

**DYNAMIC COMMUNICATIONS FOR SMALL SATELLITES USING
DISRUPTION TOLERANT NETWORK CONCEPTS**

N. Giuditta, E. Gill, B. Fernández⁺, D. Isaac^{*}

Chair of Space Systems Engineering (SSE), Faculty of Aerospace Engineering, Delft University of Technology,
Kluyverweg 1, 2629 HS Delft, The Netherlands, N.M.Giuditta.Villamil@student.tudelft.nl.: +31 (15) 27 87458, fax:
+31 (15) 27 85322

Chair of Space Systems Engineering (SSE), Faculty of Aerospace Engineering, Delft University of Technology,
Kluyverweg 1, 2629 HS Delft, The Netherlands, e.k.a.gill@tudelft.nl, ph.: +31 (15) 27 87458, fax: +31 (15) 27
85322

⁺DEIMOS Space, Ronda de Poniente 19, Edificio Fiteni VI, 28760 Tres Cantos, Madrid, Spain,
beatriz.fernandez@deimos-space.com, ph.: +34 91 806 34 50, fax: +34 91 806 34 51.

^{*}SES Astra TechCom, L-6815 Chateau de Betzdorf, Grand Duchy of Luxembourg, dan.isaac@ses-astra.com,
ph.: +352 624 285 181, fax: +352 710 725 575.

Abstract

New network technologies are providing interconnectivity in areas previously unheard of. One of these novel technologies, named Disruption Tolerant Networking (DTN), shows promise for the space industry. In order to study its suitability for University projects, a model of a University space mission has been created using network simulation software. This offers an insight into the characteristics, opportunities and challenges of this new technology.

Disruption Tolerant Networking is a pioneering approach to network-based communications, seeking to resolve technical issues for nodes that lack continuous connectivity, which current Internet protocols cannot handle. The technology is currently being tested by several research bodies, and offers great potential for interoperability over heterogeneous networks.

This paper focuses on the field of small satellites and the consequences of the future inception of DTN-based communications in upcoming space missions. Possible advantages include increased robustness to delays inherent to space communications as well as performance improvement, with little or no change to current communications infrastructure.

To test this assertion, the FAST mission has been chosen as a case study. This mission is a collaboration between the TU Delft and Tsinghua University and it is composed of two small satellites tasked with performing an analysis of global aerosol data in the atmosphere. After using a simulated scenario, experimental results show the positive effect of the implementation of DTN protocols to perform payload data downlink in heavily disrupted communications environments, while pointing out areas where future work is needed to improve this investigation.

BACKGROUND

The advent of the TCP/IP Internet model has pushed the boundaries in network communications standardization. However, there are a number of networks for which the requirements or assumptions that are valid for TCP/IP do not hold true: these are the so-called *challenged* networks. Their characteristics (primarily intermittent connection and long delays) do not allow these

standards to be applied effectively, with the consequential degradation of data delivery. Emerging from investigations into these challenged networks (such as the IPN by Cerf et al.^[1]), Fall proposed in 2003 to adapt new ideas to terrestrial networks and coined the term “Delay-Tolerant Networking” and the DTN acronym, in a paper that marked the official birth of this concept.

