2013

The Interplanetary Internet Implemented on a Terrestrial testbed

Joyeeta Mukherjee
University of Nebraska-Lincoln, joyeetam@cse.unl.edu

Byrav Ramamurthy
University of Nebraska-Lincoln, bramamurthy2@unl.edu

IEEE ICC 2013 - Selected Areas in Communications Symposium
Pages: 4298 - 4303, DOI: 10.1109/ICC.2013.6655240

The Interplanetary Internet Implemented on a Terrestrial testbed

Joyeeta Mukherjee and Byrav Ramamurthy
Department of Computer Science and Engineering
University of Nebraska – Lincoln
Email: {joyeetam,byrav}@cse.unl.edu

Abstract—**Future space exploration demands a space network that will be able to connect spacecrafts with one another and in turn with Earth's terrestrial Internet and hence efficiently transfer data back and forth. The feasibility of this technology would enable everyone to directly access telemetric data from distant planets and satellites. The concept of an Interplanetary Internet (IPN) is only in its incubation stage and considerable amount of common standards and research is required before widespread deployment can occur to make IPN feasible.**

Delay is an important factor in space communication and its nature is completely different from terrestrial delay environments. Moreover, space deployments and testing in space environments are very costly and time consuming. We propose a design of the IPN and implement it with the Interplanetary Overlay Network (ION) software module on physical nodes on the terrestrial ORBIT testbed. Two space network scenarios are designed and experimentally evaluated to verify the correctness of the network implementation. We also focus on the study of bundle transmission delay and separately evaluate the effect of bundle size and number of bundles. The experimental evaluation provides insights into the factors which caused delay in bundle transmission such as custody refusal, expiration of bundle lifetime and congestion.

I. Introduction

Space communication and networking research has added a new engineering and scientific era to the history of space exploration. The early phase of space communication used radio signal shot towards spacecraft antennas whenever they came into view. Telecommunications software lacked universality and differed from one mission to another. This, in turn, led individual flight projects to acquire and operate their own specialized space communication networks. To overcome these problems we need to develop a space network that can be interconnected, standardized and evolved over the future decades and such motivations led to novel networking architectures and technologies that could support space communication networks - such as the Interplanetary Internet (IPN).

The IPN is a store-and-forward network of Internets in support of a deep space exploration that is often disconnected, has a wireless backbone with error-prone links and delays ranging to tens of minutes, even hours, even when there is a connection. Moreover, there are a number of physical phenomena in outer space such as solar storms and magnetic interferences which interrupt normal communication between spacecrafts. Moreover, the spacecrafts which are farther away from Earth have back dated technology than the ones launched recently. The existing terrestrial Internet and the TCP/IP suite will not be able to handle the constraints posed by such extreme conditions. We propose and study a future deep space network architecture which will survive such extreme conditions. The remote networks of the solar system such as the Earth based Internet and other planetary networks support various protocols and they hook up to the IPN backbone by choosing among mobile satellite gateways that would seamlessly convert between these protocols [1].

Delay Tolerant Networking (DTN) is viewed as an overlay network on top of such regional planetary networks. It incorporates a new protocol layer called as the bundle layer on top of heterogeneous region specific lower layers. The NASA Jet Propulsion Laboratory (JPL), has developed the Interplanetary Overlay Network (ION) to implement DTN in Interplanetary environments. It is open source, modular, easy to modify and we can also plug in our own routing protocol. It implements the Bundle Protocol as in [2] along with the CCSDS File Delivery Protocol (CFDP) [3] and the Licklider Transport Protocol (LTP) found in IRTF RFCs 5325 [4], 5326, and 5327. Three experiments on space DTN were recently conducted, namely - the UK-Disaster Monitoring Constellation (UK-DMC) Experiment (2008), the Deep Impact Network Experiment (DINET) (2008) and Experiment on-board the International Space Station (ISS) (2010) and all these experiments were done in space. Considering the critical issues and cost for space deployments and testing, we propose designs of IPN deployments on the Open Access Research Testbed for Next-Generation Wireless Networks (ORBIT testbed) at Rutgers University so as to examine the network and its operations more easily. Delay Tolerant Networking is the basis for the GSTAR routing approach [5] proposed for a future Internet in the MobilityFirst project [6].

