CEAS Space J
DOI 10.1007/s12567-016-0127-3

ORIGINAL PAPER

Applicability of delay tolerant networking to distributed satellite systems

A. Freimann[1] · T. Tzschichholz[2] · M. Schmidt[3] · A. Kleinschrodt[1] · K. Schilling[1]

Received: 21 March 2016 / Revised: 20 May 2016 / Accepted: 24 May 2016

Abstract Currently, a trend towards distributed small satellite missions is emerging using cooperating satellites to achieve joint mission objectives, e.g. for earth observation. Communication is a key feature when cooperation between satellites is desired. Typically those satellite networks are affected by slow data rates, high packet loss and intermittent connectivity. To address these challenges the store-and-forward approach of the delay tolerant networking (DTN) concept is investigated in this article. Network simulations of typical scenarios were carried out and evaluated to derive statements about the applicability of the DTN approach to networks in low earth orbits.

Keywords Inter-satellite communication · Delay tolerant networking (DTN) · Distributed satellite systems (DSS)

1 Introduction

In recent years many activities concerning research and development of CubeSats have been carried out, small satellites with only a few kg mass. Now a trend towards multi-satellite systems is emerging, with the objective to use interacting small satellites to realize distributed missions. There are various fields of application for distributed satellite systems (DSS), such as earth observation missions, distributed sensor networks and communication relays for remote locations. Also the combination of traditional large monolithic satellites with distributed small satellite systems is promising. In such distributed architectures, inter-satellite communication is one of the key aspects and, therefore, potential approaches need to be investigated. Hence, inter-satellite communication is currently one of the hot topics in the satellite community. Adequate networking solutions are required to allow for efficient data exchange within challenging networks such as DSS in earth orbits. Therefore, the benefit of delay tolerant networking (DTN) for DSS is investigated, a promising approach for communication in challenging networks. Within the scope of this article, extensive simulations for typical scenarios have been performed to investigate the benefit of using DTN in those systems.

To define the scope of our investigation an introduction to DSS is given in the following section. In Sect. 3, the DTN concept is introduced and its differences compared to traditional approaches are characterized followed by an overview of available implementations and in-orbit test results. Subsequently, in Sect. 4 a description of the simulated scenarios and our simulation framework follows, in Sect. 5 the results of the simulations are discussed. Section 6 concludes with derived statements about the applicability of the DTN approach to DSS and an outlook on future developments.

2 Communication in distributed satellite systems

In the last decade, many small satellites were launched into low earth orbits. Space vehicles are considered as small

This paper is based on a presentation at the German Aerospace Congress, September 22–24, 2015, Rostock, Germany.

✉ A. Freimann
freimann@informatik.uni-wuerzburg.de

1 Universität Würzburg, Am Hubland, 97074 Würzburg, Germany

2 Zentrum für Telematik e.V., Magdalene-Schoch-Straße 5, 97074 Würzburg, Germany

3 Hochschule Bochum, Höseler Platz 2, 42579 Heiligenhaus, Germany

Published online: 22 June 2016

satellites when the total mass is less than 500 kg. Especially picosatellites (1 kg) and nanosatellites (10 kg) have successfully proven the concept of miniaturized research satellites [1, 2].

All basic subsystems (e.g. attitude and position control) for implementing DSS on a pico- or nano-satellite scale have been developed lately [3, 4]. Due to the limited mass of these satellites and the CubeSat standard [5], it is possible to launch several of them with a single launcher.

As a result, more and more space missions are designed as DSS [6], i.e. several satellites act together to achieve joint mission objectives. A distributed system is a collection of independent systems, working together to perform a desired task.

Furthermore, there are a number of terms used to describe the topology of DSS, most commonly constellation, formation and swarm/cluster [7], described in the following.

A *constellation* is a group of satellites with coordinated ground coverage complementing each other to achieve a high global coverage. Satellites in a constellation do not control their relative positions and are controlled separately from ground stations. A well-known satellite constellation is the Global Positioning System (GPS).

A group of satellites controlling their relative positions autonomously is called a *formation*. Successful formation experiments were performed with the PRISMA mission [8] and the CanX-4&5 mission [9].

A group of satellites without fixed absolute or relative positions, working together to perform a common task is called a *cluster* or *swarm*. The QB50 satellites [10] can be described as a swarm. One of the goals of this mission is to carry out atmospheric multi-point measurements in low earth orbits.

A lot of novel space mission concepts become feasible using distributed approaches. Many researchers consider DSS as the next evolutionary step of space missions with the main advantage being the increase in temporal and spatial resolution. Furthermore, novel mission concepts such as distributed virtual instruments or sensor networks in space can be implemented [11]. Especially for earth observation applications, DSS have a huge potential, e.g. for clouds height measurements [12]. Applications like weather monitoring and earth imaging clearly benefit from distributed approaches as well. An increasing number of space missions based on DSS is currently in various stages of development, most of them Earth science-related missions [6].

One of the first missions which intended to send a large number of spacecrafts in a single mission to a LEO orbit is the QB50 project. The objective of QB50 is to send 50 satellites from different research institutes to space to perform in situ measurements in lower thermosphere [10]. Another very interesting mission concept is the Orbiting Low-Frequency Array for Radio Astronomy (OLFAR) project, which aims to send more than 50 identical nanosatellites to space to build a radio telescope from an array of antennas. An efficient communication topology is necessary to gather radiometric observations to ground [13]. Within the NetSat project [14], a formation of four picosatellites is to be demonstrated. The main challenge is to use picosatellites, as satellites in this size have never been used in a formation before.

