A Survey of Delay- and Disruption-Tolerant Networking Applications

Artemios G. Voyiatzis, *Member, IEEE*

***Abstract*— Delay- and Disruption-Tolerant Networking (DTN) is a new communication paradigm that can span across multiple networks and cope with harsh conditions not envisioned in the Internet model. After more than ten years of active research in the field, numerous implementations and applications have emerged with a wide variety of performance and application domains. In this survey, we summarize recent developments in the field and highlight those areas with high potential for future development.**

***Index Terms–* Delay Tolerant Networking, Disruption Tolerant Networking, DTN, survey.**

I. INTRODUCTION

THE COMMUNICATION model of the Internet is based on some inherent networking assumptions. These include the existence of a continuous, bidirectional end-to-end path between two nodes; the relatively short round-trip delays; the symmetric data rates; and the low error rates. These assumptions led to the design of a *store-and-forward* approach: intermediate nodes receive small fragments of information (packets) and forward them to next hop as fast as possible. Each packet is only transiently stored in a network device.

In the late 1990s, the research community begun to explore how the Internet approach can fit into space communications for realizing interplanetary and deep space connectivity, starting with the Interplanetary Internet project (IPN) [1], [2]. The work initiated by the fact that the best-case sustainable throughput from Earth to Mars using TCP ranges between 1,600 bps and 250 Kbps, due to network stack limitations such as transmission timeouts [3]. Space links exhibit large bit error rate (BER) of 10^{-9} to 10^{-7}; space-to-earth BER is of 10^{-4} to 10^{-6}; and weather conditions cause 5% of transmitted frames to be lost [4]. Due to planet orbits, spacecraft movement, and harsh space conditions, a continuous communication path between two space nodes may be available only for a few minutes, with enormous round-trip delays. In an extreme, an end-to-end path among some nodes may not be available at any given time moment.

The work in the IPN project spawn Delay-Tolerant Networking, a new communication model with emphasis on deep-space communications. It was soon realized that networking in such challenging environments could be of use in (wireless) terrestrial applications, both for military and civilian applications. In this setting, delays are not caused by large propagation delays but rather by communication disruptions, intentional or not. In 2002, the IETF formed the DTN Research Group (DTNRG) with the objective to extend the concepts into an architecture for Delay- and Disruption-Tolerant Networks (DTN). Actually, the main difference between space and terrestrial environments can be accredited to the fact that space contacts are scheduled and predictable while terrestrial ones are more opportunistic in nature.

The TCP/IP protocol suite has served well the Internet until today. However, there are new environments and applications, where the Internet protocols perform poorly or cannot be used at all. In such environments, a DTN approach can offer a viable alternative for realizing communications. A point of concern for DTN is that a killer application is yet to be found and thus, it cannot unleash its full potential [5].

Some earlier surveys on DTN focus mainly on architecture and routing issues of DTN, with little emphasis on usage scenarios and applications [6], [7], [3], [8]. In this survey, we review the recent advances in realizing the DTN potential for space and terrestrial applications. The aim of the survey is to present the available options and performance issues in implementing DTN functionality today. Further, to demonstrate the rich set of applications and environments that DTN has already been used with the hope to foster future use of the technology. We note that the now-successful TCP/IP did not have a killer application 11 years after its invention [9].

The rest of the paper is organized as follows. Section II provides a review of the DTN terminology, architecture, and protocols. Section III presents the available platforms for implementing DTN in various environments; Section V summarizes the tools for emulating and simulating at large DTNs; and Section IV recaps real tests relating to network performance of DTN nodes. Section VI summarizes actual satellite experiments held or planned involving DTN and Section VII focuses on experiences in terrestrial applications and settings. Finally, Section VIII concludes the survey and discusses some open issues in DTN research.

II. DELAY-TOLERANT NETWORKING

A. Delay-tolerant network architecture

A delay-tolerant network (DTN) is a *network of regional networks*: it is an *overlay* on top of regional networks, including the Internet [10]. The communication characteristics are relatively homogeneous in a communication *region*. The wireless DTN technologies may be diverse, including not only radio frequency (RF) but also ultra-wide band (UWB), free-space optical, and acoustic (sonar or ultrasonic) technologies.

Each region has a unique *region ID* which is knowable among all regions of the DTN. DTN gateways have membership in two or more regions and are the only means of moving messages between regions.

Region ID use the same name-space syntax as the Internet's Domain Name System (DNS). Each node has a two-part name, consisting of a *region ID* and an *entity ID*. Routing *between* regions is based only on region ID while routing *within* a region is based only on entity ID.