II. Interplanetary Overlay Networks

The Interplanetary Overlay Network (ION) [7] is a product of the Jet Propulsion Laboratory (JPL) to implement DTN in Interplanetary environments and an overview of its functional dependencies are shown in Figure 1. ION uses a simple header compression scheme to improve transmission efficiency called the Compressed Bundle Header Encoding (CBHE) scheme and it is database centric unlike its predecessor DTN2 [8]. Fragmentation and reassembly is also well taken care of in the

978-1-4673-3122-7/13/$31.00 ©2013 IEEE

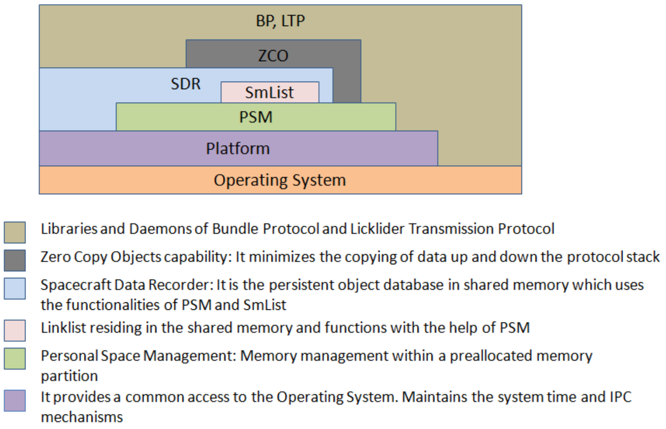

BP, LTP
ZCO
SDR
SmList
PSM
Platform
Operating System
Libraries and Daemons of Bundle Protocol and Licklider Transmission Protocol
Zero Copy Objects capability: It minimizes the copying of data up and down the protocol stack
Spacecraft Data Recorder: It is the persistent object database in shared memory which uses the functionalities of PSM and SmList
Linklist residing in the shared memory and functions with the help of PSM
Personal Space Management: Memory management within a preallocated memory partition
It provides a common access to the Operating System. Maintains the system time and IPC mechanisms

Fig. 1. ION Software Functional Dependencies.

ION infrastructure. To minimize transmission overhead and to accommodate asymmetric links in an IPN, ION wants to send downlink data in the largest possible aggregations and termed it as coarse-grained transmission. But again to minimize head-of-line blocking (delay in transmission of a newly presented high-priority item) and data delivery latency by using parallel paths (i.e., to provide fine-grained partial data delivery, and to minimize the impact of unexpected link termination), ION sends downlink data in the smallest possible aggregations and has termed it as fine-grained transmission. ION achieves both these functions at two different layers of the software stack - the Application layer and at the BP/LTP convergence layer. ION also supports Contact Graph Routing (CGR) which is a dynamic routing scheme used to compute the routes through a time varying topology of scheduled communication contacts in an IPN. Each node builds a contact graph data structure from the range and contact timeline entries and uses it to make the routing decisions. ION implements the concept of One Way Light Time (OWLT) which is the time taken for an eletromagnetic signal to travel one way between Earth and a spacecraft or some other celestial body. The node architecture and processing within the node has been further elaborated in [7].

III. ORBIT Testbed

The ORBIT testbed is a flexible wireless network testbed, supported by NSF and located at WINLAB in Rutgers University. It contains an indoor radio grid of 400 nodes arranged in a 20 by 20 matrix. Each ORBIT Radio Node is a real PC sitting at WINLAB with a 1 GHz VIA C3 processor, 512 Mb of RAM, 20 GB of local disk, two 100BaseT Ethernet ports, two IEEE 802.11 a/b/g cards and a Chassis Manager to control the node. For experimental setup we choose a group of nodes on the ORBIT testbed. Among them each single node is chosen and tailored to support the DTN architecture. The node runs Linux operating system on which we install the ION software module. A Ruby script file is executed which starts the ION application on all nodes at the same time taking a configuration file at each node as input.

IV. Network Scenario

In this section we discuss the problem of setting up a space network in a terrestrial environment and then varying the network parameters to gain an insight into the setup. We design and implement two different network scenarios to study the problem and measure the delay of transmitting a bundle from source to destination. The first scenario implements a network from Earth to Moon and the second scenario is a network from Earth to Mars. We keep alive both the scenarios for a specific duration and transmit bundles from the source to the destination while dynamically changing the topology from time to time.

