Contact Graph Routing for Interplanetary Overlay Networks - DTSN

Armaan Garg

July 3, 2022

Contact Graph Routing for Interplanetary Overlay Networks - DTSN

Armaan Garg[1] *

July 3, 2022

Abstract

Studies have been proposed for development of satellite networks and Delay Tolerant Networks (DTN) has been acknowledged as one of the potential candidate to develop these networks. All the characteristics of DTN of non-continuous and lack of end-to-end path have been considered and satellite networks have been developed accordingly. For such dynamic and huge networks (in terms of distance), topology prediction and routing techniques are very complex and the routing tables are also huge when all possible routes are taken into account which leads to a very limited and degraded forwarding performance. To form the routing tables for such complex networks there exists few algorithms, such as Contact Graph Routing(CGR), but it comes with a lot of complexity, implications and issues. Considering the DTSN constellation scenarios, the simulation results show that using these novel routing table techniques the calculation effort and network flow metrics could be reduced many-fold.

Keywords

Delay Tolerant Satellite Networks, Satellite networks, Contact Graph Routing

1 Introduction

There has been huge research boost in the field of space with many showing interest on how satellite networks could be developed and improved for better image/video and communication services. This had lead to development of huge and complex infrastructures and technological architectures to provide a steady and efficient method for transmission of data between ground and space networks. Considering the implications and huge costs for such networks, DTN has come into picture which can transfer large number of data bits from source to destination. This could help in developing such networks at relatively low-costs and less complex infrastructure. DTN architecture assumes no lower bounds on the propagation delay nor the end-to-end continuous transmission. If the

*[1]The author is with the Department of Computer Science and Engineering, Indian Institute of Technology Ropar, Punjab, India armaan.19csz0002@iitrpr.ac.in

path from source to destination is not available, the data might be stored at intermediate nodes until the path forward is active again. Each intermediate node takes the responsibility of the packets which it forwards, since on lost of transmission in-between it needs to transfer another copy of same packet again.

DTN architecture could be used for both satellite-to-satellite and ground-to-satellite communication. Such paradigm is known as Delay Tolerant Satellite Networks (DTSNs) [3]. These vary from traditional DTN model in terms of the size and topology of the network and also the agile nature of the network structure (the rate at which it changes). The current routing protocols and algorithms of transitional DTNs [6] are limited to the distance over which the system could handle the transmission, but transmission in deep space is highly varied from this interns of the network distributed area, which makes DTSNs a potential candidate for future research work in the domain of communication in deep space.

The scalability of current DTNs structure is poor and when the graph topology is extended beyond a limit the performance of such networks sees a steep drop. Apart from this, an all-to-all interconnection path is considered, which makes the network structure and its extension way more complex. Looking at this problem, a model was proposed under the name of contact graph a suitable structure that facilitates the distributed execution of adapted Dijkstra's searches. This idea lead to the development of the earlier version of distributed Contact Graph Routing (CGR) algorithms. Its has been quite a while since the development of these contact graph based counting models, recently only the research interest has been shifted towards the computational structure of the CGR. Indeed, these routing table computation has direct impact on the final network and the flow of data packets and the overall transmission effort.

At first, this study focuses on the weaknesses of the Contact Graph Routing [2] and introduces first-ending and first-depleted which improve the building of the routing tables in terms of efficiency and precision. Also, techniques have been proposed to tackle the issue of dynamic and fast changing topology pattern of DTSNs. To validate the methodology, one-route and per-neighbor techniques were presented which are novel alternatives to dynamically update route tables on-demand, cutting out on the computational effort without sacrificing data-forwarding efficiency. To compute and the performances of these techniques the reference CGR implementation in ION v3.6 has been used and analysed by means of simulations over two appealing Low-Earth-Orbit (LEO) DTSN constellations [7].

2 Routing in Delay Tolerant Satellite Networks

2.1 Contact Graph Routing

Contact Graph Routing (CGR) [4] is a dynamic routing system that computes routes through a time-varying topology of scheduled communication contacts in a network based on the DTN (Delay-Tolerant Networking) architecture. It

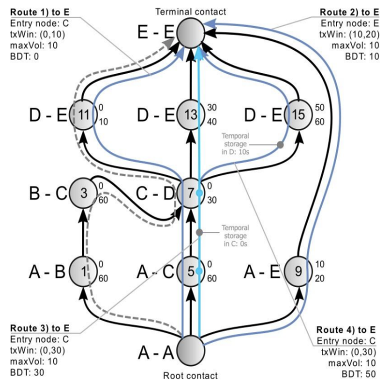


Figure 1: Contact Graph CG^E

is designed to enable dynamic selection of data transmission routes in a space network based on DTN. This dynamic responsiveness in route computation should be significantly more effective and less expensive than static routing, increasing total data return while at the same time reducing mission operations cost and risk. The basic strategy of CGR is to take advantage of the fact that, since flight mission communication operations are planned in detail, the communication routes between any pair of bundle agents in a population of nodes that have all been informed of one another's plans can be inferred from those plans rather than discovered via dialogue (which is impractical over long one-way-light-time space links).