B. Bundle layer

The unit of information exchange in a DTN is a *bundle*. A DTN *node* is an entity with a bundle layer. A node may be a *host*, *router*, or *gateway* (or some combination) acting as a

Manuscript received April 8, 2012; revised May 22, 2012.

A.G. Voyiatzis is with the Industrial Systems Institute/RC Athena, Patras Science Park building, Stadiou Str., Platani GR-26504, Greece.

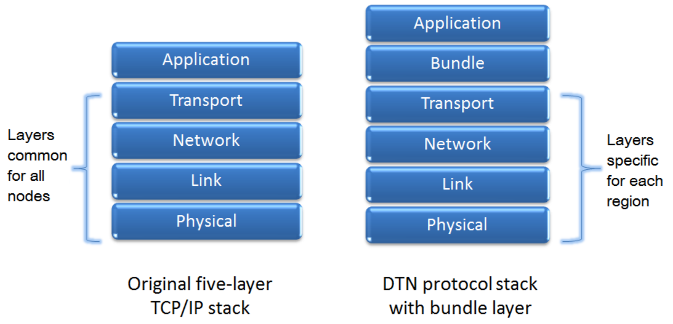

Fig. 1. DTN specific stack

source, destination, or forwarder of bundles. A router forwards bundles *within* a single DTN region while a gateway forwards bundles *between* DTN regions.

In a typical network, applications on different nodes communicate using a common set of network layers (such as TCP and IP). In a DTN, the bundle layer is placed below the application layer and hides the actual network- or region-specific communication layers, as depicted in Figure 1. A network-specific convergence layer is used underneath the bundle layer as to interface with each different network layer protocol used.

The Bundle Protocol (BP) is defined in RFC 5050 and implements a bundle layer in the DTN architecture defined in RFC 4838. The Bundle Protocol provides six classes of service (CoS) for a bundle, namely: Custody Transfer, Return Receipt, Custody-Transfer Notification, Bundle-Forwarding Notification, Priority of Delivery (bulk, normal, expedited), and Authentication.

In DTNs, forwarding nodes (routers and gateways) can be authenticated. Also, the sender information is authenticated by forwarding nodes, so that network resources can be conserved by preventing the carriage of prohibited traffic at the earliest opportunity.

The unique characteristic of the bundle layer is the support for *in-transit* storage. Bundles received from a sender can be stored in an intermediate node for an excessive amount of time (minutes, hours, or even days). These store operations are performed by the network stack, at the bundle layer, transparently to the application. The in-transit storage is the means to overcome the delays and disruptions induced while a bundle moves hop by hop to its final destination; to avoid costly end-to-end retransmissions due to errors or timeout; and to allow exchange of information between two nodes that share no end-to-end communication path at any given time moment. The bundle protocol defines a custody operation which allows an intermediate node to handle bundle delivery to final destination on behalf of a more distant sender.

C. *Design issues and concerns*

The DTN architecture originates from the work on the Interplanetary Internet. It was then extended and popularized to include challenged networks in general [2]. It is now considered as *the single solution* for all DTN scenarios by DT-NRG footnote`http://www.dtnrg.org/`, which develops the Bundle Protocol. The Bundle Protocol "*can be described as a complex extensible container format, with optionally secured payloads, carried by the supporting local network infrastructure*" [11].

There is some scepticism on the applicability of DTN and the bundle protocol (BP) in particular. Why two different DTNs will need to communicate, how the BP is a better approach, and why existing Internet technologies are not appropriate even for space applications are open questions [11]. Clearly, the existing design contains inherently some architectural problems that cannot be addressed [12]. The major critique to DTN architecture is the lack of an end-to-end reliability mechanism and lack of error detection at the bundle layer [12]. Other points of concern include time synchronization, fragmentation, and metadata parsing complexity. We review these issues in the following paragraphs.

1) Reliability and error detection: Error detection is left out of the BP design for both headers (bundle metadata) and payload data (bundles); the application is responsible to perform any error detection, if requirements dictate so. However, an application layer entity cannot know how to detect errors in BP headers (metadata), since they are not accessible to it.

End-to-end reliability must be implemented in the bundle layer because no single transport-layer protocol operates end-to-end across a DTN. Only regional reliability can be ensured by some transport layers e.g., TCP. We note that in-order data delivery is not totally guaranteed even in the case of TCP according to [13]. Even in the case a bundle crosses two regions that use different reliable transport layers, there is always the possibility that an error is introduced in the gateway node while storing or moving the bundle; such an error will go undetected at the destination.

Furthermore, there are cases where custody transfer is used to improve delivery. In such cases, the bundle may disappear altogether from the network if the custodian node does so and no other node (including the sender) is aware of the bundle's existence. The BLER protocol is an attempt to provide a solution to the "missing bundle" problem: the sender keeps a timer for each bundle sent and retransmits the bundle once the timer expires [14]. Additionally, BLER offers a guaranteed transfer for critical data and is currently available in the ION implementation (cf. next sections).