A. Scenario 1: Lunar Mission

The Lunar network scenario is shown in Figure 2 where Node 1 represents the Mission Control Center (MCC) - an entity managing aerospace flight operations from the point of launch to the end of a mission. It is an Earth based node and all telemetric data is destined for this node after it is received by the DSN stations. Nodes 2 and 3 act as DSN stations at Goldstone and Madrid respectively. Finally, Node 4 represents the Shackelton station on the south pole of Moon. Any information either scientific data or image collected by Node 4 is sent through the DSN stations to Node 1. We consider One Way Light Time (OWLT) from Earth to Moon as 1.28 light seconds which is an average of the OWLTs for Earth to Moon communication at perigree (closest = 360000km) and at apogee (farthest = 405000km). The uplink datarate from Earth to Moon is set as 1 Kbps while the downlink datarate is 128 Kbps. The network remains alive for 10 minutes. The contact graphs over this period are shown in Figure 3. For the first 5 minutes the Madrid DSN station is not in sight with the Moon Shackelton station. It is in Line of Sight (LoS) with the Goldstone DSN station. The Earth rotates and after 5 minutes the Goldstone station moves out of LoS while Madrid comes into view. These two contact graph snapshots are fed into all the 4 nodes which gives them the network topology so as to construct the dynamic route, which a bundle should take during transmission.

B. Scenario 2: Mars Mission

The Mars network scenario is shown in Figure 4 and it consists of 6 nodes. Nodes 1, 2 and 3 are the same as Scenario 1. Node 4 represents a Mars orbiter called Mars Atmosphere and Volatile EvolutioN (MAVEN). It is part of NASAs' Mars Scout 2013 mission and we included it into our design as a futuristic vision. Node 5 represents the Mars Reconnaissance Orbiter (MRO). Finally, Node 6 represents a Mars rover called the ExoMars which is an autonomous six-wheeled terrain vehicle planned for a future robotic mission to Mars. Nodes 2 and 3 act as the major elements in the network to receive all telemetric signals sent from the Mars orbiters i.e., nodes 4 and 5 and then they are sent to the MCC Node 1. The OWLT for Earth to Mars communication is taken as an average of 324 light seconds. It varies as Mars moves to the farthest and closest to Earth. The OWLT between a Mars

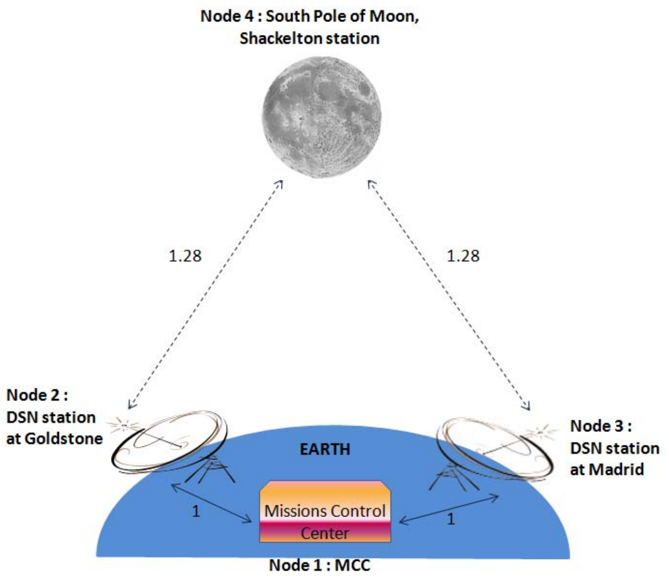


Fig. 2. Scenario 1 depicting the Earth to Moon communication configuration.

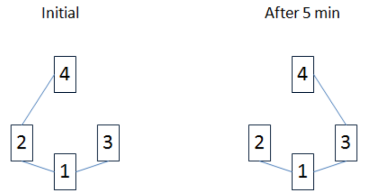


Fig. 3. Contact graphs for Scenario 1: Lunar Mission.

orbiter and the Mars surface is taken as 20 light seconds and the communication time between MCC and DSN stations is a maximum of 1 light second which is the normal terrestrial Internet speed. The communication datarates are as follows:

- Mars rover ↔ Mars orbiter = 16 Kbps
- Earth → Mars = 1 Kbps
- Mars → Earth = 128 Kbps

The network is alive for a duration of 2 hours. Figure 5 shows the contact graphs over this time period. Communication becomes a challenge as both Earth and Mars rotate and bringing the celestial bodies in LoS is difficult. Furthermore the Mars orbiters have their own orbital periods - MAVEN has an orbital period of 4.5 hours and MRO has an orbital period of 35 hours. Keeping these numbers and ephemeris data in mind we design the contact graphs so that within the small time duration of 2 hours each node has a chance to be part of a route for bundle transmission.