Messages that convey this planning information are used to construct contact graphs (time-varying models of network connectivity) from which CGR automatically computes efficient routes for bundles. Automatic route selection increases the flexibility and resilience of the space network, simplifying cross-support and reducing mission management costs. There are no routing tables in Contact Graph Routing. The best route for a bundle destined for a given node may routinely be different from the best route for a different bundle destined for the same node, depending on bundle priority, bundle expiration time, and changes in the current lengths of transmission queues for neighboring nodes; routes must be computed individually for each bundle, from the Bundle Protocol agent's current network connectivity model for the bundle s destination node (the contact graph). This places a premium on optimizing the implementation of the route computation algorithm. The scalability of CGR to very large networks remains a research topic. The information carried by CGR contact plan messages is useful not only for dynamic route computation, but also for the implementation of rate control, congestion forecasting, transmission episode initiation and termination, timeout interval computation, and re-transmission timer suspension and resumption.

A contact plan captures the time-evolving nature of a dynamic topology

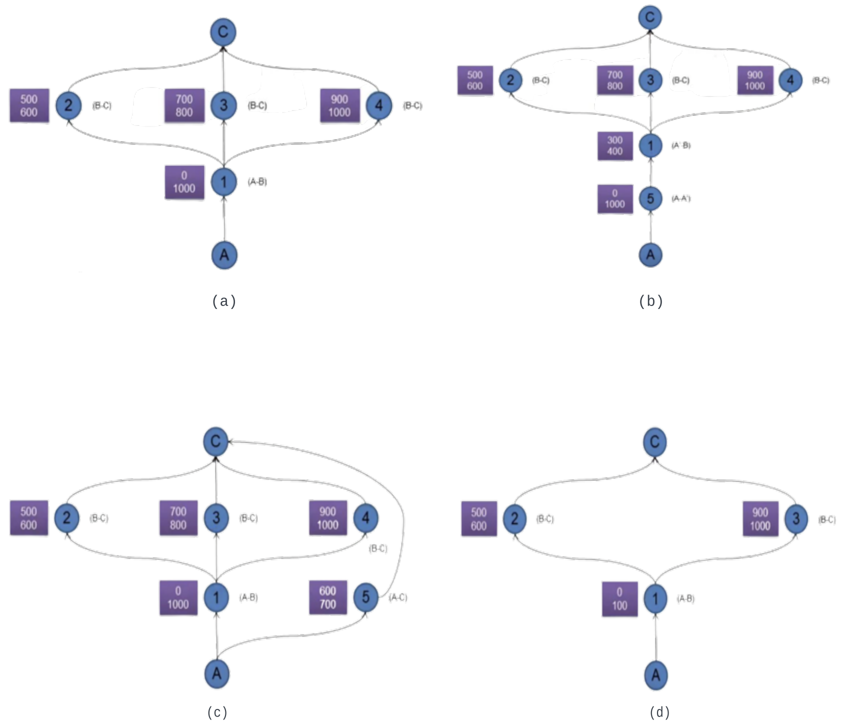


Figure 2: Illustrating Anchoring

which can be presented in a static graph or in a timeline view. A contact graph for destination node D at source node S is a conceptual directed acyclic graph $CG^D = (V, E)$ where, V is $C^{t1,t2}$ in the contact plan and E is edge added where receiving node of a contact matches the source node of the next contact in the path. Figure 1 shows an example contact graph with different possible routes from source A to destination E.

Anchoring technique [1] is applied when the initial is not the limiting contact (the node with earliest end time in path), then suppresses the remote limiting contacts while anchoring the initial contact works in the favour. In figure 2(a) node 1 serves the purpose of anchor node with end time 1000 greater than all other nodes, now if we suppress remote nodes 2, 3, 4 one by one as we keep recording the paths; it would prevent the duplication of paths in the routing table. But anchoring also has some limitations which are described further.