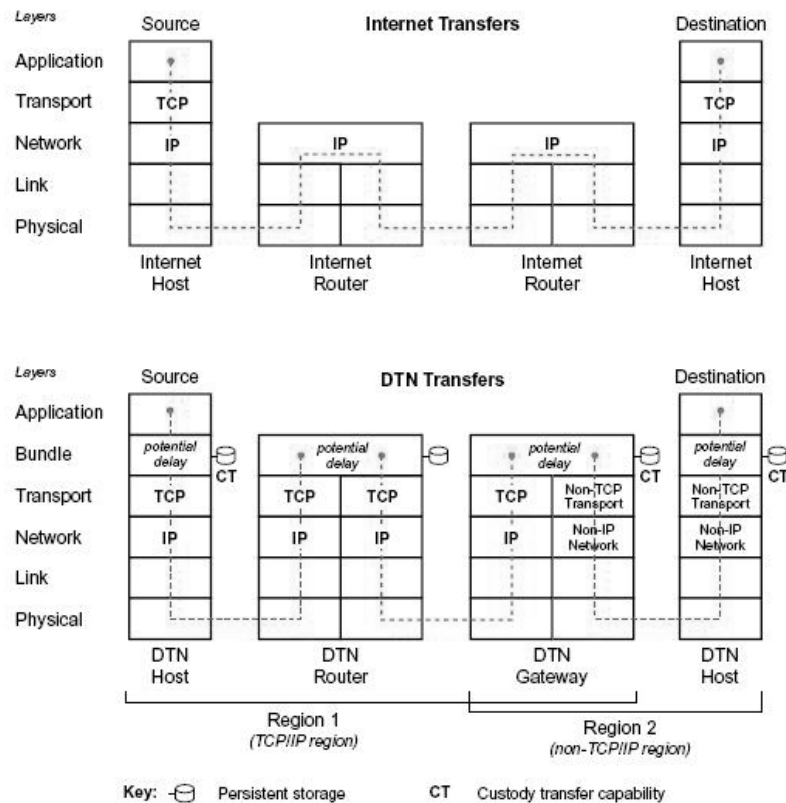


Figure 1 - Differences between TCP/IP and DTN model

Fall proposed to create an overlay architecture, which instead of substituting the existing network protocols, it supports their interoperability by accommodating long delays; an architecture that could run on top of current Internet protocols or any specific ones and could create a higher-level group, or “network of regional networks”^[2]. The principal goals of DTN are to support interoperability over heterogeneous networks while dealing with delay or disruption of the connections. The main technique to achieve this lies in using store-and-forward methods. This way, data packets are not directly sent over the network from source to destination. Instead they are moved between storage nodes, following a route leading to their desired destination. When a connection is not available (delayed or disrupted), the node stores the data packet and waits until the link is re-established to send the information.

This technology has quickly gained a lot of interest and numerous researchers and companies are pushing it forward, suggesting a variety of applications, from water quality sensor networks in lakes^[3] to providing connectivity in inhospitable regions^[4].

In this paper we take a look at the latest advances in the field and how this concept can be applied to the emerging topic of small satellite communications. Space is a perfect example of the disadvantages encountered by challenged networks: An unforgiving environment, small windows of communications with the ground segment, and a limited network.

The pairing of these spacecraft with DTN techniques shows promise in terms of testing new networking capabilities directly on missions while reducing risks: A number of past space missions have used small satellites working collaboratively to accomplish a common goal (i.e. ESA’s Cluster). This trend continues with “collaborative” missions such as LISA and Darwin. DTN protocols can not only enhance the communications capabilities of satellite constellations, but also offer higher performance than ad-hoc protocols when connecting to heterogeneous ground control networks.

However, the relative novelty of DTN means that there is not yet a standard protocol accepted by the networking community, rather “a range of possible solutions, each of which will be suitable for use in

different environments”, as stated by Farrell and Cahill^[5]. Nevertheless, there have been a number of Internet Drafts published, dealing with different aspects of DTN, and one of these presents the “bundle protocol”^[6], which appears to be the leading candidate for standardization, under development by the DTNRG. This protocol attempts to implement store and forward message switching overlaying a new protocol layer, called the “bundle layer”, between the application layer and the rest, tying together heterogeneous networks so that application programs can communicate between themselves. Bundles are the equivalents of IP data packets, and they attach a header (similar to TCP), extending the hierarchy of the data packet. This means that the bundle layer creates a new intermediate step in the processing of the data, including the necessary information to allow a DTN to correctly transmit the object.

The most advanced implementation of the bundle protocol today is Saratoga^[7], a file transfer protocol developed by Cisco Systems, SSTL and NASA. It has been operational since 2004^{[8][9][10]} with the DMC (Disaster Monitoring Constellation) series of satellites, first with the UK-DMC and most recently with the Deimos-1 (Fig. 2), the newest member of the constellation (launched on July 29th, 2009). Saratoga is designed with performance as the main driver considering that the satellites in the DMC can only enjoy brief periods in connectivity. As such, the protocol sends as much data as possible during these windows of opportunity and then checks if the data bundles have arrived correctly.