V. Experimental Results

A. Verifying Network Operation

In this section, we present the network statistics that help us to understand IPN and its implementation. We run Scenario 1 for a duration of 10 minutes over 2 tests. In the first Test A, 4 bundles are sent from the source Node 4 to the destination Node 1 during the first 5 minutes. After that the network is left idle for the next 5 minutes. In the second Test B, 7 bundles are sent from the source Node 4 to the destination Node 1. In the first 5 minutes we send 2 bundles and in the last 5 minutes we send 5 more bundles. After running the tests the statistics are drawn from both the networks (Earth - Moon, Earth - Mars) and they are shown in Figures 6 and 7 respectively. The network statistics gives the bundle transmission route and 8 different types of network activities as described in [9]. The figures verify the contact graphs of the Lunar Mission and the correctness of network operation. We also notice that for Test A the bundle processing at Node 3 is zero while the bundle processing at Node 4 and Node 2 are equal and higher than that required at Node 1. Similarly if we interpret the Figure

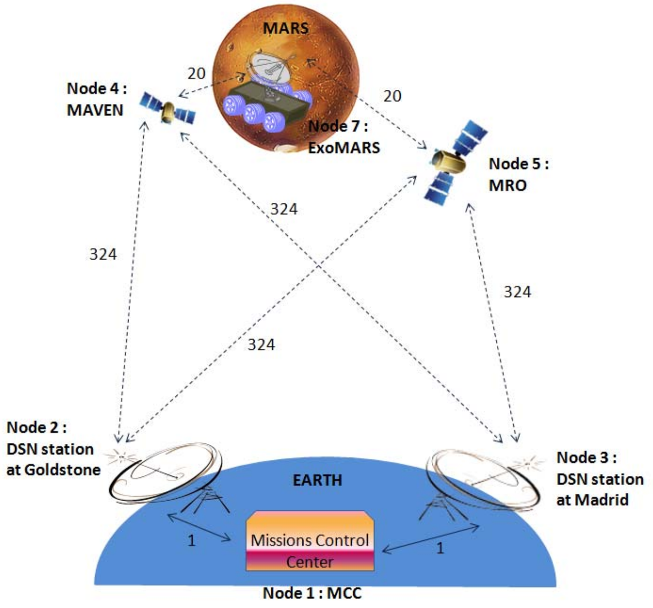


Fig. 4. Scenario 2 depicting the Earth to Mars communication configuration.

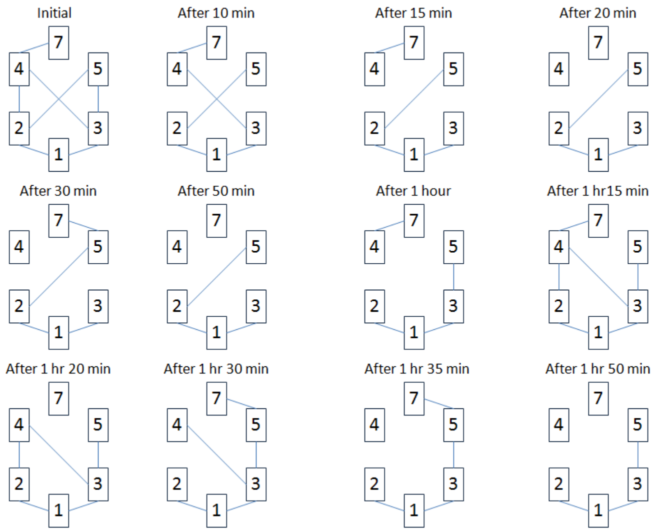
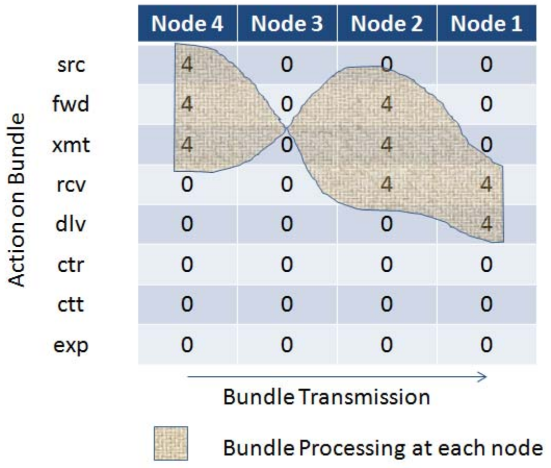