2) Time synchronization: The Bundle Protocol specification assumes that all nodes share a common clock so that the timestamps can be interpreted and handled correctly. This requirement is not necessarily practical or deployable [12]. Interestingly enough, a time protocol based on bundle exchange cannot be used to learn the correct time since the very bundles asking or carrying the time may be considered expired or invalid and thus, discarded [15].

Space communications assume connectivity on scheduled contacts, so at least neighboring nodes have a common sense of time or else they cannot meet [16]. However, there are two responses. The first is that once we move towards multi-agency missions and complex connectivity scenarios, DTN nodes of different agencies will have to agree on a common time. From a security point of view, which agency's time is correct and if an advertised one is acceptable by the respective policy is unclear. The second and more important one is that DTN is designed not only for space missions but for terrestrial applications too; how the time can be synchronized across different DTN nodes and technologies and if adjustment is allowable and technically feasible is not clear.

3) Fragmentation: The Bundle Protocol allows bundle fragmentation, both proactive and reactive. This is a welcomed functionality, as different regions and networking technologies may be willing to handle different maximum sizes and respective mechanisms may or may not be available at the underlying layers.

The DTN architecture and the BP do not define a mechanism to advertise and/or negotiate the maximum bundle size a node can accept, both for transmission and for storage [12]. We put emphasis on storage, since the large delays and opportunistic contacts can generate storage contention in otherwise cooperative nodes. Since contention cannot be advertised, precious bandwidth may have been consumed until the time a sender is notified that the bundle cannot be accepted for further processing on a congested node.

The deep interactions of bundle fragmentation and the rest of DTN functionality is not yet clearly understood. Problems can arise if bundle security is activated and bundles are fragmented and also in scenarios that involve custody transfers. Emulations and real-world experiments have shown that the fragment size must be carefully selected, as it definitely affects network performance [17].

4) Parsing complexity: The bundle protocol can be used to transfer pieces of information with large and variable size. It also has metadata fields with variable lengths, such as addresses in URI format. The Self-Delimiting Numeric Values (SDNV) format, described in RFC 6256, was chosen to encode arbitrary-length non-negative integers and arbitrary-length bitstrings with minimum overhead.

The usage of SDNV format minimizes the parsing complexity. A single bit of each byte indicates if the field continues to next byte or not. This approach reduces significantly the information and processing overhead at a DTN node. However, it also makes at the same time the protocol more susceptible to parsing errors due to corruption, since a single bit flip will affect both current and all following fields. We do not consider this fact as a disadvantage of the format; it rather highlights the need for an error detection mechanism at the bundle layer.

D. Applicability of technology

A lot of concern is expressed on the applicability of technology, in both space and terrestrial environments, mainly due to the lack of a "killer application" [5]. The major concern relates to terrestrial settings. There, a rich set of applications and services has already been deployed and the transition path to a DTN approach is not clear. In order to make applications "DTN-ready", one must devote significant effort developing additional functionality and integrating it into them. Furthermore, the penetration of cellular and wireless networks is rapidly increasing and the world becomes fully connected. It is unclear what an architecture that overcomes lack of connectivity can offer in such an environment.

In a connected-world scenario, network outages *do occur*, in some cases for long times. Even in always-connected settings, variations and fluctuations in network capacity occur and can result in disconnections. A network architecture that inherently deals with disruptions is clearly better [9]. Furthermore, there are cases where network infrastructure cannot be installed or it is not economically feasible to. Providing some kind of connectivity can be useful in these cases. Also, social network and community interactions do not always require Internet connectivity; they can take place on a local level, like the commuting bus or train [9]. Last but not least, we note that even in environments with rich connectivity (such as an urban setting), a large-scale network (for example, sensor or crowdsourcing applications) may incur prohibitive high costs to operate, in financial and energy costs. In such cases, it may be better to trade an "expensive always-on" connectivity in favor of a lower cost alternative, at least for non-real-time critical applications.

III. DTN PLATFORMS

Systems and applications that cope with delays and disruptions have been engineered for many years now. The standardization efforts concluded in the basis of the DTN standards, namely the RFC 4838 for DTN architecture and the RFC 5050 for the bundle protocol specification. In parallel, development efforts realized DTN implementations for specific system needs, platforms, and processing architectures. These implementations share varying (or not at all) compliance with the aforementioned standards. We note that an agreement was reached in IETF-62 meeting that all future DTN-compliant implementations would need to comply with the upcoming specification. We review in next those implementations.

