Implementation of (O-)CGR in The ONE

A. Berlati°, S. Burleigh§, C. Caini°, F. Fiorini, J. J. Messina°, S. Pozza, M. Rodolfi°, G. Tempesta°
§NASA -Jet Propulsion Laboratory California Institute of Technology, Pasadena, CA, USA
°DEI/ARCES, University of Bologna, Italy
Alessandro.berlati@studio.unibo.it scott.burleigh@jpl.nasa.gov, carlo.caini@unibo.it, federico.fiorini3@studio.unibo.it,
jakojo.messina@studio.unibo.it, simone.pozza@studio.unibo.it, michirod@gmail.com, giuseppe.tempesta2@studio.unibo.it

***Abstract*— Routing in Delay-/Disruption-Tolerant Networking (DTN) requires specific solutions as link impairments prevent the use of ordinary Internet algorithms, based on a timely dissemination of network topology information. Among DTN routing algorithms there is a dichotomy between opportunistic and deterministic (scheduled) solutions. The former are numerous and apply to terrestrial environments; CGR is the most widely supported algorithm designed for scheduled connectivity, and it is usually applied to space networks. However, in an attempt to provide a unified approach, an opportunistic variant of CGR, Opportunistic CGR (OCGR) has been recently proposed by some of the authors. Performance evaluations are normally carried out for opportunistic solutions by means of simulators, such as The ONE considered in this paper. CGR by contrast is more often studied by means of small testbeds. As the simulation approach could be complementary for CGR, and essential for OCGR, the authors have recently ported both of them into The ONE, by developing and releasing as free software a specific additional package. The aim of this paper is to show the rationale of this choice and discuss the many challenges that needed to be tackled to achieve this primary goal.**

Keywords-component; Routing, Delay-/Disruption- Tolerant Networking, The ONE, CGR, Space Networks.

I. Introduction

The Delay- and Disruption-Tolerant Networking architecture has been designed to allow communications in those scenarios where the ordinary TCP/IP architecture cannot provide satisfactory performance, because one or more of the fundamental assumptions on which the Internet architecture is based are not met. These assumptions are: short RTTs, availability of at least one end-to-end path, channel symmetry, low error rates [1], [2]. Networks where at least one of these conditions are not met are sometimes called “challenged”; this category includes space networks (both satellite and interplanetary), Mobile Ad-Hoc Networks (MANETs), emergency networks, sensor networks, military and underwater networks. The aim of the DTN architecture based on the introduction of the Bundle protocol [3] layer between Application and lower layers is to offer a common general solution instead of a variety of specific solutions limited in scope [2], [4]. DTN standardization started in IRTF and is now carried on in IETF [5]; CCSDS standardization for space applications works in parallel.

Routing in DTN networks has always been a challenging research topic [6], because channel impairments in DTN prevent the use of Internet routing algorithms based on an up-to-date comprehensive knowledge of network topology. To this end, it must be specified that DTN networks are highly heterogeneous and can be split into two main classes. Space networks are characterized by scheduled intermittent connectivity: contacts between nodes are deterministic and are known a priori, as they derive from the motion of space assets and planets. By contrast, most terrestrial DTNs are characterized by random intermittent connectivity, as contacts are opportunistic because they typically arise from casual encounters. Given this assumption, totally different routing algorithms have historically been studied for the two environments. DTN routing algorithms fall into two main categories: opportunistic, where the status information is totally or partially unknown, and deterministic, which are assumed to have a perfect knowledge of the network. Contact Graph Routing (CGR) [7] is the most widely supported algorithm designed to cope with deterministic scheduled connectivity, while for opportunistic networks there are many proposed approaches that usually employ a flooding-based strategy with some form of control dependent on their algorithm. Among the many opportunistic proposals (see [8] for a survey) the most important are: Epidemic routing [9] , ProPHET [10], Spray – and – Wait [11], RAPID [12], MaxProp [13]. An opportunistic version of CGR, called Opportunistic CGR (OCGR) has also been designed as an attempt to provide a unified approach, but it is still in an evolutionary phase [14].