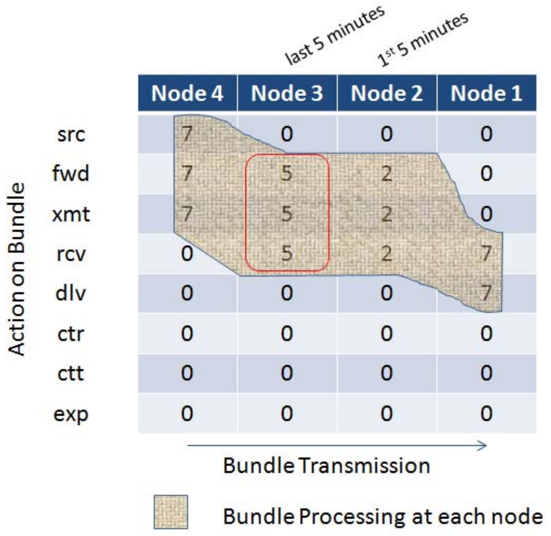
Fig. 5. Contact graphs for Scenario 2: Mars Mission.

for Test B, the red border around Node 3 indicates a higher processing requirement among the two intermediate nodes. Resources in space networks are very expensive. Hence we try to gain an insight into what factors lead to the exhaustion of a node. We can clearly see that a balance is needed on the number of bundles sent over alternative routes and it also requires a knowledge of the contact times over a given network. Appropriate load balancing with contact graph knowledge can increase the lifetime of the network. However, there are also other factors that might affect the balance such as spacecraft size, link capacity and asymmetric contact durations at different parts of the network, which we do not look into. In the next section we study the bundle transmission delay based on Scenario 1 and Scenario 2.



	Node 4	Node 3	Node 2	Node 1
src	4	0	0	0
fwd	4	0	4	0
xmt	4	0	4	0
rcv	0	0	4	4
dlv	0	0	0	4
ctr	0	0	0	0
ctt	0	0	0	0
exp	0	0	0	0

Fig. 6. Test A: Network statistics for transmitting 4 bundles.

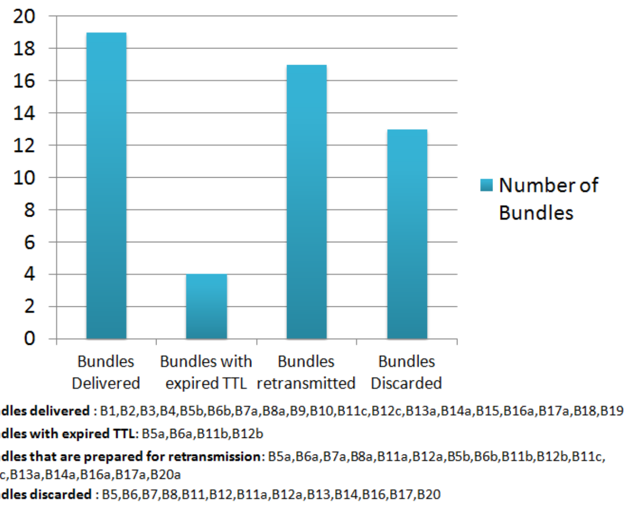


	Node 4	Node 3	Node 2	Node 1
src	7	0	0	0
fwd	7	5	2	0
xmt	7	5	2	0
rcv	0	5	2	7
dlv	0	0	0	7
ctr	0	0	0	0
ctt	0	0	0	0
exp	0	0	0	0

Fig. 7. Test B: Network statistics for transmitting 7 bundles.

