Some comments on Delay-Tolerant Networking for Space Communications (draft version)

V. Tsaoussidis, I. Psaras, C. V. Samaras, G. Papastergiou
Dept. of Electrical and Computer Engineering
Democritus University of Thrace
Vas. Sofias 12 Str., 67100
Xanthi, Greece
{vtsaousi , ipsaras, csamaras, gpapaste}@ee.duth.gr
COMputer NETworks Research Lab (COMNET)

http://comnet.ee.duth.gr

Email: comnet@lists.duth.gr

Abstract

We review and evaluate the main goals and requirements for Delay-Tolerant Networking, as well as the functionalities provided by the newly standardized Bundle Protocol. We identify and discuss a number of open issues, which pertain to the operational properties of the Bundle Protocol for Delay/Disruption-Tolerant Networking. Key issues that we attempt to adress are (i) the type of DTN architecture that is suitable for Space communications, (ii) the conflicting functionalities between IP and DTN, including routing, and (iii) the need for a space transport layer and its corresponding functionality.

1 Introduction

Space communications typically rely on scheduled links and require less sophistication from communication protocols. In this context, the link layer was the dominant layer for space communications; routing was never an issue; end-to-end reliability was frequently overlapping with reliability of a single hop; congestion and overflow were absent due to strict scheduling of communication activities and admission control; and the limited required sophistication was shifted to the application layer.

However, two new major properties have changed the spectrum of potential architectural choices for space communications: (i) the multihop architecture, which is required to reach deep space and (ii) the increasing number of alternative communicating paths that may be used to reach a single receiver. Along these two properties, the demand for interoperability among space agencies have also contributed towards the emerging field of *Delay/Disruption-Tolerant Networking* (DTN)[1] [4]. DTN, however, has evolved primarily as a technology for internetworking; although its main goal is to deal with long service disruptions and certainly this appears very similar, as an effect, to long propagation delays in deep space, there are some disimilarities. Perhaps the most important is the requirement of space applications to store and FORWARD data, instead of STORE and forward data, which seem to evolve as the design of choice for standard DTN. That said, a distinction is made here between Space DTN and Internet DTN until it will be proven that such a distinction is not necessary. Our work here contributes towards the direction of clarifying the goals and requirements of DTN for Space; contrasts the design perspective of Interent DTN; and identifies but leaves open a number of questions that call for further research.

The structure of the report is organized as follows: in Section 2, we provide a framework for Delay-Tolerant Networking. In Section 3, we discuss a number of open issues for Delay-Tolerant Networking in the form of questions and answers. In Section 4 we highlight some interesting points for the DTN architecture, and Section 5 concludes the report.

2 A Framework for Delay-Tolerant Networking

A generic **Delay-Tolerant Network** is determined to handle communications within a set of networked nodes/entities that suffer one or more of the following conditions:

1. They are too far-away from each other and therefore, the propagation delay of the connecting link is high.

---

[1] In the following, we consider that the term Delay-Tolerant Networking encompasses disruptions in connectivity as well.

2. They are energy-constrained and therefore, cannot transmit or receive data at any given time.
3. Their mobility pattern constraints ubiquitous connectivity.
4. They are connected through a highly error-prone link.

Therefore, a Delay-Tolerant Network incorporates proactive algorithms, mechanisms and protocols that are tolerant against sparse or intermittent connectivity, potentially huge propagation delays and high bit error rates.

By and large, the need for deployment of some form of Delay Tolerant Networking for Space, relies on three major arguments:

1. Deep space communications will involve several hops to reach their destinations. A direct and simultaneous connectivity among all hops is becoming more difficult; therefore, extensive storing capacity will be required.
2. Deep space communications may have the opportunity in the near future to take advantage of several alternative paths; therefore, end-to-end management will be required.
3. Inter-agency communication will soon need a common platform to allow for more natural communication that may replace a series of encapsulation and tunneling patches.

Having the goal clearly identified, the real issue regarding Delay-tolerant-networking for Space becomes its architecture. What protocols does it have to incorporate? How should it split the functionality among layers? How to avoid the overlapping functions of routing, reliability and naming/addressing? Eventually, the questions can be summarized in three major open issues that need to be elaborated in more depth and find a clear answer.

1. What form of DTN is required for Space?
2. What are the conflicts between DTN and IP?
3. Do we need a transport layer as well?