In figure 2(b), if node 5 serves as the anchoring point with maximum end time, so the next immediate node 1 will be suppressed according to the technique when 1st path 5-1-2-C is found. This would prevent the other paths from being discovered which are not even redundant. Similarly, in figure 2(c) node 1 will become the anchoring node which would definitely lead to discovery of the three paths via 2, 3, 4 but the node 5 will be left undiscovered. Further in figure 2(d), removal of initial/ limiting contact 1 would lead to blocking of path via 3 which is not even anchoring exactly.

2.2 Routing Table Methodologies

2.2.1 Static Route Table

In the basic implementation of CGR, all the paths are calculated once and stored in the routing table. The following methods concentrate on forming as precise as possible table with minimum duplicates but all the paths possible to ensure bundle delivery.

- **First Ending:** Unlike anchoring where initial node is suppressed, this method suppresses the ending contact in the last path found.
- **First Depleted:** All the methods discussed till now were based on the time at- tribute, but here the main factor is the volume. The node whose volume gets fully booked if data were to flow through that path, is suppressed.

2.2.2

Here we highlight a different perception to maintain the routing tables with as minimum as possible but best path entries.

- **One Route:** In this method, each time when a bundle is forwarded, the entry validity is verified and updated if there is no route in the entry (initial condition of the table), other attributes like tWin is due or maxVol was reached are also considered.
- **Per Neighbour Route:** The first contacts leading to different nodes than the corresponding n entry node shall be suppressed from the search when computing the n position in the route table for the critical data which is expected to be forwarded through all possible paths to a destination. Basically, a fixed n number of path entries are maintained in the routing table.

3 Design

We have used ION 3.6 for the implementation of Contact Graph Routing. The Inter-planetary Overlay Network (ION) is an implementation of Delay-Tolerant Networking (DTN) architecture as described in Internet RFC 4838. It is designed to enable inexpensive insertion of DTN functionality into embedded systems such as satellites. The DTN architecture is much like the architecture of the Internet, except that it is one layer higher in the familiar ISO protocol stack. The DTN analog to the Internet Protocol (IP), called Bundle Protocol (BP), is designed to function as an "overlay" network protocol that interconnects Internets – including both Internet-structured networks and also data paths that utilize only space communication links as defined by the Consultative Committee for Space Data Systems (CCSDS) – in much the same way that IP interconnects subnets such as those built on Ethernet, SONET, etc. The ION distribution comprises of various packages like ICI (interplanetary communication infrastructure), a set of libraries that provide flight-software-compatible support for functions on which the other packages rely, such as dynamic memory management, non-volatile storage management, and inter-task communication via shared memory; BP (bundle protocol); DGR (datagram retransmission), a UDP reliability system that implements congestion control; LTP (licklider transmission protocol), a DTN convergence layer; etc.

3.1 Configurations and Flow Model

Figure **??** shows the basic flow of the implementation done for fetching the final contact plan. The following gives the overview about various modules involved:

- IONadmin is the administration and configuration interface for the local ION node contacts and manages shared memory resources used by ION. It specifies contact bandwidths and one-way transmission times. This is important in deep-space scenarios where the bandwidth must be artificially limited and where acknowledgments must be timed according to propagation delays. It is also vital to the function of contact-graph routing.
- LTPadmin is the administration and configuration interface for LTP operations on the local ION node. It specifies spans, transmission speeds, and resources for the Licklider Transfer Protocol convergence layer.
- BPadmin is the administrative interface for bundle protocol operations on the local ion node. It specifies all of the open endpoints for delivery on your local end, which convergence layer that you intend to use. With the exception of LTP, most convergence layers are fully configured in this file.
- IPnadmin is the administration and configuration interface for the IPN addressing system and routing on the ION node.It maps endpoints to convergence-layer addresses. For example use TCP/IP and LTP (over IP/UDP), so it maps endpoint IDs to IP addresses. This file essentially functions as the static routing table for the IPN naming scheme.
- DTNadmin is the administration and configuration interface for the DTN addressing system and routing on the ION node. It acts as the routing table for the DTN naming scheme.
- killm is a script which tears down the daemon and any running ducts on a single machine (use IONstop instead).