B. Studying Bundle Transmission Delay

A bundle is transmitted from source to destination over a given route based on contact graph snapshot at that instant of time. In Delay Tolerant Networks a bundle may leave the source for its destination but there might be a disconnection in the middle of the network. In such conditions ION has the provision to store and take custody of the bundle at the local node until a connection is established. In our second scenario we try to study the delay in bundle transmission over a duration of 2 hours during which network partitions are created. The bundle lifetime or TTL (Time to Live) is set to a value of 600 seconds. A total of 20 bundles are sent over the duration of 2 hours and the bundles to corresponding time slots are shown in Table I. Each bundle is sent from the source Node 7 to the destination Node 1. After running the experiment we show the status of the bundles in the network along with their corresponding counts in Figure 8. We find that there are a total of 19 bundles that are successfully delivered to the destination. Due to the disconnection of the source node from the network in the last time slot as shown in the contact graph, the 20th bundle remains undelivered. Transmission of a bundle through the DTN can fail mainly due to two reasons - contact failure and custody refusal. Situations may arise where a contact between two nodes does not occur for a long time period during which the bundle TTL expires or a bundle cannot be transmitted within the set contact period of the two communicating nodes. For both the cases the bundle is removed from its outbound transmission queue and the Dynamic Route Computation Algorithm is re-applied to the bundle so that an alternate route can be computed. For our experiment bundles which undergo this kind of failure are given a small alphabet version (such as B5a) when its route is recomputed. There are a total of 17 bundles whose route needs to be recomputed. However, a node might also refuse custody of a bundle because it finds



Bundles delivered : B1,B2,B3,B4,B5b,B6b,B7a,B8a,B9,B10,B11c,B12c,B13a,B14a,B15,B16a,B17a,B18,B19

Bundles with expired TTL: B5a,B6a,B11b,B12b

Bundles that are prepared for retransmission: B5a,B6a,B7a,B8a,B11a,B12a,B5b,B6b,B11b,B12b,B11c, B12c,B13a,B14a,B16a,B17a,B20a

Bundles discarded : B5,B6,B7,B8,B11,B12,B11a,B12a,B13,B14,B16,B17,B20

Fig. 8. The status of the bundles in the network and their corresponding counts.

TABLE I

BUNDLES SENT AT DIFFERENT TIME SLOTS (IN HOURS:MINUTES) FOR SCENARIO 2: MARS MISSION.

0-10	**10-15**	**15-20**	**20-30**	**30-50**	**50-60**	**1-1:15**	**1:15-1:20**	**1:20-1:30**	**1:30-1:35**	**1:35-1:50**	**1:50-2**
B1	B3	B5	B7	B9	B11	B13	B15	B16	B18	B19	B20
B2	B4	B6	B8	B10	B12	B14		B17			

it impossible to forward the bundle. Such bundles are simply discarded, but discarding any such bundle that is marked for custody transfer will cause a custody refusal signal to be returned to the bundles current custodian. Again the bundle route needs to be recomputed.

Figure 9 shows the delay in transmission for each bundle. We perform two tests - Test A and Test B on Scenario 2. All parameters for both the tests are kept same. We get similar results except for a few differences which is accountable to the multiprocessing capability of a node leading to different bundle processing times. However, in both the tests we find that the bundles 11 and 12 take the maximum time to reach the destination. It is almost $1732.3/345.69 = 5.011$ times the nominal delay for the network. It is a consequence of these two bundles being repeatedly discarded at different times in different parts of the network. Figure 10 through Figure 13 demonstrates the delay in bundle transmission based on a number of different factors. We now use Scenario 1 to study the bundle transmission delay. Two new applications are developed, implementing delay calculation and bulk bundle transmission. They are named as *send* and *recv* for sending and receiving bundles respectively and are designed to take the source and destination eids as input. Calculating delay seems to be quite challenging since it requires the whole network to be synchronized. We achieve this by writing a script to synchronize the whole network through ORBIT console. As shown below we subtract the sum of the bundle's creation time (provided in the BpDelivery structure that gets populated when we call bp_receive function in the recv application) and 946684800 value (required to convert from the internal DTN Epoch 2000 time to Unix epoch time), from the current time at the moment bp_receive function returns (as provided by the getUTCTime function in the ici library of ION software module). This finally gives us the time taken by a bundle to reach from the source to the destination in seconds.