3 Open Issues on Delay-Tolerant Networking

3.1 What is the *proposed* versus *required* functionality for Space DTN?

The Delay-Tolerant Networking Research Group (DTNRG) [6] has proposed an architectural scheme to provide network connectivity in challenged environments: the DTN Architecture [4] and the accompanying Bundle Protocol [9] specification documents present a means for data communication on potentially heterogeneous networks characterized by high delays or disruptions. However, the DTN architecture alone cannot cope with these challenged environments. It essentially glues together dissimilar protocol stacks, but it relies upon underlying network services adapted to the special networking conditions.

We next summarize the main features of the Bundle Protocol [9]:

- *Store-and-Forward Message Switching*

The bundle protocol deploys store-and-forward message switching. The protocol data unit of the bundle protocol is the so-called *bundle*. More specifically, the bundle layer transforms each application data unit into one or more bundles.

- *Permanent In-Network Storage*

Permanent-storage devices provide data store for possibly long periods.

- *Custody Transfer*

Custody transfer allows the source to delegate retransmission responsibility to the next available node on the path to the destination.

- *Reliability*

Reliability is facilitated by the custody transfer feature. Nevertheless, only coarse-grained retransmission is optionally deployed. Reliable data delivery service is assumed to be provided by the underlying protocols.

- *Naming and Late Binding*

DTN nodes are identified by (at least) one text-string identifier, which is expressed syntactically as a Uniform Resource Identifier (URI). Late binding allows name-based routing, while destination name-to-address translation need not be performed at the source node.

- *Routing*

DTN provides routing across potentially heterogeneous networks. Routing computation algorithms remain a challenging task under investigation.

- *Fragmentation*

A large application data unit or bundle may be fragmented in order to utilize the contact periods more efficiently.

What form of DTN is required for Space?

DTN's storing functionality and its property to delegate custody seem to match two major requirements for space communications: one is the requirement for permanent storage due to long interruptions in space communications; another is the requirement for hop-by-hop reliable communication until the receiving end.

However, DTN is proposed as a global overlay architecture, which raises several further questions:

- What is the overhead cost for global convergence instead of a specialized architecture for space?
- What is the real benefit for a convergence between space and Internet technologies?
- Should the main property of DTN be "STORE and forward" or "store and FORWARD"?

Current trends in Delay/Disruption-Tolerant Networking follow the "STORE and forward" approach to bundle transmission. That is, once the bundle is generated at the source, it is transferred on a hop-by-hop basis until it reaches its destination. However, once a node accepts the custody transfer of an incoming bundle, it has to receive the whole bundle before it can forward it to the next node. Although this approach seems to be suitable for long-delay, intermittently connected links, it may also become an obstacle to fast end-to-end bundle transmission, from a resource-availability perspective. That is, provided that connectivity is sparse and intermittent, a node should exploit all available transmission opportunities. In that context, if the link to the next node is operational, then "store and FORWARD" may comprise a more elegant approach to bundle delivery. In other words, the receiving node stores the incoming bundle, *but* it may also begin transmission of the bundle-fragment that has already arrived, to the next hop, if this is feasible.

Moreover, DTNs may consist of relatively short-propagation delay regions, such as planetary networks. In this case, alternative paths may serve the bundle transmission process if the original link fails. In contrast, current practice (i.e., "STORE and forward") does not allow for bundle-fragmentation /re-fragmentation and autonomous routing on the fly. The "store and FORWARD" functionality, however, may require modifications to the Bundle Protocol.

3.2 IP, DTN or both?

One research direction for interoperable space communications favors deploying the Internet Protocol (IP) in order to interconnect space-based networking entities among themselves, and with the ground systems. In the simplest case, a single IP address can be assigned to a spacecraft, thus affecting only the interface from the Command and Data Handling system to the space link. A more advanced implementation is to configure distinct IP addresses for various subsystems and devices on-board the spacecraft. In either case, IP's global addressing can provide a common addressing scheme, which can be considered as the first step towards interoperable communications. If a modified IP could solve the routing and storage problem, and given that the naming/addressing scheme would have naturally found a global solution, the only remaining issue for DTN is the custody tranfer and the delay-tolerant transmission policies that needed to support space applications. That is, a single protocol such as LTP-T[12] or DTTP[10], for example, could provide the component missing from the spectrum of desired DTN functions. Note that such an approach is architecturally more elegant; the routing is managed at the network layer and is not unnecessarily hierarchical; the transport relies on routing below and therefore allows for flexible paths between the sender and the receiver; unlike a two-tier routing policy that allows for flexible routes only within the regions. Hence, the decision to reach a new arcitecture may require further justification.