Opportunistic algorithms are generally studied by means of DTN simulators, such as The ONE (Opportunistic Networking Environment) simulator considered in this paper, which is the most widely adopted and was designed to allow direct comparisons among the many opportunistic proposals, the most important of which are already included in the package [15]. These simulators are generally based on random motion of nodes to establish contacts and have been conceived to simulate many nodes, as statistics must be derived. By contrast, deterministic routing, i.e. CGR, has generally been studied by means of small testbeds, involving a limited number of nodes, as it was of primary interest to study the ability of the algorithm to cope with particular challenges (see [16] for an interesting exception). The advantage of these testbeds running on real or virtual machines is that the full protocol stack is involved and that channel emulators can be inserted between nodes to add any kind of link impairments; the disadvantage is that only a limited number of nodes can be involved. That said, the authors believe that the simulation approach can effectively complement the use of small scale testbeds for CGR, in particular to study the scalability of the algorithm, for which of course a high number of nodes would be necessary. Moreover, OCGR obviously requires an opportunistic environment for performance evaluation. Therefore, it was decided to port (O-)CGR into The ONE, in an attempt to have a unified platform for DTN routing evaluations. The challenges to face and the results of this study are described in this paper. If accepted, a demo will be presented at the conference.

II. CGR AND OCGR

A. CGR

CGR is a dynamic algorithm that computes routes based on the "contact plan," a time-ordered list of scheduled transmission opportunities, i.e. "contacts". Each contact entry is defined by a start and a stop time, and by a nominal transmission speed; as channels in space are often asymmetric, two entries with possibly different transmission speeds are usually present for each contact. Each node uses the contacts in the contact plan to build a "routing table" data structure. It is worth stressing that by contrast to Internet algorithms, routes do not need to be continuous. Each segment of the path from source to destination is an opportunity to send data from node X to node Y; once a bundle has reached node Y if the link to node Z is temporarily closed the bundle may be kept in storage, awaiting the start of the next contact. Each route is also associated to a *forfeit time*, i.e. the latest time by which the bundle must be forwarded to the route's entry node in order to have any chance of traversing the route itself.

As routes are known in advance, route computation is performed as soon as a bundle is passed to the bundle protocols by upper layers, i.e. when it is generated on the source node, or when it arrives at intermediate nodes. As a result, the bundle is inserted in a queue towards the first node of the selected path. By contrast to opportunistic algorithms, bundles are *forwarded*, i.e. no additional copies are created, except in the case of "critical" bundles, for which multiple copies can be created to maximize the chances of success and minimize the delivery time. Routes are recomputed whenever the contact plan is updated.

CGR is naturally more complex than most opportunistic algorithms and has undergone many modifications [7], [17] and at present it is under standardization by CCSDS, relabeled as Scheduled Aware Bundle Routing [18]. Among the latest enhancements introduced in the ION implementation [19] used in this work are the computed "earliest transmission opportunity" (ETO), to take into account the delay due to bundles already enqueued (currently on the first hop only), and Overbooking Management, to efficiently manage the contact oversubscription that can derive from the forwarding of high priority bundles whenever the selected contact had previously been allocated to lower priority bundles [20].

B. OCGR

Opportunistic Contact Graph Routing is an extension to CGR aimed at enlarging its applicability from deterministic space networks to opportunistic terrestrial networks. To this end, the contact plan has been extended to add discovered and predicted contacts in addition to scheduled ones. These contacts have different level of "confidence" (a sort of likelihood of happening). In brief we have:

- Scheduled contacts; these are known a priori and have confidence 1.
- Discovered contacts; these opportunistic contacts are added to the contact plan on the spot, when they happen. Upon termination, their start and stop times, as well as their volumes (the product of the contact length with the transmission speed) are saved. They have confidence 1.
- Predicted contacts; these are calculated on the basis of discovered contacts, to take advantage of history of previous encounters (in the hope that they are not completely random but follow predictable patterns); their level of confidence is less than 1.