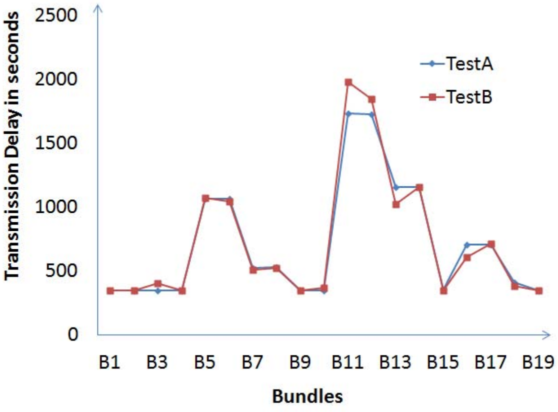


Fig. 9. Transmission Delay of Bundles through the DTN.

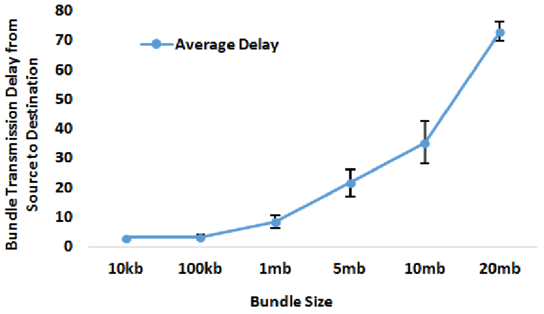


Fig. 10. Average Bundle Transmission Delay with varying bundle size along with 95% confidence interval.

In Figure 10 we plot the average delay of 4 tests - Test A, Test B, Test C and Test D with the same environmental parameters and topology while varying the size of the bundle from 10Kb to 20Mb. We can clearly notice the increase in bundle transmission delay time with increasing bundle size. Henceforth, we always take the average of 4 tests, to obtain a single delay value. This removes the unpredictability due to node multiprocessing that was observed for Test A and Test B in Figure 9.

In our next experiment we introduce a variation in Scenario 1 topology as depicted in Figure 11. An extra node - Node 5 and Node 6 is included in each of the routes, 4-2-1 and 4-3-1 respectively. In another variation we include 2 extra nodes in each route - nodes 5, 7 and nodes 6, 8 respectively. The contact graph routes remain the same as in Scenario 1 over the duration of 10 minutes. The plot in Figure 12 shows the delay in bundle transmission based on bundle size which is varied from 10Kb to 20Mb. It is seen that the bundle transmission delay gradually increases as more number of nodes are added to the topology. An average increase of 43.25% is noticed when we add 1 extra node to the route and about 87.92% when we add 2 extra nodes to the route. This clearly depicts the effect of increasing hops on the delay of bundle transmission. In order to evaluate the impact of the increase in number of bundles on the transmission delay, we repeat the experiment of Scenario 1 with varying number of bundles while keeping the bundle size fixed at 10Kb and 10Mb. Figure 13 shows us the data plot. We find that for 10Kb bundles the transmission delay and its variation is low till the number of bundles increase to 100. When the number

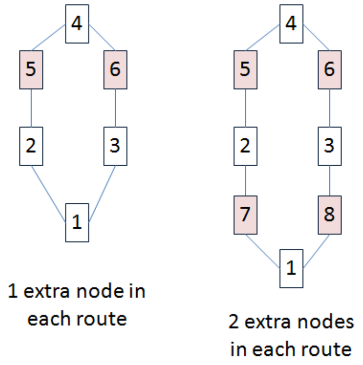


Fig. 11. Variation in Scenario 1 by adding extra nodes.

is increased to beyond 100 to 200 the delay shows a steeper rise. A similar behavior is also noticed for 10Mb bundles. This can be explained as a result of congestion in the network due to increase in data volume. Congestion in a DTN is the imbalance between data enqueuing and dequeuing rates that results in exhaustion of queuing (storage) resources at a node, preventing continued operation of the protocols at that node. ION implement methods to compute congestion forecast and admission control mechanisms. Whenever a congestion is predicted by ION it sets an alarm which prevents further data being pushed from the application layer. Data rate control at the link layer and revised contact plans can help to avert the anticipated resource exhaustion and in turn congestions. Furthermore, we notice that the delay takes a steeper rise for 10Mb bundles than for 10Kb bundles with increasing number of bundles. Hence, we can conclude that delay is affected by both the bundle size and the number bundles pushed towards the destination.