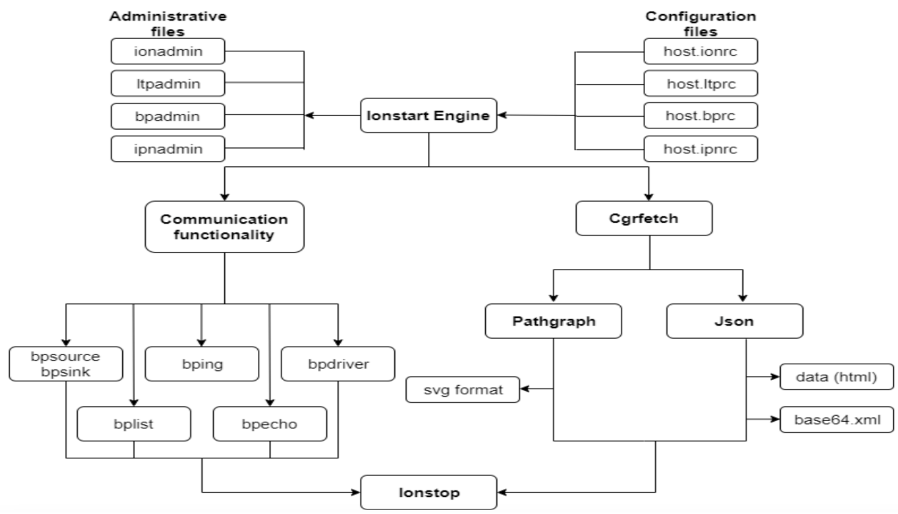


Figure 3: Implementation Flow

- IONstart is a script which completely configures an ION node with the proper configuration file(s). ionstop is a script which completely tears down the ION node.
- IONscript is a script which aides in the creation and management of configuration files to be used with IONstart.
- BPsource and BPsink are for testing basic connectivity between endpoints. BPsink listens for and then displays messages sent by BPsource. BPsendfile is used to send a file between ION nodes.

4 Contact Graph Routing in ION

CGR relies on accurate contact plan information provided in the form of contact plan messages that currently are only read from ionrc files and processed by ionadmin, which retains them in a non-volatile contact plan in the RFX database, in ION's SDR data store [5]. Contact plan messages are of two types: contact messages and range messages. Each contact message has the following content: the starting UTC time of the interval to which the message pertains, the stop time of this interval (in UTC), the Transmitting node number, the Receiving node number, the planned rate of transmission from node A to node B over this interval (in bytes per second).Each range message has the following content: the starting UTC time of the interval to which the message pertains, the stop time of this interval (in UTC), node number A and B, the anticipated distance between A and B over this interval (in light seconds).

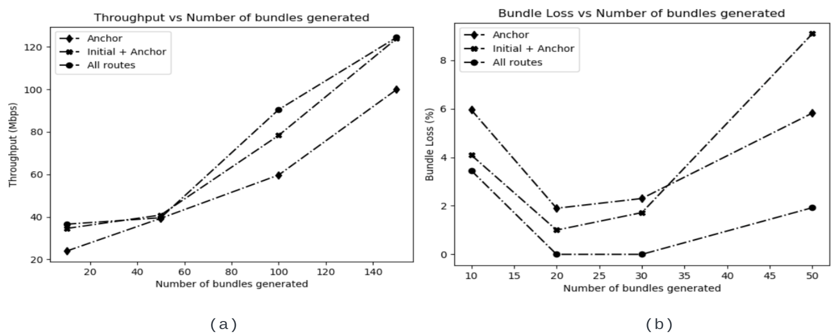
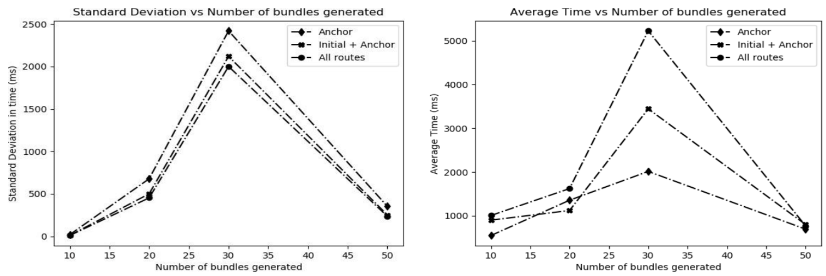


Figure 4: Variation of different parameters vs number of bundles generated

Each node uses Range and Contact messages in the contact plan to build a routing table data structure. The routing table constructed locally by each node in the network is a list of entry node lists, one route list for every other node D in the network that is cited in any Contact or Range in the contact plan. Entry node lists are computed as they are needed, and the maximum number of entry node lists resident at a given time is the number of nodes that are cited in any Contacts or Ranges in the contact plan. Each entry in the entry node list for node D is a list of the neighbors of local node X; included with each entry of the entry node list is a list one or more routes to D through the indicated neighbor, termed a route list. Each route in the route list for node D identifies a path to destination node D, from the local node, that begins with transmission to one of the local node's neighbors in the network– the initial receiving node for the route, termed the route's entry node. For any given route, the contact from the local node to the entry node constitutes the initial transmission segment of the end-to-end path to the destination node.