For any newly discovered contact, the communicating nodes exchange all contact log entries, then discard all previously computed predicted contacts and use the updated contact history to compute new predicted contacts. Routing is performed on the basis of the updated contact plan in the usual way, except that the confidence in the resulting forwarding decisions is less than total if the selected path includes predicted contacts. If the level of confidence is less than a given threshold, OCGR sends multiple copies of the bundles, thus extending the mechanism previously limited in CGR to "critical bundles". Note that OCGR is still a work in progress, and all elements of the design remain open to discussion and revision [14].

III. THE ONE

The ONE is a Java based simulator designed to test and compare opportunistic routing algorithms [15]. It was developed at Aalto University who now maintain it together with Technische Universität München (Connected Mobility). It is released under GPLv3 and the latest version (at present v1.6.0) can be downloaded from [21]. Its main characteristics and features, knowledge of which is necessary to understand CGR integration, are reported below.

A. Node movement, contacts and routing

The simulation environment is based on node movements, which can follow different random models or be derived from real traces. Nodes have one or multiple radio interfaces, with associated ranges and transmission speeds. Contacts are opportunistic and derive from the movement of nodes: a contact start when two nodes with a common radio interface become close enough, and stops when they go out of range. During a contact, bundles (but in The ONE the more generic term "messages" is used) are exchanged between nodes at the speed associated to the interface, following the rules dictated by the routing protocol adopted. Messages are generated at random intervals by message generators (several options available) which can be activated on specific subsets of nodes, which can also be destinations, while other nodes act only as relays.

B. Visual interface and logs

The ONE visual interface serves two aims. First, it allows the user to fine-tune the running simulation, not only by means of the start/stop/pause commands, but also by setting specific conditions that pause the simulation when met. Second, it allows the user to follow the node movement and the data exchanged. Once paused, the user can inspect the situation at a given moment (e.g. which messages are exchanged or buffered). When intensive simulations are needed, The ONE can run in "batch" mode, i.e. without the graphic interface, making the simulation faster. The ONE can produce a wide variety of reports, as well as general statistics, such as the percentage of messages delivered, relayed, etc.

C. *Settings*

The ONE simulations are based on the parameters contained in one or more settings files. The basic rule is that general settings are given in the "default_settings.txt" file, then these settings can be overridden or augmented by additional files. This method proves both effective and convenient. As The ONE is very flexible, many parameters must be set; thus it is preferable to work differentially, by changing only a few parameters at a time, which can easily be done by this mechanism.

D. *Routing algorithms*

From the Java class "Active router" derive the classes implementing the opportunistic routing protocols in The ONE (see the "routing" directory of The ONE package).

IV. CGR Implementation

The CGR implementation in The ONE has two primary aims: to allow the user to test CGR scalability, by running CGR on networks consisting of many nodes, and to facilitate the Opportunistic version development, by allowing direct performance comparisons with the best opportunistic routing protocols. To keep the code as modular as possible, we tried hard not to modify the current code of ONE, unless strictly necessary, and instead add new classes whenever possible. Most importantly, we decided to avoid any duplication of the CGR code, by transplanting it verbatim from ION into The ONE instead of writing a new implementation. The rationale of this was to avoid any inconsistency and facilitate future updates.

In implementing CGR into The ONE we had to tackle various challenges, notably:

- The ONE is written in Java but CGR in C. We used JNI (Java Native Interface) to link CGR C code to the new CGR routing classes in Java, a major task.
- CGR algorithm is contained in a single file "libcgr.c" in ION. The code, however, is interfaced with many routines and structures present in the ION environment but not in The ONE. It was therefore necessary to build a sort of ION emulation environment within The ONE, to avoid modifying the CGR code.
- While priorities have been considered in RFC4838 (the DTN architecture) and in CGR, these are missing in The ONE. As we deemed priorities enforcement an important, if not essential, feature of CGR, we have introduced them in The ONE. More precisely, instead of modifying the existing code, we have introduced a new generator class and a variant of the Epidemic router that is able to enforce priorities. CGR and OCGR classes have been developed in two variants, with and without priority support, the latter for compatibility with standard traffic generators. The Overbooking Management mechanism, which is related to priorities, has also been implemented on The ONE side.
- CGR was designed for scheduled contacts and obviously assumes that the list of contacts provided in the "contact plan" are really going to happen; the problem here is that although we can easily pass a contact plan to CGR, these contacts are not enforced by The ONE. In fact, in The ONE all contacts are random, deriving from the motion of nodes. Although it is possible to use real traces instead of random movements, any attempt to emulate the motion of space assets and planets would be clearly impractical. Therefore, we introduced the possibility of enforcing contacts in The ONE on the basis of an external contact plan (in ION format) as an alternative to the usual way based on node mobility. Note that OCGR versions do not require (but are compatible with) this feature, as they can rely on discovered and predicted contacts only.
- In ION, each contact between two nodes has its own transmission rate. In The ONE transmission rates are associated to radio interfaces: one interface, one rate. It was therefore necessary to modify The ONE code to enforce different rates when contacts are dictated by an external contact plan.
- In the space environment nodes can be very far away and signal propagation time cannot be neglected; by contrast, in The ONE there is little notion of anything related to real transmission (from Bundle to Physical layer), as it is considered irrelevant to overall routing evaluations. Therefore, as "range" instructions in ION contact plans cannot be enforced in ONE, they are ignored, by assuming zero propagation delay instead.
- In CGR, bundles are routed and put in queues as soon as generated, or received, as present and future contacts are known (or just predicted in OCGR). In The ONE, bundles are routed only on the spot, when there is an encounter between two nodes. To accommodate the CGR behavior, it was necessary to build into The ONE queues towards proximate nodes, i.e., in practice the nodes to be encountered in the contact plan. One queue is implemented for each priority class (bulk, normal, expedited, as in [2]). Interestingly, this mechanism has somewhat influenced the development of the latest ION version (3.6.0, recently released), where queues have been moved from the convergence layer to the bundle layer, as here.
- In The ONE, transmission is normally half-duplex, i.e. a node cannot transmit before having completed reception of all messages from its peer; as this assumption could potentially stress relay node buffers (obliging them to inflate with all incoming messages before being deflated by the transmission of the first outgoing messages), we have made transmission full duplex in the new router classes; this also makes the simulations more faithful to real systems.
- A still missing feature to complete the support of deterministic environments, useful for CGR testing, is the implementation of deterministic message generators. This is left to future versions.

V. The cgr-jni-Merge Package

As a result of the implementation work described above, the package "cgr-jni-Merge" (cgr-jni, merged version), has been released as free software [22]. This package includes:

- Four CGR-related new router classes, namely ContactGraphRouter (without priorities), PriorityContact-GraphRouter, OpportunisticContactGraphRouter (without priorities) and PriorityOpportunisticContactGraphRouter.
- The new "PriorityEpidemicRouter" class, to be used alone or as a benchmark for priority versions of CGR routers.

- The "PriorityMessageEventGenerator" class for generating messages with priorities.
- The class CPEventsReader, to enforce contacts provided in an external contact plan (in the ION format).
- The classes PriorityMessageStatsReport and OCGR MessageStatsReport to derive priority and OCGR stats.
- The CPEventLogReport class to log contacts opened and closed by The One as a result of node movement.
- An independent Java program, called ContactPlanCreator to convert the .txt file created by the CPEventLogReport into an ION compliant contact plan.

Note that only the four classes related to CGR require linkage with native C code (included in the package). All the others are written in Java and some of them could be used independently of CGR routers, such as those related to priorities. The rationale for the ContactPlanCreator program is worth an explanation. It could be used to pass to CGR contacts that are actually derived from motion. In practice, this is possible by running a simulation twice with the same random generator seed. In the first run, the Epidemic router could be used and opportunistic contacts logged; then, after converting the log file into an ION contact plan thanks to the ContactPlanCreator, the simulation could be repeated with CGR instead of Epidemic. CGR would be passed the obtained contact plan, which would "magically" predict the pseudorandom contacts, as the same seed as before is used. This way, it is possible to compare performance achievable with Epidemic with that which is theoretically achievable by CGR. The former does not exploit any state information, while the latter has a full knowledge. The two cases, being extreme, could be used as opposite benchmarks for other evaluations, where routers have only a limited knowledge of future contacts.