Nevertheless, there exists a number of challenging problems regarding IP adaptation for space environments. IP may be considered as a delay-tolerant protocol since its forwarding functionality is not inherently constrained by time; its association with time constraints comes only as a side-effect of the limited buffering capacity.

Both IP and the Bundle protocol encompass routing tasks, which present increased complexity in intermittently connected networks with excessive propagation delays and moving network nodes. Thus, each approach will eventually have to devise equivalent methods for construction of routing tables, though from a different perspective. Bundle protocol hides the underlying protocol stacks upon which it operates, and requires routing information that connects bundle-nodes at the application layer. That does not preclude routing protocols at a lower layer. Instead, routing can actually occur at two levels: inside each autonomous system (assuming a set of heterogeneous networks), and also between the higher-level bundle-nodes. IP, on the other hand, lays a common addressing scheme that can seemlessly integrate differing networks. We consider that implementing delay-tolerant behavior for IP routers in space, is essentially a feasible task. Assuming a lower layer or connection management engine responsible for ceasing/resuming transmission, IP packets (like in the case of bundles) can certainly be stored in permanent storage, waiting for the connection to a destination peer to become available.

Given the prior knowledge pertaining to the relevant movement of space objects - and ground-based network nodes, a more-or-less static routing system is expected to govern space internetworking. At least, ground stations (either on Earth, Moon, Mars, or elsewhere), and relay satellites with predetermined orbits, offer a deterministic routing framework that can serve as the backbone for space data communications.

That said, the appropriate layer for deployment of the routing functionality may be different for scheduled and opportunistic or random contacts and correspond to the requirement for flexible or predetermined paths[2], respectively. In case of scheduled contacts, the routing functionality may be implemented at the higher layers, since the bundle route is known and predetermined. In contrast, opportunistic contacts need to rely upon a flexible routing scheme below the Bundle layer in order to be able to alter between various alternative paths. That is, once a bundle is generated, it either needs to know its route to the destination from the beginning (i.e., scheduled contact) in order for the bundle layer to route it accordingly, or the unknown destination has to be determined on the fly by lower layers. Alternatively, the Bundle Protocol will need to be modified in order to fragment bundles into smaller data units and include routing information in each fragment. By the same token, the routing policy will very much determine whether the "STORE and forward" or the "store and FORWARD" transmission policy will be adopted.

3.3 Do we need a Transport layer?

A transport layer cannot be naturally replaced by application-level functions; and cannot be replaced by link layer functions. At the link level, end-to-end reliability cannot be provided with multiple hops and flexible paths; when error correction or connection fails at some link, the task fails as well, even if there is alternative path. Also, enhanced sophistication at the link level cannot be presently incorporated.

At the application layer, the potential advantage of packet-oriented feedback for scheduling the retransmissions is lost and we necessarily deal with messages. See DS-TP as an example protocol that takes advantage of packets.

---

[2] Even if the route is scheduled *a priori*, service interruptions may exist due to unpredictable events; in that case, there may exist alternative communication paths, which could not be exploited. In the same context, the term *certainty* is associated only with the known descriptors of a communication event.

A transport layer can assist end-to-end reliability, bandwidth exploitation for the whole end-to-end path and, in the future, congestion control as well. The transport layer handles reliability on a per packet basis, whenever feedback is gathered from the receiver – see DS-TP [8] as an example of open-loop protocol regarding the transmission scheduling, but closed-loop regarding the retransmission scheduling.

That said, the essential difference between the transport and the link layer of the networking protocol stack is packet-based congestion and flow control. Presently, congestion control in DTNs mainly refers to storage exhaustion rather than the (temporary-) buffer overflow. Flow control, on the other hand, regulates the amount of data that the sender is allowed to insert into the network, so that it doesn’t overflow the receiver's buffer. Thus, flow and congestion control in DTN environments acquire the same meaning.