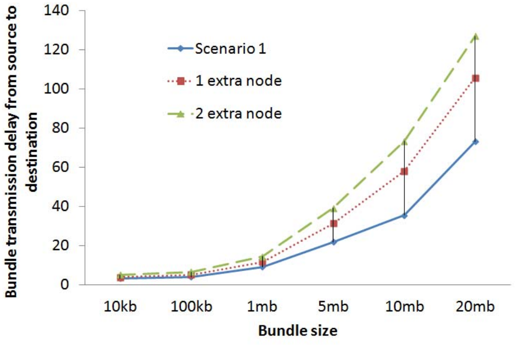


Fig. 12. Bundle Transmission Delay with varying bundle size along with the implementation of extra nodes.

VI. Conclusion

We propose an approach where we implement the Interplanetary Overlay Network (ION) software distribution on real time physical nodes on the ORBIT testbed. Two space network scenarios are designed which help us to emulate the Interplanetary Internet (IPN). We first test the experimental data to verify the correctness of the network implementation and also discuss the bundle processing required at each node. Resources in space networks are very expensive. This discussion gives us an insight into what factors lead to the exhaustion of a node. We next focus on the study of delay in bundle transmission. Disconnection in network leads to almost 5 fold increase in bundle transmission delay as bundles repeatedly get discarded. Finally we study the impact of increasing the number of bundles, increasing the bundle size and deploying extra nodes on bundle transmission delay.

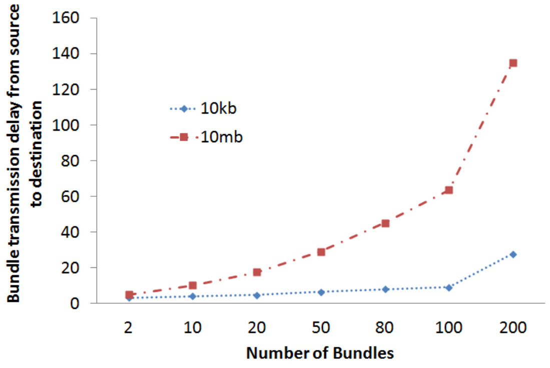


Fig. 13. Transmission Delay of Bundles through the DTN.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. (CNS-1040765) and a NASA Nebraska Research Grant. We thank Mr. Philip Tsao and Dr. Loren Clare (both at NASA JPL) for assistance with the ION software module and Mr. Ivan Seskar, Dr. Kiran Nagaraja and Prof. Dipankar Raychaudhuri (all at Rutgers University) for access to and help with the ORBIT testbed.

References

[1] J. Mukherjee and B. Ramamurthy, "Communication technologies and architectures for space network and interplanetary internet," in *IEEE Communications Surveys & Tutorials*, Jul. 2012.

[2] K. Scott and S. Burleigh, "Bundle protocol (BP)," IETF Request for Comments, RFC 5050, http://tools.ietf.org/html/rfc5050, Nov. 2007.

[3] NASA, "CCSDS file delivery protocol (CFDP) part 1: Introduction and overview," CCSDS 720.1-G-3, Green Book, http://public.ccsds.org/publications/archive/720x1g3.pdf, Apr. 2007.

[4] S. Burleigh, M. Ramadas, and S. Farrell, "Licklider transmission protocol (LTP) - motivation," IETF Request for Comments, RFC 5325, http://www.rfc-editor.org/rfc/rfc5325.txt, Sep. 2008.

[5] S. C. Nelson, G. Bhanage, and D. Raychaudhuri, "GSTAR: generalized storage-aware routing for mobilityfirst in the future mobile internet," in *Proceedings of the sixth international workshop on MobiArch*, ser. MobiArch '11. New York, NY, USA: ACM, 2011, pp. 19–24.

[6] D. Raychaudhuri *et al.*, "Mobilityfirst future internet architecture project," http://mobilityfirst.winlab.rutgers.edu/, Sep 2010.

[7] S. Burleigh, "Interplanetary overlay network: An implementation of the DTN bundle protocol," in *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE*, Jan. 2007, pp. 222 –226.

[8] K. Nichols, M. Holbrook, R. Pitts, K. Gifford, A. Jenkins, and S. Kuzminsky, "DTN implementation and utilization options on the international space station," in *SpaceOps 2010 Conference*, Apr. 2010.

[9] J. Mukherjee, "Routing over the interplanetary internet," Master's Thesis, Department of Computer Science and Engineering, University of Nebraska-Lincoln, Aug. 2012.