Inter-satellite links are generally required for formation control, payload data forwarding, clock synchronization and other purposes, depending on the mission. Inter-satellite communication is not implemented in all existing DSS missions, as in the case of GPS, but most DSS benefit from inter-satellite communication, such as Iridium. For others, such as formations inter-satellite communication is obviously necessary, as communication over the ground segment implies too high latencies. Generally, inter-satellite communication is the key feature required when cooperation between satellites is desired.

Terrestrial communication networks usually show low link latencies, static routes, low packet loss and high connectivity. Links are fast and symmetrical, CPUs fast, memory cheap and networks can be managed and configured by central nodes.

Experiences from previous CubeSat missions, such as the UWE-3 mission [15], clearly show that radio communication in space is highly affected by interferences, resulting in packet loss and link disruptions.

In contrast to terrestrial networks space communication technologies need to handle various challenges, such as short contact times (e.g. ground station contacts), frequent interferences, asymmetrical links, limited energy (for antenna pointing) and highly heterogeneous networks (e.g. different satellites and ground stations). Communication systems of small satellites suffer from high signal attenuation, limited bandwidth, low transmission power, small antenna gains resulting in low data rates and frequent interruptions. Hands-on repair during a mission is impossible and reconfiguration capabilities are limited; therefore, communication systems need to be particularly robust.

On the other hand, satellite motions are highly predictable due to orbital dynamics and also data traffic is predictable to a certain degree. All those challenges and advantages of communication in DSS need to be handled by appropriate communication technologies.

Communication protocols are required to cope with the specific properties of space radio links and the dynamic topology of DSS in low earth orbits. Typical design goals of space networks are reliability, power efficiency, minimized communication overhead, autonomous reconfigurability, short and predictable latencies as well as ad hoc networking capabilities.

The following subsection describes Delay Tolerant Networking, a communication approach particularly suitable for challenging mobile networks.

3 Delay tolerant networking

To date CubeSat missions mostly rely on the AX.25 protocol for point-to-point communication between ground station and satellite. AX.25 is a data link layer protocol providing error detection and very limited routing features. It is supported by low-cost amateur radio equipment, thus it is used to simplify ground station setup and utilize the amateur radio community to increase downlink opportunities. Since most existing CubeSat missions are single satellite missions, there was no need for the development of appropriate networking protocols [16]. For future distributed satellite missions, the capabilities of AX.25 are not sufficient. Protocols with ad hoc networking capabilities and efficient routing algorithms are required. Although there is a large variety of terrestrial protocols providing most of the required capabilities, satellite networks encounter specific properties that make existing networking protocols inefficient, as for example, the need for instantaneous end-to-end connections.

DTN protocols are designed to address the challenges of space networks by introducing hop-by-hop store-and-forward communication [17].

The term DTN refers to a networking approach that is able to handle high delays and interrupted connectivity. Originally, DTN protocols were developed to support interplanetary links where high latencies and intermittent connectivity had to be overcome to allow communication with Mars rovers, for example. But soon, the potential of the DTN approach for LEO networks was recognized too [18].

The main difference compared to traditional mobile ad hoc network (MANET) protocols is the store-and-forward policy in DTNs.

Common MANET protocols try to establish a complete route between source and destination node and then start message transfer only if such an end-to-end path is available, as depicted in the upper part of Fig. 1. If those paths are not available or interrupted frequently, as in space networks, this approach is very inefficient and fault-prone.

In contrast, using a store-and-forward approach data is incrementally stored and forwarded across the network until they finally reach their destination. Therefore, packets are stored in each node of the path until they can be forwarded to next node. Furthermore, packets can be transferred over multiple paths. This multicast routing increases reliability in case of high packet