Additionally noted in each route object are all of the other contacts that constitute the remaining segments of the route's end-to-end path. Each route object also notes the forwarding cost for a bundle that is forwarded along this route. CGR is configured to deliver bundles as early as possible, so best-case final delivery time is used as the cost of a route. Other metrics might be substituted for final delivery time in other CGR implementations. However, if different metrics are used at different nodes along a bundle's end-to-end path it becomes impossible to prevent routing loops that can result in non-delivery of the data. Finally, each route object also notes the route's termination time, the time after which the route will become moot due to the termination of the earliest-ending contact in the route.



(a) (b)

Figure 5: Variation of different parameters vs number of bundles generated

5 Results

The all-routes based method in CGR (i.e, basic CGR) has been compared with CGR based on anchor point (one node is suppressed) and suppression of an initial node along with anchoring node. Figure 4 shows the variation of parameters like throughput, delivery time, bundle loss with number of bundles generated. In 4(a), it is observed that throughput for all routes is more than both the other methods and similarly throughput for initial + anchor is more than anchoring one. This is evident as in case of all routes, the maximum amount of bundle delivery (bundles in bytes) is guaranteed as all possible paths are available and somehow the bundle will reach the destination, when anchoring is employed some paths might have been blocked when compared to initial+anchor, therefore only anchoring performs worst. The same reasoning also explains the patterns in figure 4(b) for bundle loss as all-routes exhibit minimum share of bundle loss or in other words maximum bundle delivery. Again, anchoring without suppressing initial node exhibits maximum bundle loss.

In figure 5 we compare time variations where all routes take the maximum time because all types of paths are entered in the routing table which means it takes more processing/ computation time when compared to anchoring with and without initial node suppression. Anchoring takes the minimum time as more paths are blocked while calculating than, when initial node is also suppressed.

6 Conclusion

The main aim of the project was to extend the wireless communications networks to space satellites. ION-DTSN is a brief explanation of that idea and NASA is currently working to extend these delay tolerant networks to deep space communications. The CGR routing procedures used in ION simulator respond dynamically to the changes in network topology that the nodes are able to know about, i.e., those changes that are subject to mission operations control and are known in advance rather than discovered in real time. This dynamic responsiveness in route computation should be significantly more effective and less expensive than static routing, increasing total data return while at the same time reducing mission operations cost and risk. Further, our results have been thoroughly evaluated in terms of delivery, performance and size efficiency under varying conditions. The time-evolving nature of DTSNs requires different route table calculation paradigms i.e, the nature of the DTSN topology is even more relevant and dictates how different DTN algorithms and strategies perform, suggesting that there is not a one-size-fits-all solution. Further research is being carried out in this field by space organizations focused on the energy saving techniques which degrade the communication over wireless channels substantially, for example instead of radio waves magnetic waves can be used for more efficient communication, the latest being the Lifi.

References

[1] Catalina Aranzazu Suescun and Mihaela Cardei. Anchor-based routing protocol with dynamic clustering for internet of things wsns. *EURASIP Journal on Wireless Communications and Networking*, 2019, 05 2019.

[2] J. A. Fraire, P. Madoery, S. Burleigh, M. Feldmann, J. Finochietto, A. Charif, N. Zergainoh, and R. Velazco. Assessing contact graph routing performance and reliability in distributed satellite constellations. *Journal of Computer Networks and Communications*, 2017:1–18, 2017.

[3] Juan Fraire, Jorge Finochietto, and Scott Burleigh. *Delay-Tolerant Satellite Networks*. 01 2018.

[4] Juan Fraire, Olivier Jonckère, and Scott Burleigh. Routing in the space internet: A contact graph routing tutorial. *Journal of Network and Computer Applications*, 174:102884, 01 2021.

[5] Sanjeev Gunawardena, Thomas Pany, and James Curran. Ion gnss software-defined radio metadata standard. *NAVIGATION*, 68(1):11–20, 2021.

[6] Seung-hun Oh, Ilkyun Park, Seok-Kap Ko, Byung-Tak Lee, and Young Kim. Transitional approaches in dtn routing algorithm. pages 1–4, 01 2011.

[7] Zhicheng Qu, Gengxin Zhang, and Jidong Xie. Leo satellite constellation for internet of things. *IEEE Access*, PP:1–1, 08 2017.