Figure 2 - Deimos-1 First Image, Delivered by Saratoga

In parallel to Saratoga, the Licklider Transmission Protocol^[11], or LTP, has also been developed. This protocol applies strictly to point to point communications, so there are no routing issues to consider. Data is simply transmitted between two peers with no intermediaries concerned. Although

originally designed for high-delay cases (deep space communications), the protocol is also useful for terrestrial applications, and could be implemented in sensor networks. There are a number of implementations currently being investigated, but, unlike Saratoga, no thorough testing has been performed in space environments.

RELATED WORK

Similarly to the somewhat chaotic development of DTN protocols, the lack of a standard means that the simulation has followed diverse paths to arrive to conclusions. Most researchers have opted to use existing networks simulators (OMNET++, ns-2) and then wrote DTN-specific modules, while others have designed a simulator entirely devoted to running DTN messaging simulations with different routing protocols (ONE^[12]). It is possible to find implementations ranging from Java-based solutions used for urban transport scenarios, to C++ modules and TCP scripts used with the ns-2 simulator^[13]. However, since this project focuses on small satellites, we have concentrated on simulations using DTN in space scenarios.

Most of the work performed in this area follows two directions: either comparing DTN solutions to TCP-adapted ones, or trying DTN approaches for satellites and seeing how they behave. For the first option, Caini et al.^[14] tested and compared the bundle implementation versus adapted TCP protocols in a scenario mixing terrestrial and satellite connections. They point out the advantages and disadvantages of this approach, namely the increase in performance and robustness, against the overhead traffic introduced due to the addition of a new header to the data. Their simulations show that in terms of performance, DTN rivals or surpasses adapted-TCP while offering greater robustness.

On the other hand, Shahriar^[15] takes the Saratoga protocol and simulates its performance using ns-2, in order to evaluate NEMO (NEtwork MObility) protocols for space networks, showing that Saratoga's throughput is not affected by low uplink bandwidth (unlike TCP, for example). Another simulation of the bundle protocol is presented by JPL researchers^[16], with the objective of using it with spacecraft flight software, in order to interoperate with a terrestrial peer. The LTP is also put through its paces thanks to Farrell and Cahill^[17], who create a simple model of the Deep Space Network to transmit data back to Earth coming from simulated nodes in Mars and Jupiter, presenting a generated schedule of transmissions and the latency for a sensor in the network.

SIMULATIONS

Considering the above, it is clear that DTN technology offers interesting possibilities for small satellite missions operating in degraded communications environments. In order for those possibilities to become certainties, it is necessary to perform simulations of DTN protocols on suitable University-related scenarios. One of these scenarios is the joint TU Delft – Tsinghua University FAST (Formation for Atmospheric Science and Technology Demonstration), an Earth observation mission which involves formation flying with two different spacecraft working together to analyse aerosol formation.

First, a general description of the FAST project will be given, so as to define the best approach to perform the desired tests and to find out the characteristics needed to allow for a correct modelling of the mission, which will be the following part, delving deeper in the software used to undertake the simulations. Then, the methodology and the assumptions for the simulations will be identified. Then the results are presented, with a brief analysis afterwards. The paper concludes with suggestions for the direction of future research.

Case Study: FAST Mission

FAST^[18] (Formation for Atmospheric Science and Technology demonstration) is a collaboration between Delft University of Technology in the Netherlands and Tsinghua University in Beijing, China. It is a formation flying mission consisting of two micro-satellites: FAST-D (Delft) and FAST-T (Tsinghua), designed to study atmospheric aerosols and seasonal variations of height profiles in the cryosphere using spectropolarimeter and altimeter payloads on both spacecraft. Other objectives include the demonstration of autonomous formation flying using various communication architectures with distributed propulsion systems and MEMS technology, as well as to serve as an educational platform for both universities. Mission lifetime will roughly last for two years, and both satellites will be launched into a sun-synchronous orbit (SSO approximately 650 km), working in an along-track formation, with a variable separation, as seen in Fig. 3.