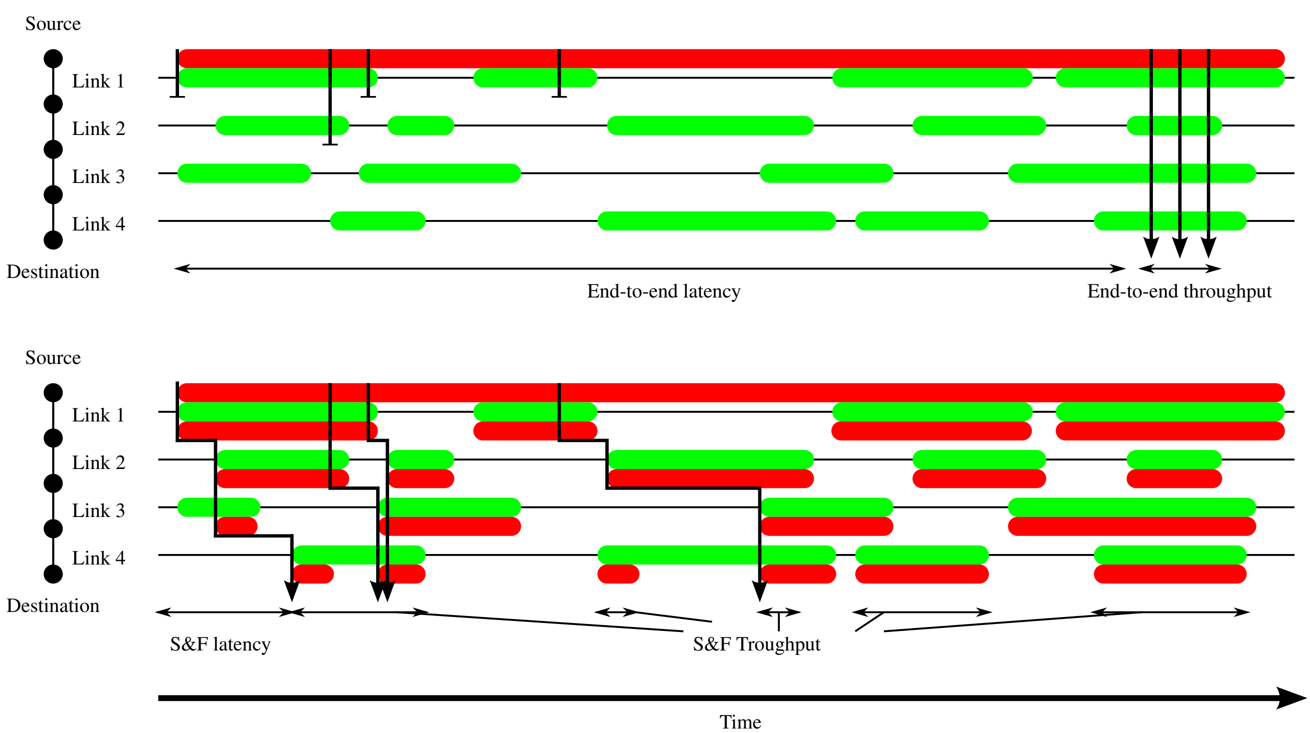


Fig. 1 Comparison between message transport in MANETs (*top*) and DTNs (*bottom*). While MANETs can only transport messages across the network, when there is an instantaneous path between source and destination, DTNs can transport messages partially in steps between the nodes. Contacts are shown in *green* and data transmissions in *red*

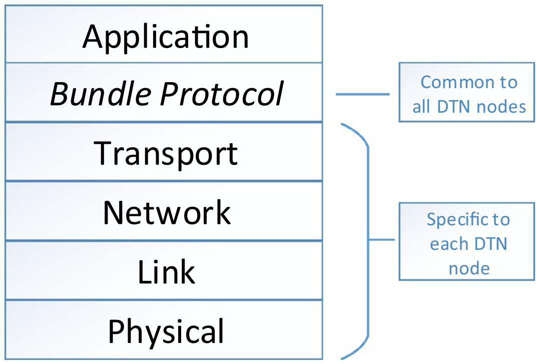


Fig. 2 DTN protocol stack including the bundle protocol

loss or unpredictable network topologies. Generally, the store-and-forward concept also reduces overhead for retransmissions and in networks with sparse contacts it makes better use of available communication opportunities. In this way latencies can be decreased and throughput increased, as depicted in the lower part of Fig. 1. As an example, DTN protocols enable satellite networks to efficiently downlink payload data by increasing the number of opportunities to link with other satellites and ground stations. Additionally, the latency in case of transmission errors is decreased, as a message does not have to travel across the entire network again, but only through the link where a transmission error occurred [19]. This is beneficial in networks affected by high interferences such as space networks.

The Bundle Protocol (BP) is the de-facto DTN standard protocol. It provides store-and-forward transport of so-called bundles (a series of contiguous data blocks) and acts as an overlay network over a variety of heterogeneous networks, such as ground station and satellite networks.

Figure 2 illustrates the integration of the BP into ISO protocol layers. The BP is designed to be used throughout a DTN. Also different BP implementations can work together. Furthermore, it is independent of lower layer protocols. Those are chosen according to the specific characteristics of each communication environment.

Therefore, the BP is a promising candidate to provide a common basis for future space networks, such as the Internet Protocol (IP) serves as a common basis for most terrestrial networks.

The BP has already been used in space [20, 21], e.g. at ISS [22]. However, it is still under development. In the literature, problems were identified concerning security, reliability and the required clock synchronization [23].

DTN2,[1] ION[2] and IBR-DTN[3] are the most popular BP implementations. DTN2 is designed as a general and flexible implementation for research and development based on the BP, whereas ION and IBR-DTN are designed rather for specific applications.

In the following section, a network simulation environment is presented for investigation of the applicability of the BP to typical DSS scenarios.

4 Simulation

Considering satellite communication, one crucial design challenge is the testing of newly developed components and protocols since in-orbit experiments are only possible once the system is completed and operational in space. Possibilities for subsequent changes are limited and necessary corrections to compensate for errors in the system may not be possible. Therefore, simulations are an important tool to evaluate algorithms and protocols. Accordingly, our main goal was to implement a simulation environment for performance investigations of DTN protocols in selected scenarios.

4.1 Simulated scenarios

We investigated DTN protocols in two different scenarios.

For our investigations we selected two typical topologies. The first (Fig. 3a) is a Walker constellation, an evenly distributed configuration, used for communication and earth observation missions because of their high ground coverage. The selected configuration is a very sparse system consisting of 18 satellites in a $45^\circ : 18/6/0$ constellation, which means the satellites are distributed on six different orbits with equal inclination and three satellites per orbit.

The second scenario (Fig. 3b) is a satellite swarm, similar to the QB50 mission, where a lot of satellites will be launched as a string of pearls. This scenario encompasses 50 satellites in a mesh-like network on slightly different orbits.

The altitude is 700 km and the orbital period approximately 99 min in both scenarios. The satellites are assumed to be CubeSats with typical properties such as low transmission power, small omnidirectional antennas and transmission frequencies in the UHF band. Therefore, a data rate of 10 kbps was assumed. The maximum link distance is assumed to be the line of sight distance. Of course the maximum distance may be considerably less in real scenarios depending on hardware parameters. In our opinion, this is a meaningful worst case assumption regarding the benefit of DTN in DSS, since the benefit is decreasing in case of higher network connectivity.