VI. Conclusions and Future Work

This paper is focused on the inclusion of CGR and its opportunistic variant OCGR into The ONE, the most widely adopted DTN simulator. Although the "cgr-jni-Merge" package described in this paper has been primarily developed to this end, it also adds a few auxiliary features that could be used independently of CGR/OCGR routers. These extensions are: the support of priorities (message generators with priorities, Epidemic routing with priorities); the possibility of converting a ONE log into an ION contact plan, and last but foremost, the possibility of enforcing deterministic contacts provided in an external contact plan. The authors hope that all these extensions, released as free software, can be useful to both the opportunistic and deterministic DTN research communities. To this end these additions will be proposed to The ONE maintainers for possible inclusion in the official version.

Acknowledgments

Part of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, R. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary Internet," IEEE Commun. Mag, vol. 41, no. 6, June 2003, pp. 128-136.

[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] IETF DTN web site: https://datatracker.ietf.org/wg/dtn/about/

[6] Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant network", in Proc. of ACM SIGCOMM 2004, Portland, Aug/Sept. 2004, pp. 145-157.

[7] 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 Commun. Mag., vol. 53, no. 3, March 2015, pp. 38-46.

[8] R. J. D'Souza and Johny Jose, "Routing Approaches in Delay Tolerant Networks: A Survey", International Journal of Computer Applications (0975 - 8887), Volume 1 – No. 17, 2010.

[9] Amin Vahdat and David Becker, "Epidemic routing for partially connected ad hoc networks", Technical Report CS-2000-06, Department of Computer Science, Duke University, April 2000.

[10] A. Lindgren, A. Doria E. Davies, and S. Grasic, "Probabilistic Routing Protocol for Intermittently Connected Networks", Internet RFC 6693, Aug. 2012.

[11] T. Spyropoulos, K Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks", in Proc. of 2005 ACM SIGCOMM workshop on Delay-tolerant networking, WDTN'05, 2005, pp. 252-259.

[12] A. Balasubramanian, B. Levine, and A. Venkataramani, "Replication routing in dtns: A resource allocation approach," IEEE/ACM Trans. on Netw., vol. 18, no. 2, pp. 596 –609, Apr. 2010.

[13] J. Burgess, B. Gallagher, D. Jensen, and B. Neil Levine., "MaxProp: Routing for vehicle-based disruption-tolerant networks", in Proc. of IEEE INFOCOM, 2006, Barcelona, Spain, pp. 1-11.

[14] S. Burleigh, C. Caini, J.J. Messina, M. Rodolfi, Toward a Unified Routing Framework for Delay-Tolerant Networking, in Proc. of IEEE WiSEE 2016, Aachen, Germany, Sept. 2016, pp. 82 - 86, DOI: 10.1109/WiSEE.2016.7877309

[15] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation", in Proceedings of the 2nd International Conference on Simulation Tools and Techniques. New York, NY, USA: ICST, 2009.

[16] J. A. Fraire, P. Madoery, S. Burleigh, M. Feldmann, J. Finochietto, A. Charif, N. Zergainoh, R. Velazco, "Assessing Contact Graph Routing Performance and Reliability in Distributed Satellite Constellations", Journal of Computer Networks and Communications, in press.

[17] 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

[18] CCSDS "Schedule-aware bundle routing", CCSDS White Book, May 2017, work in progress.

[19] S. Burleigh, "Interplanetary overlay network design and operation V3.3.1," JPL D-48259, Jet Propulsion Laboratory, California Institute of Technology, CA, May 2015. [Online]: http://sourceforge.net/projects/ion-dtn/files/latest/download

[20] N. Bezirgiannidis, C. Caini,V. Tsaoussidis, "Analysis of contact graph routing enhancements for DTN space", International Journal of Sat. Commun. and Networking, pp.695-709, No.34, On line Nov. 2015.

[21] The ONE web site: https://akeranen.github.io/the-one/

[22] cgr-jni-merge download site: https://github.com/alessandroberlati/cgr-jni/tree/Merge