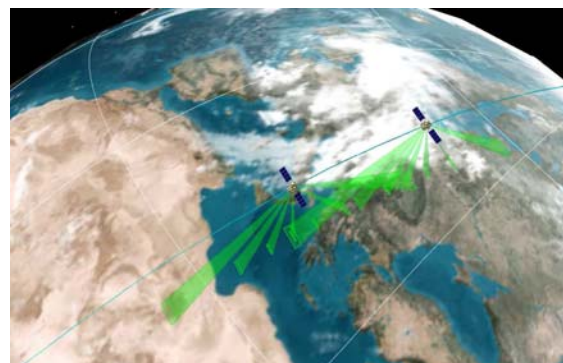


Figure 3 - Graphical representation of FAST orbit

The satellites will be similar in design, although their payload will be different. FAST-D will carry two scientific payloads: the mentioned spectropolarimeter, called SPEX, and a miniature laser altimeter called SILAT; while its Chinese counterpart, based on the Tsinghua-2 spacecraft design, will also carry SPEX and a radar altimeter. The correlation of altimeter data with aerosol measurement made by SPEX will allow an improved characterization of the indirect aerosol forcing which is currently the most uncertain contribution in the climate forcing bureau. The spacecraft are small and box-shaped, not exceeding 50 kg, and will perform formation flying aided by GNSS receivers onboard, upgraded by relative navigation data exchanged between the satellites through an inter-satellite link (ISL). Formation and AOCS is achieved by means of a cold gas propulsion system capable of providing a total delta-V of 12 m/s.

The selection of FAST for simulation, over other University projects, was done based on the following criteria: it offers a suitable scenario for our simulations thanks to its relative simplicity in terms of networking (2 nodes and a ground station); there is an ISL to establish a connection between the devices and most importantly, the space segment will have an open architecture, that is, complementary satellites from other nations or institutions can join the formation or train in the future. This makes FAST the kernel of the first microsatellite train in space, and the perfect target to implement communications protocols which improve interconnectivity between the different satellites in the mission.

Simulation Set-up

The simulation platform is a Pentium 4 desktop with OMNET++ software, which is a component-based, modular and open-architecture discrete event network simulator, released under an Academic Public License. It was chosen due to its extensive documentation and its Eclipse-based IDE, which simplified the learning curve and allowed for easier development and debugging. Modelling in OMNET++ is done using a combination of C++ and NED (Network Description) files. C++ is used to describe the specific behaviour of every node in the network, with the NED file showing the overall topology of the model.

In the case of the FAST mission, the topology was fairly straight-forward following the specifications of the scenario and making some assumptions where there was a lack of clarity, due to the early phase of the project. For example, two ground stations are modelled, even though this is not clear yet. These ground stations are named “Delft” and “Tsinghua”, and are interconnected between themselves. At the same time, each center is linked to its FAST spacecraft, “FAST-D” and “FAST-T” respectively. The last nodes to be included are those referring to the payloads: SPEX, SILAT and the unnamed radar altimeter in the Chinese satellite. Hence, “FAST-D” is connected to “SPEX-d” and to “SILAT”, and “FAST-T” is connected to “SPEX-t” and to “Radar-alt” forming an eight-member network, as seen in Fig. 4.

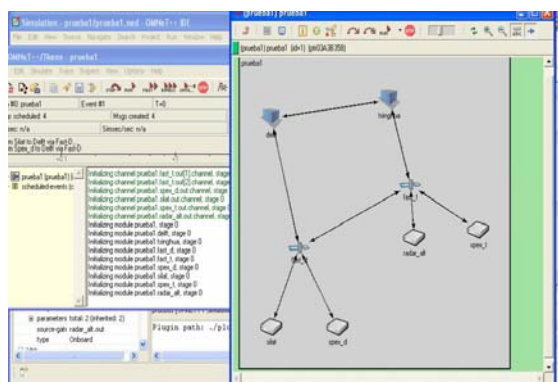


Figure 4 - Network topology model of the FAST mission

The next step lies in defining the parameters of each node in the network, in order to simulate the overall behaviour. To do this, first a scenario for the mission needs to be developed based on an assumption of the routine activities of the FAST satellites included in the model. Based on the specifications seen in the previous chapter, a

realistic example would be the following: With both satellites orbiting at 650 km altitude, separated by approximately 800 km, the payloads onboard start working. They generate a number of data packets which are transmitted to the communication subsystems of FAST. The satellite stores the data (“custody transfer”) until the ground station is in view and then downlinks the data, sends a response to the payload confirming the reception and delivery of the previous data packet, and asks for the next one.