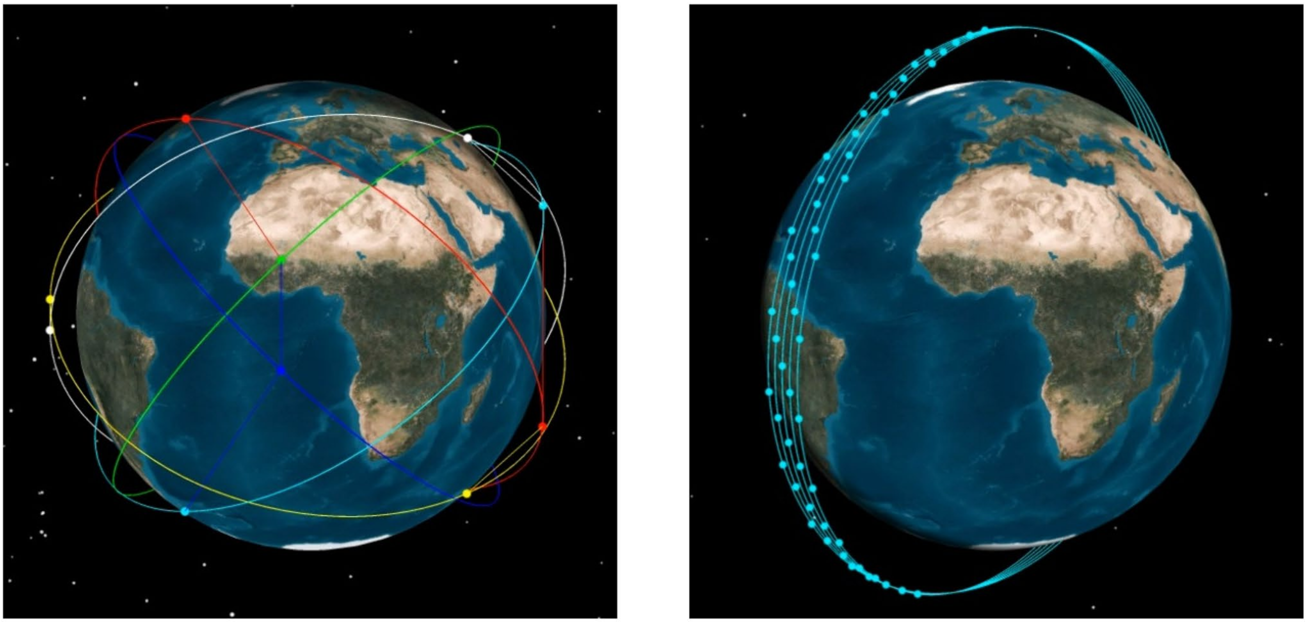
4.2 Simulation environment

The performance of DTN protocols in the defined scenarios was investigated by network simulations. Figure 4 illustrates the structure of the entire simulation environment.

[1] http://www.dtnrg.org/.

[2] http://sourceforge.net/projects/ion-dtn/.

[3] http://trac.ibr.cs.tu-bs.de/project-cm-2012-ibrdtn.



(a) Scenario 1 consists of 18 satellites in a Walker constellation: six different orbits, three satellites per orbit.

(b) Scenario 2 consists of 50 satellites in a mesh-like network.

Fig. 3 The two different scenarios relevant to the project. Due to the significant impact of contact times, which are not relevant for scenario 2, we focus on scenario 1 within this article. Image courtesy of Analytical Graphics, Inc. (http://www.agi.com)

For the presented work, the NS-3[4] network simulator was used. NS-3 is a widespread object-oriented discrete event network simulation tool. It enables for the simulation of mobile radio networks, providing models for nodes, links, mobility and implementations of well-known communication protocols. Since it does not support satellite communication and DTN protocols, additional features have been implemented by our team. Satellites were integrated as models, defining their main properties such as transmission power, transceiver losses, antenna gains as well as mobility models describing the satellite trajectory. The radio channel was modelled, including losses and geometric models. The trajectories of the satellites were calculated according to the SGP4 model [24], taking into account perturbations due to the oblateness of the Earth as well as third body perturbations and atmospheric drag. Therefore, Two-Line Element sets (TLE) were generated for the defined scenarios and trajectories were calculated for simulation periods of up to 7 days. Application models are defined for data generation, e.g. housekeeping, sensor or payload data. The simulation environment integrates all system models to analyse the resulting communication performance. During simulation, the entire network traffic is captured for evaluation of performance parameters such as throughput, delays and packet loss.

Available implementations of the BP have been investigated, e.g. DTN2, ION and IBR-DTN. Since most implementations are tailored to specific applications, for the presented simulations the flexible DTN2 reference implementation was selected.

The integration of DTN2 was performed using LXC,[5] a lightweight operating system-level virtualisation. It was used to generate a virtual machine for each satellite, whereby network interfaces and name spaces are isolated from each other to avoid undesired interactions between network nodes, affecting the simulation results.

For lower protocol layers, an adaption of IEEE 802.11 was integrated. Its suitability for satellite formations is acknowledged in [25–27].

All software components are integrated on a single computer running a Linux operating system. As shown in Fig. 4, the physical parameters of the satellites are defined in NS-3, which produces models of the satellites and the communication channel in between, based on the defined parameters. Furthermore, the lower protocol layers are implemented within the NS-3 environment. Each

[4] https://www.nsnam.org/.

[5] https://linuxcontainers.org/lxc/.