Last, but not least, trasport is a misunderstood layer; quite often is confused with application layer protocols. In this context, DTN may require two types of “transport” protocols: one, hierarchically at a higher layer, to manage DTN transport service and eventually replace DTN functionality when DTN cannot operate efficiently or it is absent (such protocols are the DTTP [10] and the LTP-T[12]); and another to handle end-to end transmissions over multiple hops. LTP[11] or DS-TP[9] may be such canditates entitled to an autonomous transport service..

4 Discussion

Routing functionality may either be implemented below the Bundle Layer or alternatively the Bundle Protocol will need operational and structural modifications. In particular,

- the bundle layer will need to fragment bundles into smaller packets/segments according to the path Maximum Transmission Unit (MTU), and
- the bundle header will need to be included in all packets/segments produced by the bundle layer.

Otherwise, individual packets will not include enough information in order to be dynamically and autonomously routed to the destination. Unfortunately, such an approach will increase the Bundle Protocol's byte-overhead.

In essence, the bundle layer provides a common method for interconnecting heterogeneous gateways, and overcomes communication disruptions by employing store-and-forward message switching based on non-volatile storage. DTN introduces another layer of abstraction offering applications an interface to cross over a number of differing networking environments. Also, the custody transfer capability facilitates data trasports exploiting in-network storage. Nevertheless, it is neither obvious nor assured that DTN offers better performance throughput-wise, compared to other alternatives. However, implementing DTN requires adding relevant components to the target network infrastructures and induces overhead, which has to be quantified on the basis of experiments and simulations.

Finally, we note that DTN does not eliminate the need for a space agency to cross-support others, so that they can share their facilities. DTN is layer-agnostic and thus requires an appropriate convergence layer, in accordance with the underlying protocol stack. Should the offered services by the lower-level layers not provide the desired functionality for DTN applications, the convergence layer should implement the missing functionality. Also, the bundle protocol introduces byte-overhead which is appended to the application data units during bundles construction.

5 Conclusions

Delay/Disruption-Tolerant Networking is a relatively new but promising research area. DTN departed from the need to incorporate delay tolerance into the Internet; and emerges as a solution to deep space communications - this is admitedly a wide area of applicability with many inherent diveristies. We identified three major issues, of strategic importance for selecting the architecture of choice for Space DTN, that need to be clarified prior to standardizing the DTN architecture and protocols.

References

[1] I.F. Akyildiz, O.B. Akan, and J. Fang. Tp-planet: A reliable transport protocol for interplanetary internet. *IEEE Journal on Selected Areas in Communications (JSAC)*, 22(2):348–361, 2004.

[2] Scott Burleigh, Esther Jennings, and Joshua Schoolcraft. Autonomous congestion control in delay-tolerant networks. In *Proceedings of AIAA 9th International Conference on Space Operations (SpaceOps), Rome, Italy*, June 2006.

[3] L. Wood et al. Saratoga: A scalable file transfer protocol, Internet Draft, Oct 2007.

[4] V. Cerf et al. Delay tolerant network architecture, RFC 4838, April 2007.

[5] Consultative Committee for Space Data Systems. Space Communications Protocol Specification (SCPS) - Transport Protocol (SCPS-TP), CCSDS 714.0-b-2. Blue Book. Issue 2. Washington, D.C., Oct 2006.

[6] Delay Tolerant Networking Research Group. http://www.dtnrg.org/.

[7] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *Proceedings of SIGCOMM 2004, Oregon, USA*, Aug 2004.

[8] Ioannis Psaras, Giorgos Papastergiou, Vassilis Tsaoussidis, and Nestor Peccia. DS-TP: Deep space transport protocol. In *Proceedings of IEEE Aerospace Conference 2008, Montana, USA*, March 2008.

[9] K. Scott and S. Burleigh. Bundle protocol specification, RFC 5050, Nov 2007.

[10] C. V. Samaras and V. Tsaoussidis, “A Delay-tolerant transport protocol for Space Internetworks", Technical Report: TR-DUTH-EE-2008-3.

[11] M. Ramadas, S. Burleigh, S. Farell, Licklider Transmission Protocol – Specification, <draft-irtf-dtnrg-ltp-06.txt>, April 2007.

[12] S. Farrell, V. Cahill, “LTP-T: A Generic Delay Tolerant Transport Protocol”, Technical Report: TCD-CS-2005-69.