Based on the orbital parameters, the FAST satellites would follow the same ground track, with roughly 15 revolutions per solar day. The preliminary data output of the payload onboard FAST, according to current estimations, is in the order of 43.2 Gbit/day for the SILAT and 16.6 Gbit/day for SPEX. Assuming data output of SILAT is equal to the presently unknown radar altimeter on FAST-T, and lossy compression methods which could account for an 80:1 compression ratio, the total data budget for onboard payload is approximately 93.5 MB/day, which is an acceptable amount to be downlinked during the different passes over the ground station each day.

In order to have more control over the data packets being sent, not only the different nodes are defined, but also the structure of the datagram. This is necessary to ensure the correct routing of the packet. Each one is identified by its source and given a destination at its creation, which is then used by the rest of the members of the network when forwarding the packets. Extra parameters have also been created to allow for the compilation of statistics. Also, all data packets which arrive at Tsinghua ground station are then forwarded to Delft, this is done to simulate the data sharing between the ground nodes of the network.

Delay is implemented both in the space segment and in the downlink to the ground station. A random value is first selected to account for data gathering by the various payloads. Then the data packets are stored in the satellite’s storage module, waiting for the establishment of the connection with the ground station. Once the data is sent, the on-board computer of FAST sends an “acknowledgment” message to the payload (as seen in Fig. 5), so they can send mission data again. This is done to prevent any possible overloading of the storage memory, dividing the data packets in several pieces which will then be assembled at the ground stations. For now, disruption has not been modelled, although it should be the next step in the project.

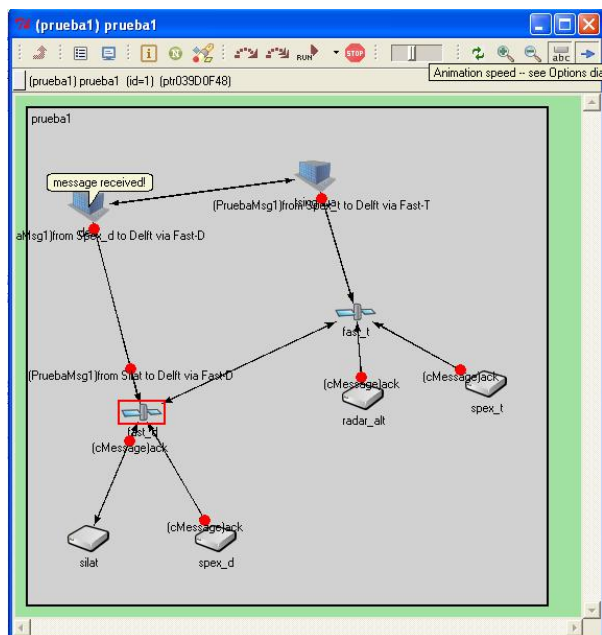


Figure 5 - FAST simulation in the middle of its run

The simulation output was oriented to help understand the behaviour of the network when using DTN principles. Since this is not a comparison between protocols, but an investigation regarding DTN implementation, the results obtained are related to the message schedule of the data packets, and the latency experimented from its creation at the various payloads until its arrival at the Delft ground station. Another parameter deemed interesting was the number of hops that each message experimented until its arrival; however, the limited size of the network did not account for noteworthy variations and has been discarded in the final analysis.