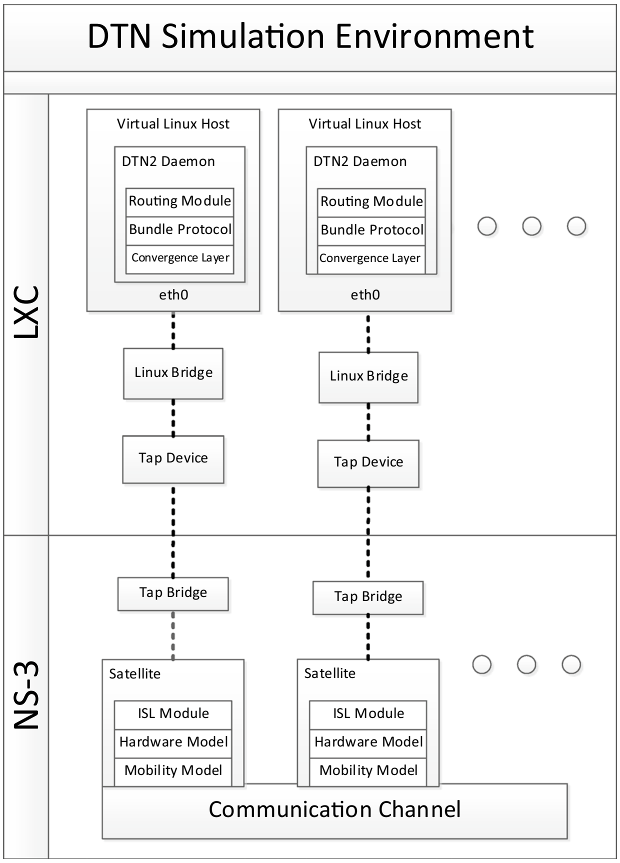


Fig. 4 Simulation environment based on NS-3 and LXC

satellite model is connected to a virtual machine outside the NS-3 environment via a network bridge. On these virtual machines, the second part of the satellites is implemented, mainly the DTN protocols and the application model. The flow of a single packet (Bundle) can be roughly described as follows: the DTN2 daemon on a virtual machine of a source satellite generates data and sends it as payload of a Bundle to the NS-3 environment. There the Bundle is by the satellite model and transmitted through the simulated communication channel to all satellites in reach according to the current locations of all satellites. From these receiving satellites, the Bundle is then forwarded to their virtual machine, respectively. There the DTN2 daemon receives the Bundle, logs the reception and decides whether to forward the Bundle to further satellites. After the simulation, Matlab is used to analyse the recorded network traffic and calculate performance parameters and produce plots.

5 Results

In the following, the results are presented, obtained from network simulations of the two scenarios, described in Sect. 4.1. Both simulations were performed within the simulation environment presented in Sect. 4.2. In essence, two networks of satellites were simulated, which differ in the spatial configuration and hence, the availability of links between the satellite nodes as well as the change over time.

5.1 First scenario

Simulations carried out for the first scenario show significant results for the applicability of DTNs in networks with a lower number of nodes and intermittent connectivity. In particular, message delays, local message storage and throughput for different simulation parameters were analysed.

When not stated otherwise, all plots relate to the satellite pair 2–16, which we have picked because it has almost no direct contact and hence is interesting for investigating performance issues. When relevant, we have mentioned additional effects or relations to other pairs in the text.

5.1.1 Simulation parameters

We focused on pairs of nodes, where no direct end-to-end path exists between both satellites and thus store-and-forward communication is necessary to deliver messages. Therefore, the throughput of common MANET routing protocols would be equal to zero and the benefit of store-and-forward communication is easily measurable. For the simulations one satellite is defined as data source and the other is defined as data sink.

Regarding the software configuration parameters, we have not tuned parameters specifically for the measurements. Any parameter changed by us will be indicated in the remainder of the text.

We used “flood” routing of the DTN2 implementation, which is based on simple broadcasting of messages. Thus, whenever a node wants to transmit a message it will send it to every node in range. These nodes will repeat the process and send a copy of the message to each of their respective neighbours. This approach is also called “Epidemic Routing” since packets spread over the network like a virus. The simplicity of this algorithm is an advantage in terms of reliability. As packets spread over all available paths in the network, the probability of a successful delivery is maximized. Another advantage is the latency of delivered packets. Since all paths are included, the shortest one is definitely among them. Obviously, this algorithm is not very efficient since a lot of overhead in terms of redundancy is generated. This overhead leads to network congestion and exhaustion of node buffers which decreases the overall network performance. Nevertheless, epidemic routing is beneficial in case of smaller error-prone networks such as this simulation scenario. Furthermore, since no specific routing decisions are made, the influence of those is eliminated and our simulation results show the pure benefit of the store-and-forward principle.

The simulation results are depending on a lot of different parameters of the BP implementation such as discovery intervals as well as lower layer protocols and corresponding parameters. Therefore, we just draw general conclusions by identifying trends from extensive simulations and present selected plots and numbers.

5.1.2 *Contact times*

In scenario one, contact times are rather sparse. This can be seen in Fig. 5. The matrix shows the time in minutes per orbit, during which a link between a pair of satellites can be established (either direct or indirect). This analysis is based on the assumption that data exchange is possible over the whole line of sight distance. In case of lower link distances the connectivity would be even worse, so this is a best case assumption with respect to network connectivity.

While there are a few pairs, which can communicate with each other most of the time (such as 11 and 1), there are also pairs, which do not have any direct contact at all (for instance 10 and 12). Thus, it is important, that by store-and-forwarding messages along direct link opportunities, communication is made possible.

5.1.3 *TX/RX throughput*

The total throughput of the system in scenario 1 was simulated with different delays between the transmission of messages (bundles). The bundle payload size was 300 bytes.

