleigh, “Bundle Protocol Specification”, Internet RFC 5050, Nov. 2007.

[4] C. Caini, H. Cruickshank, S. Farrell, M. Marchese, “Delay- and Disruption-Tolerant Networking (DTN): An Alternative Solution for Future Satellite Networking Applications”, Proceedings of IEEE, Vol. 99, N. 11, pp.1980-1997, Nov. 2011.

[5] CCSDS 734.0-G-1 “Rationale, Scenarios, and Requirements for DTN in Space.”, CCSDS Green Book, Issue 1, Aug. 2010. http://public.ccsds.org/publications/SIS.aspx.

[6] A. McMahon, S. Farrell, "Delay- and Disruption-Tolerant Networking", IEEE Internet Computing, vol. 13, no. 6, pp. 82-87, Nov./Dec. 2009.

[7] S. Burleigh, “Contact Graph Routing,” Internet-Draft, July 2010. work-in-progress http://tools.ietf.org/html/draft-burleigh-dtnrg-cgr.

[8] ION code and manual: http://sourceforge.net/projects/ion-dtn/.

[9] J. Schoolcraft, S. Burleigh, R. Jones, E. Wyatt, L. Torgerson, “The Deep Impact Network Experiments–Concept, Motivation and Results”, in Proc. of SpaceOps 2010 Conference, Huntsville, USA, Apr. 2010

[10] J.Segui, E. Jennings, S. Burleigh, "Enhancing Contact Graph Routing for Delay Tolerant Space Networking,", in Proc. of IEEE GLOBECOM 2011, Houston, TX, USA, Dec. 2011, pp1-6.

[11] E.Birrane, S.Burleigh and N. Kasch, "Analysis of the contact graph routing algorithm: Bounding interplanetary paths ", Acta Astronautica, Vol. 75, pp. 108-119, June-July 2012

[12] N. Bezirgiannidis, F. Tsapeli, S. Diamantopoulos, V. Tsaoussidis, “Towards Flexibility and Accuracy in Space DTN Communications”, in Proc. of ACM CHANTS 2013, Miami, FL, USA, Sept. 2013, pp1-7.

[13] F. Apollonio, C. Caini, M. Lülf, “DTN LEO Satellite Communications through Ground Stations and GEO Relays”, in Proc. of PSATS 2013, Toulouse, France, June. 2013, pp. 1-13.

[14] M. Ramadas, S. Burleigh and S. Farrell, “Licklider Transmission Protocol – Motivation,” Internet RFC 5325, Sept. 2008.

[15] S. Burleigh, “Bundle Protocol Extended Class Of Service (ECOS)”, IETF Internet Draft, Apr.2013, work in progress, http://tools.ietf.org/html/draft-irtf-dtnrg-ecos-04.

[16] M. Demmer, J. Ott, “Delay Tolerant Networking TCP Convergence Layer Protocol”, IRTF Internet Draft, March 2014, work in progress, http://tools.ietf.org/html/draft-irtf-dtnrg-tcp-clayer-09

[17] I. Komnios et. al., "SPICE Testbed: A DTN Testbed for Satellite and Space Communications", 9th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2014), Guangzhou, China, May 5-7, 2014, to appear.