Results and Analysis

Figure 6 shows the message schedule of the network over 1 day of activity. The X axis represents the simulation time, in seconds, and the Y axis represents the state of the message, separated by their origin (which payload created the message):

1. Data packet is generated
2. Data packet is forwarded through FAST satellite
3. Data packet arrives to Tsinghua ground station
4. Data packet arrives to Delft ground station

The illustration shows the average life of a data packet from its creation at the payload modules to its arrival at the ground station. At the beginning, all the packets are clumped together but as soon as the random delay in the data gathering starts to have an effect, their generation is changed until arriving to a more constant state (the difference between 40000 sec and 80000 sec is much less noticeable than in the first half). The most important delay, of course, lies in the establishment of the link between the satellite and ground, but this issue is inherent to the case. A possible solution would be to recognize which satellite is closer to its ground station, and sending the data packet over inter-satellite link. However, the ISL configuration planned for the FAST mission allows for low data-rates (i.e. telemetry) and is not suitable for large data transmissions.

Once the message schedule was known, it was possible to extract another important parameter regarding communications in the FAST mission, which is the latency, or the amount of time it takes for a data packet to arrive at its destination. In this case, the origin would be one of the different payloads and its target would be the Delft ground station. In order to put these values in context, ENVISAT can be used as a starting point. The Earth-observing satellite, a landmark mission for ESA, shares similar orbit characteristics with FAST (sun-synchronous, 800 km altitude) and their payload data products exhibit a minimum latency of approximately 3 hours^[19]. Figure 7 shows the latency of a number of packages sent during a normal operational day for FAST. Results show an average lag of 107.41 minutes in obtaining payload information, which is a good result taking into account the orbital period of the satellites, which is approximately 89 minutes. Surely these values will have to be re-evaluated once there is more information regarding the data handling and processing of the payload modules, since it has been assumed that they work quite regularly. On the other hand, different processing could yield better results in latency for certain packets, since the network could be configured to check whether the ground station is in view at the moment of packet delivery to the FAST antenna, instead of waiting for the full revolution to complete, as it has been done in the simulation.

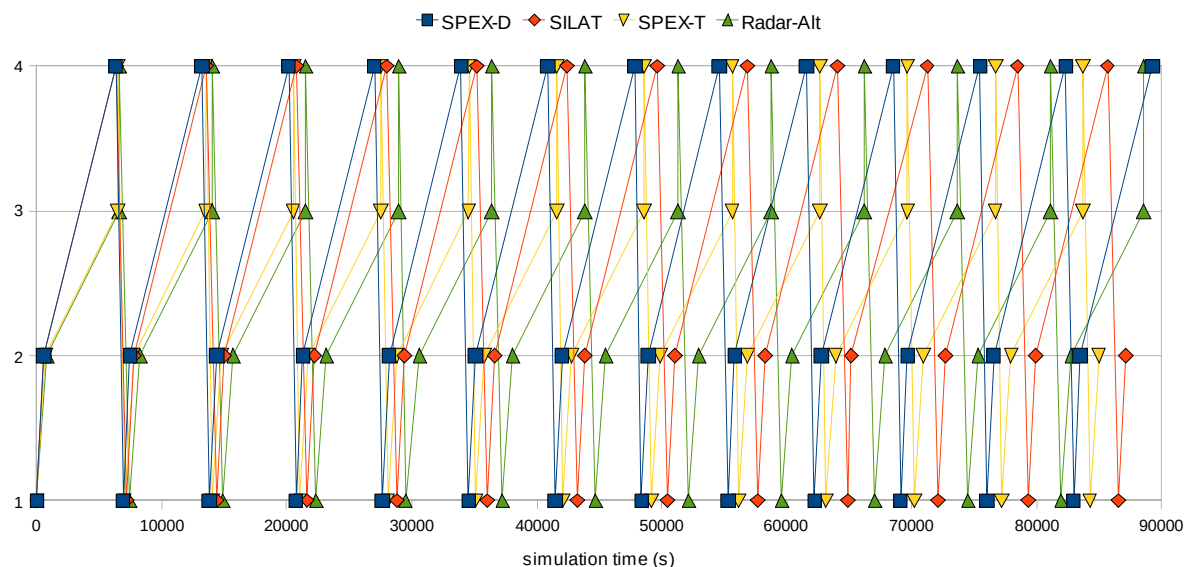


Figure 6 - Message schedule of FAST Network (1 day)