1	99	0	0	5	0	32	0	2	53	0	76	0	46	2	0	43	0	4
2	0	99	0	33	0	0	68	0	2	0	0	65	0	46	2	4	40	0
3	0	0	99	0	34	0	2	46	0	60	0	0	2	0	63	0	4	32
4	5	33	0	99	0	0	0	0	32	0	2	46	0	54	0	46	2	0
5	0	0	34	0	99	0	32	0	0	63	0	2	0	0	41	0	46	2
6	32	0	0	0	0	99	0	32	0	2	46	0	41	0	0	2	0	57
7	0	68	2	0	32	0	99	0	0	0	0	40	0	2	46	0	65	0
8	2	0	46	0	0	32	0	99	0	32	0	0	57	0	2	0	0	58
9	53	2	0	32	0	0	0	0	99	0	38	0	2	46	0	64	0	0
10	0	0	60	0	63	2	0	32	0	99	0	0	0	0	34	0	2	46
11	76	0	0	2	0	46	0	0	38	0	99	0	32	0	0	53	0	2
12	0	65	0	46	2	0	40	0	0	0	0	99	0	33	0	2	68	0
13	46	0	2	0	0	41	0	57	2	0	32	0	99	0	0	5	0	32
14	2	46	0	54	0	0	2	0	46	0	0	33	0	99	0	32	0	0
15	0	2	63	0	41	0	46	2	0	34	0	0	0	0	99	0	32	0
16	43	4	0	46	0	2	0	0	64	0	53	2	5	32	0	99	0	0
17	0	40	4	2	46	0	65	0	0	2	0	68	0	0	32	0	99	0
18	4	0	32	0	2	57	0	58	0	46	2	0	32	0	0	0	0	99
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Fig. 5 Contact times in minutes between two satellites (including indirect links) for one orbit

As can be seen in Table 1, in most cases the received data rate was lower than the generated data rate. This is related to message losses in the network (transmission errors). However, in the measurement where the time delay between messages was set to 40 s, the receiving node was actually receiving more data than the sending node was providing. This shows that sometimes, multiple paths can transport the same message, what is a large advantage with regard to reliability.

The bundle delivery rate shows to be highly dependent on the general network load. When bundles have been generated faster and thus, more bundles were transferred through the network rising its load, the delivery rate dropped significantly. However, since collisions and fine-grained timing are random, we also observed outliers, as in the case in which the message generation interval was set to 40 s. It is possible that the receiving node was overloaded due to a large number of duplicate bundles it has received. This is supported by the high RX data rate measured. In the end, this is a characteristic disadvantage of "flood" routing.

5.1.4 *Message latency*

Each message sent through the network of nodes has a certain latency. It depends on the contact opportunities with other nodes, as well as the local node storage capacity. When the message buffer of the local node is full, it may take multiple attempts to deliver the message. However, for our tests, we did not impose an artificial limit for the message buffer to have the results reflect the true delivery rates of the store-and-forward system. The only limit was imposed by the storage capacity of the hard drive, but due to several magnitudes of storage space of difference in between, this is not a limitation.

We measured the distribution of the message latency for different message rates.

As shown in Fig. 6, messages travelled through the network for more than 30 min. No message could be delivered in less than 25 min. The reason for these latencies is the dynamic topology. Contacts between the satellites are

Table 1 Throughput and bundle delivery rate (BDR) of a two node links in the network described in scenario 1 for different intervals between generated bundles

Interval (s)	TX (B/s)	RX (B/s)	BDR (%)
10	14.4	11.9	17
20	7.31	6.3	44
30	4.88	2.94	100
40	3.67	8.45	44
50	2.94	1.84	100

The RX throughput can be larger than the TX throughput, when data are received from more than one path sparse and thus, time passes, as they get in contact with each other.

Sometimes, depending on the message rate, even a saturation effect can take place: in this case, messages cannot be delivered at once, as an important link between two satellites is severed for one orbit, and transmission continues when the constellation returns. Depending on the number of messages stored and the number of duplicates, it may take multiple orbits until the last messages can be finally delivered. This can be seen in Fig. 7. Here, messages travelled through the network for more than 4 h.

Overall, we observed different distributions of the message latency depending on the selection of nodes and packet generation. This is intended in the design. As can be seen, DTNs are very good at delivering messages across challenging networks, where contacts between the nodes are sparse. Beyond that, store-and-forward can be advantageous in situations where a complete end-to-end link is not available, but intermittent links between nodes on the path are available on a regular basis. Compared to end-to-end connection-based routing, the positive effect of store-and-forward routing is reflected in a significant increase of message delivery rates in such configurations.

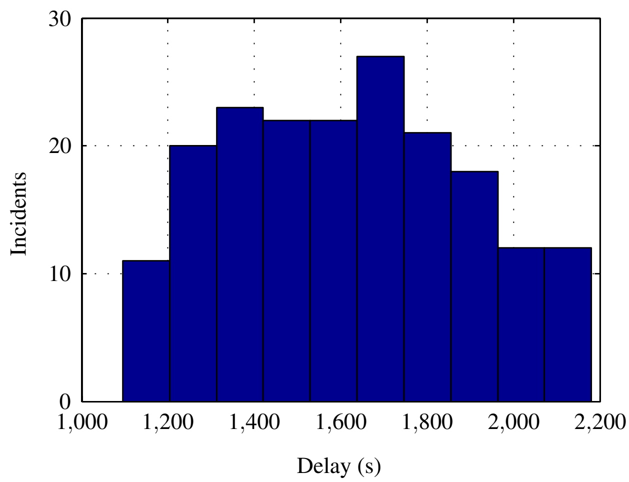


Fig. 6 Message (bundle) delay histogram (distribution) for a message delay of 10 s