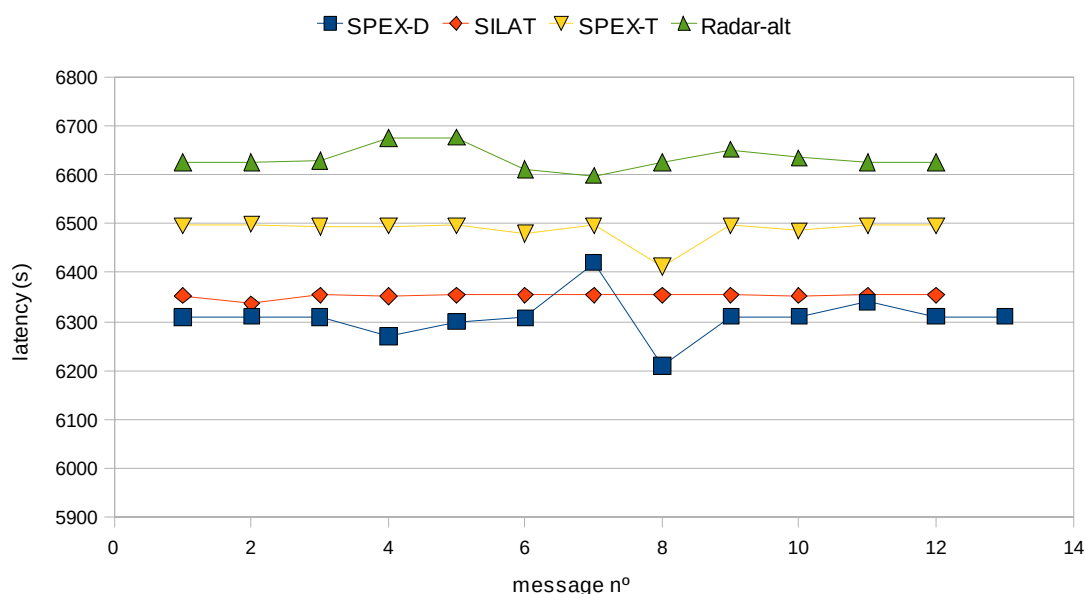


Figure 7 - Time elapsed from message generation to arrival at Delft

FUTURE WORK

Future work will be devoted mainly in implementing more and more characteristics of DTN protocols, such as reliability/robustness measures, and testing them against random errors in the simulations. The final objective is to simulate a full implementation of a protocol, either BP or LTP, and simulate them against other available techniques, such as adapted TCP protocols, in order to arrive to a reasoned

conclusion regarding the suitability of the inclusion of these protocols in future TU Delft missions.

Another direction for future research will deal with modelling more complex missions, which may offer different responses to the simulation of DTN techniques. An example would be the QB50 mission^[20], a proposed network of 50 Cubesats tasked with different research objectives regarding the lower thermosphere and spacecraft atmospheric re-entry.

CONCLUSIONS

We have modelled a University space mission to understand the consequences of implementing DTN techniques in the handling of onboard and space-ground data communications. The objective using DTNs is to create an easily accessible network in space. Current and future TU Delft-related missions will involve several satellites working together. Hence, communication protocols which improve interconnectivity and performance are of great interest. A case study has been simulated using OMNET++ software, to observe the behaviour of each node within the network, as well as the orbital characteristics of the mission, using an abstraction of a DTN protocol. Preliminary results show low values on latency, while offering an insight on how the message schedule will develop.

However, in order to gauge the suitability of DTN techniques, more thorough testing is needed. Future work will involve implementing more features of these protocols in order to have a complete prototype which can be compared to present solutions. Furthermore, we intend to introduce more complex scenarios, where the unique routing possibilities of this technology can be better highlighted.

REFERENCES

- [1] V. Cerf et al., “*Delay-Tolerant Network Architecture*”, IETF RFC 4838, informational, April 2007.
- [2] R. Warthman, “*Delay-Tolerant Networks (DTNs) – A Tutorial*”, Warthman Associates, March 2003.
- [3] S. Farrell, “*Sensor Networking with Delay Tolerant – Quick Overview*”, Department of Computer Science, Trinity College, December 2006. Website: <https://down.dsg.cs.tcd.ie/sendt/>
- [4] M. Demmer and K. Fall, “*DTLSR: Delay Tolerant Routing for Developing Regions*”, ACM SIGCOMM Workshop on Networked Systems in Developing Regions (NSDR), August 2007, **pp. 1-6**.
- [5] S. Farrell and V. Cahill, “*Delay and Disruption Tolerant Networking*”, Artech House, 1st Edition, 2006. ISBN 1-59693-063-2.
- [6] K. Scott, S. Burleigh, “*Bundle Protocol Specification*”, IETF RFC 5050, experimental, November 2007.
- [7] L. Wood, W. M. Eddy, W. Ivancic, J. McKim, C. Jackson, “*Saratoga: A Delay-Tolerant Networking convergence layer with efficient link utilization*”, International Workshop on Space and Satellite Communications (IWSSC’07), September 2007, **pp. 168 – 172**.
- [8] W. Ivancic, W. M. Eddy, D. Stewart, J. Northam, C. Jackson, A. daSilva Curiel, “*Use of the Delay Tolerant Networking Bundle Protocol from Space*”, NASA/TM-2009-215582; IAC-08-B2.3.10; E-16810; 59th International Astronautical Congress and Exhibition 2008 (IAC), Glasgow, Scotland, United Kingdom, 29 Sep. - 3 Oct. 2008
- [9] W. Ivancic, L. Wood, P. Holliday, W.M. Eddy, D. Stewart, C. Jackson, J. Northam, “*Experience with delay tolerant networking from orbit*”, ASMS 2008, 26-28 Aug. 2008, **pp. 173-178**
- [10] L. Wood, A. daSilva Curiel, W. Ivancic, D. Hodgson, D. Shell, C. Jackson, D. Stewart, “*Operating a terrestrial Internet router onboard and alongside a small satellite*”, Selected Proceedings of the 56th International Astronautical Federation Congress, Fukuoka, Japan, 17-21 October 2005.
- [11] S. Burleigh, M. Ramadas, S. Farrell, “*Licklider Transmission Protocol – Motivation*”, Internet draft, RFC 5325.
- [12] A. Keränen, J. Ött, T. Kärkkäinen., “*The ONE Simulator for DTN Protocol Evaluation*”. SIMU Tools 2009, Rome, Italy. ISBN 978-963-979945-5.
- [13] H. Ericsson, P. Jönsson, “*Implementation and Analysis of the Bundling Protocol for DTN Architectures*”, Lulea University of Technology, M.Sc. Thesis, 2005.
- [14] C. Caini, P. Cornice, R. Firrincieli, D. Lacamera, S. Tamagnini, “*A DTN Approach to Satellite Communication*”, IEEE Journal on Selected Areas in Communications, Vol. 26, No. 5, June 2008, **pp. 820-827**.
- [15] A. Shahriar, “*Saratoga (a file transfer protocol) implementation in ns-2*” University of Oklahoma, Technical Note, October 2008.
- [16] J. Segui, E. Jennings, “*Delay Tolerant Networking – Bundle Protocol Simulation*”, Space

Mission Challenges for Information Technology, IEEE International Conference, **pp. 235-240**, 2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06), 2006.

[17] S. Farrell and V. Cahill, "Delay and Disruption Tolerant Networking", Artech House, 1st Edition, 2006. **p. 171.**

[18] D. Maessen, J. Guo, E. Gill, B. Gunter, Q.P. Chu, G. Bakker, E. Laan, S. Moon, M. Kruijff, G.T. Zheng, "*Conceptual Design of the FAST-D Formation Flying Spacecraft*", 7th IAA Symposium of Small Satellites for Earth Observation, Berlin, Germany, 4-8 May 2009.

[19] J. Benevise, "*The ENVISAT Radar Altimetry Mission Data Products*" SP-461, ERS-ENVISAT Symposium, Goteborg, Sweden 16-20 October 2000.

[20] J. Muylaert, C. Asma, W. J. Ubbels, J. Rotteveel, V. Danilkin, S. Makhankov. "*QB50: An International Network of 50 CubeSats in LEO for Lower Thermosphere and Re-Entry Research*" 23rd Annual Conference on Small Satellites, CubeSats Developer's Workshop, Utah State University, USA, August 8-9th, 2009