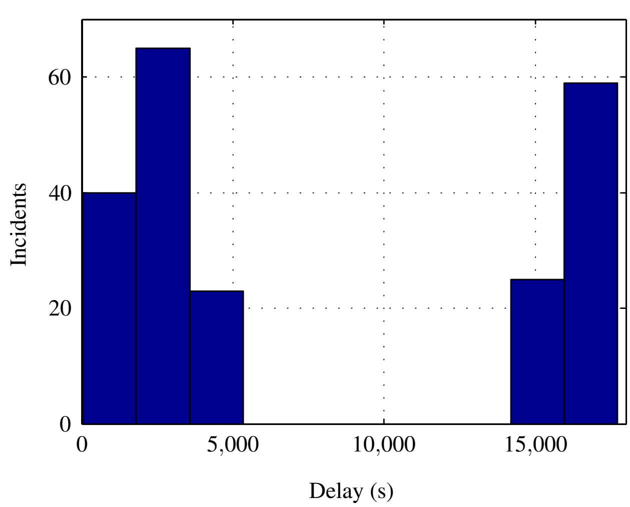


Fig. 7 Message (bundle) delay histogram (distribution) for a message delay of 30 s

5.1.5 Node message storage

Another very interesting quantity is the node message storage. When a node receives a message, it is locally stored until it expires. When the node message storage is recorded over time, some interesting effects can be seen, which are discussed next.

We investigated the storage size per node. A direct comparison between these values allows tracing the paths of the messages and more importantly, the node utilization. A node with a large number of contact opportunities among the path will have more messages stored, while a node with less contacts will have less. For the already discussed measurement with a message rate of 1/30 s, the node storage size is shown in Fig. 8.

As can be seen, when the sending node starts generating messages, other nodes quickly pick them up. Depending on the path length and the connectivity of the respective node, the other nodes receive all messages or only a part of them. In addition, orbit constellations are indirectly visible in the form of sudden message receptions—often in larger numbers—whenever links become suddenly available.

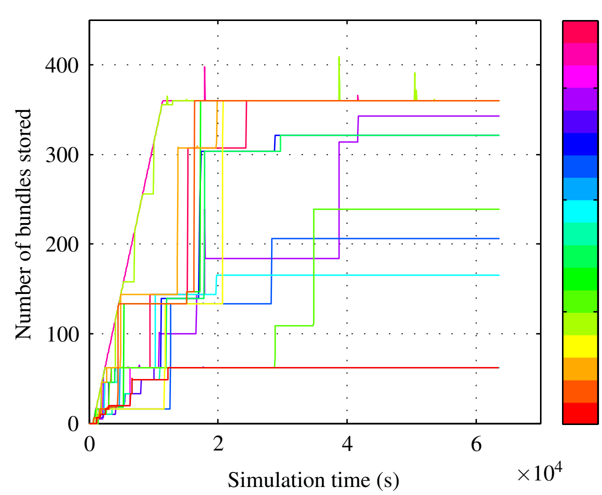


Fig. 8 Numbers of stored messages for all nodes in the simulated network, using a message rate of 1/30 s. The *colors* correspond to the node numbers

5.2 Second scenario

The second scenario is a dense mesh network, such that there are paths in the network between all satellites. Due to this property, the results for the second scenario showed that the performance of DTN protocols in this kind of satellite networks is very low since the store-and-forward mechanism has no benefit. Furthermore, the overhead of the BP produces additional load on the links. This causes collapses of the network communication due to low data rate, high number of nodes and duplication of packets within the network. This problem could be observed in scenario 1 as well in cases where the message generation interval was small. Hence, the simulations of scenario 2 showed that efficient networked communication is not possible (also in comparable scenarios) using the applied DTN protocols. This shows that the BP together with the applied routing approach is no general solution for comparable satellite networks. For mesh-like networks, protocols with lower overhead should be preferred.

6 Conclusions

In this article, the applicability of DTN for use in DSS with different topologies was analysed. With respect to the two selected scenarios, the results showed that especially for topologies, where contacts are sparse, DTN protocols do have a large advantage. Most importantly, a store-and-forward approach is required to allow communication between nodes with a lack of end-to-end connections.

The main problem we encountered is the fact that messages are not being repeated when they are lost. We observed messages being sent but not being received due to transmission errors and the sending node did nothing to compensate. This confirms the statement in [23] that the BP lacks reliability.

We have also found that the overall results depend heavily on random occurrences of packet losses. The impact of randomness is much larger than in MANETs, as the consequences of a single packet loss are amplified when the network is sparse.

In addition, with regard to pico- and nanosatellites, the BP as such is too computationally intensive, especially regarding memory. While the source and destination identifiers are very flexible and powerful, string processing and variable-length headers complicate things in systems with a small memory and CPU footprint. Therefore, we suggest to derive a small-footprint DTN protocol, using numeric IDs as node identifiers and eliminating all variable-length fields in the header.

Additionally, since in small satellite networks, payload data are mostly telemetry with a small size, the large header of the BP introduces a lot of overhead.

If these shortcomings are fixed, we believe that DTNs are indeed a very powerful and flexible approach to the communication requirements of DSS.

Future investigations will focus on more sophisticated routing protocols for the deterministic dynamics, more efficient communication approaches for mesh-like networks, integration of ground station communication and hardware-in-the-loop experiments.

Acknowledgments Main parts of the investigations were performed in the course of the BayKoSM project, funded by the Bavarian Government.

References

1. Schmidt, M., Zeiger, F.: Design and implementation of in orbit experiments for the pico satellite UWE-1. In: International Astronautical Congress, IAC-06-E2, 1 July 2006 (2006)
2. Busch, S., Bangert, P., Schilling, K.: Attitude control demonstration for pico-satellite formation flying by UWE-3. In: 4S Symposium (2014)
3. Bangert, P., Busch, S., Schilling, K.: Performance characteristics of the UWE-3 miniature attitude determination and control system. In: Proceedings of the 2nd IAA Conference on Dynamics and Control of Space Systems, Roma (2014)
4. Schilling, K.J.: Preparing technologies for cooperating pico-satellite networks: the UWE-3 and -4 misions. In: Proceedings of the IAA International Workshop on Constellations and Formation Flying, Lisbon, IWSCFF-2013-01-06 (2013)
5. Munakata, R., et al.: CubeSat Design Specification. The CubeSat Program. California Polytechnic State University, San Luis Obispo (2015)
6. Bandyopadhyay, S., Subramanian, G.P., Foust, R., Morgan, D., Chung, S.J., Hadaegh, F.: A review of impending small satellite formation flying missions. In: Proceedings of the 53rd AIAA Aerospace Sciences Meeting, Kissimmee (2015)
7. Schilling, K.: Networked distributed pico-satellite systems for earth observation an telecommunication applications. In: IFAC Workshop Aerospace Guidance, Navigation and Flight Control Systems, Samara (2009)
8. Persson, S., Bodin, P., Gill, E., Harr, J., Jörgensen, J.: PRISMA—an autonomous formation flying mission. In: ESA Small Satellite Systems and Services Symposium (4S), Sardinia, pp. 25–29 (2006)
9. Bonin, G., Roth, N., Armitage, S., Newman, J., Risi, B., Zee, R.E.: CanX-4 and CanX-5 precision formation flight: mission accomplished! In: 29th Conference on Small Satellites (2015). http://digitalcommons.usu.edu/smallsat/2015/all2015/3/
10. Thoemel, J., Singarayar, F., Scholz, T., Masutti, D., Testani, P., Asma, C., Reinhard, R., Muylaert, J.: Status of the QB50 cubesat constellation mission. In: Proceedings of the 65th International Astronautical Congress, Toronto (2014)
11. D'errico, M.: Distributed Space Missions for Earth System Monitoring. Springer, New York (2013)
12. Zaksek, K., Gerst, A., von der Lieth, J., Ganci, G., Hort, M.: Cloud photogrammetry from space. Int. Archiv. Photogr. Remote Sens.Spat. Inf. Sci. **40**(7), 247 (2015)

13. Budianu, A.: OLFAR: adaptive topology for satellite swarms. In: International Astronautical Congress 2011 (2011)
14. Schilling, K., Bangert, P., Busch, S., Dombrovski, S., Freimann, A., Kleinschrodt, A., Kramer, A., Nogueira, T., Ris, D., Scharnagl, J., Tzschichholz, T., Islas, G., Zhou, L.: Netsat: a four pico/nano-satellite mission for demonstration of autonomous formation flying. In: 66th International Astronautical Congress (2015)
15. Kleinschrodt, A., Freimann, A., Schmidt, M., Schilling, K.: Lessons learned from in orbit operations of the UWE-3 pico-satellite. In: 66th International Astronmautical Congress, Jerusalem (2015)
16. Muri, P., Mcnair, J.: A survey of communication sub-systems for intersatellite linked systems and CubeSat missions. J. Commun. **7**(4), 290–308 (2012)
17. Fall, K.: A delay-tolerant network architecture for challenged internets. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 27–34 (2003). doi:10.1145/863955.863960
18. Burleigh, S.C., Birrane, E.J.: Toward a communications satellite network for humanitarian relief. In: Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief, pp. 219–224. ACM, New York (2011)
19. Warthman, F., et al.: Delay- and disruption-tolerant networks (DTNS) a tutorial. Interplanetary Internet Special Interest Group (2015)
20. Ivancic, W., Eddy, W.M., Stewart, D., Wood, L., Northam, J., Jackson, C.: Experience with delay-tolerant networking from orbit. Int. J. Satell. Commun. Netw. **28**(5–6), 335–351 (2010). doi:10.1109/ASMS.2008.37
21. Wyatt, J., Burleigh, S., Jones, R., Torgerson, L., Wissler, S.: Disruption tolerant networking flight validation experiment on NASA's EPOXI mission. In: First International Conference on Advances in Satellite and Space Communications, 2009, SPACOMM 2009, pp. 187–196. IEEE, New York (2009)
22. Jenkins, A., Kuzminsky, S., Gifford, K.K., Pitts, R.L., Nichols, K.: Delay/disruption-tolerant networking: flight test results from the international space station. In: 2010 IEEE Aerospace Conference, pp. 1–8. IEEE, New York (2010)
23. Wood, L., Eddy, W.M., Holliday, P.: A bundle of problems. In: 2009 IEEE Aerospace Conference, pp. 1–17. IEEE, New York (2009). doi:10.1109/AERO.2009.4839384
24. Vallado, D.A., Crawford, P.: SGP4 orbit determination. In: Proceedings of AIAA/AAS Astrodynamics Specialist Conference and Exhibit, pp. 18–21 (2008)
25. Knoblock, E.J., Wallett, T.M., Konangi, V.K., Bhasin, K.B.: Network configuration analysis for formation flying satellites. In: 2001 IEEE Proceedings on Aerospace Conference, vol. 2, pp. 2–991. IEEE, New York (2001)
26. Megla, E.C., Konangi, V.K., Wallett, T.M., Bhasin, K.B.: Comparison of IEEE 802.11 and wireless 1394 for intersatellite links in formation flying. In: 2002 IEEE Aerospace Conference Proceedings, vol. 3, pp. 3–1077. IEEE, New York (2002)
27. Vladimirova, T., Sidibeh, K.: WLAN for earth observation satellite formations in LEO. In: ECSIS Symposium on Bio-Inspired Learning and Intelligent Systems for Security, 2008, BLISS'08, pp. 119–124. IEEE, New York (2008)