A. Reference implementations

An early-stage DTN-RI (Reference Implementation) was developed by Trinity College Dublin, Ireland [18]. It was ported to Linux, Solaris, Win32 (through Cygwin environment), Linux on PDA (ARM), FreeBSD, and Mac OS X. DTN-RI was developed primarily in C++, included a discrete event simulator for prototyping and testing, and some sample applications like `dtnping`, `dtnsend/dtnrecv`, and `dtncp`.

The DTN-RI codebase evolved into DTN2 that is now the DTN reference implementation. DTN2 is written in C++ and tested on Linux (x64 and 64-bit x86) and Mac OS X (PPC and x386). It is available under an open-source license and it is hosted on SourceForge code repository. DTN2 supports table-based routing, Bonjour, ProPHET, DTLSR, and epidemic routing. It also supports external routing via XML messaging. HBSD[1] (implemented in C++) and RAPID (implemented in Java) are two examples of external routers for DTN2 [19]. Neighbor routing is also supported but it is expected to be removed from the code soon. DTN2 also supports `tca-router`, a specialized `TableBasedRouter` where the route table is manipulated in response to certain control bundles. It is based on the Tetherless Computing Architecture (TCA) layering control-over-data.

The current version of DTN2, namely 2.8.0, supports the TCP, UDP, NORM, AX.25, and Bluetooth convergence layers (CL). An Underwater Convergence Layer (UCL) has also been demonstrated using the External Convergence Layer XML-based functionality of DTN2 [20]. DTN2 can also be built against the `LTPlib` software as to support the LTP convergence layer[2]. DTN2 implemented an Ethernet CL in the past but it is no longer supported. A file CL is described but it is not yet implemented; a serial (RS232) CL is available.

The current version introduced support for the IPN naming scheme (needed by CBHE), for the CBHE functionality itself, and for the Age Extension Block, as described in draft-irtf-dtnrg-bundle-age-block-01. DTN2 provides some partial support for the Bundle Security Protocol (BSP) as defined in

---

[1] `http://planete.inria.fr/HBSD_DTN2/`

[2] LTPlib is available at `http://down.dsg.cs.tcd.ie/ltplib/`

RFC 6257. The implementation is based on the OpenSSL library. However, most functionality for handling cryptographic operations is absent.

For the sake of completeness, we mention DTN1, a prototype implementation in C that was tested on both x86 and StrongARM versions of Linux. This software is no longer in active development and has been superseded by the DTN2.

B. ION, a space-oriented implementation

The work at the JPL for the IPN project spawned the ION project at the University of Ohio, USA. ION is a software implementation of the bundle protocol stack and routing functionality suitable for a space environment. ION provides an implementation in the C language of the Bundle Protocol (BP); the Licklider Transmission Protocol (LTP); the CCSDS protocols CFDP and AMS; and the Contact Graph Routing (CGR) algorithm, which is considered to be the most suitable for space contacts.

ION is tested on Linux, Mac OS X, FreeBSD, Solaris, RTEMS, and VxWorks. It supports TCP, UDP, and LTP as convergence layers. The Saratoga space protocol has also been demonstrated to work with ION [21]. The ION software was used in all four (until now) spaceflights tests involving DTN (cf. next sections). Furthermore, ION is interoperable with DTN2. Earlier versions of the ION software were distributed by a somehow restricted “Open Channel Software/Caltech Licence (OCS) version 4.0”. Starting from version 2.5, the ION software is distributed through the SourceForge repository under a BSD license. Thus, the software can be distributed freely provided that its copyright notice remains unaltered.

C. IBR-DTN for OpenWRT routers

IBR-DTN is a very portable, slim, and extensible implementation in C++ of the bundle protocol [22], [23]. It is tested on Linux (x386 and MIPS) and runs on embedded systems and especially wireless access points (AP) with modified firmware based on OpenWRT. OpenWRT allows to support various hardware platforms, from smartphones up to laptops. IBR-DTN is developed on a Mikrotik Routerboard 532 using uClibc++. It was successfully tested on low-cost AP like Netgear WGT634U, Linksys WRT54G3G, and an ultra-low-budget FON FON-2200.

IBR-DTN includes support for TCP, UDP, and HTTP convergence layers and provides experimental support for Internet Drafts in the area of DTN. It also claims full compliance with the bundle security protocol (BSP) specification. IBR-DTN supports table-based routing; TCP and UDP discovery; IP neighbor; and epidemic routing with an efficient bloom filter.

D. Smartphones implementations

Smartphones and other personal portable devices can be involved in DTN scenarios, especially the ones that are based on social connectivity. Furthermore, they can harness multiple communication opportunities, like for example via WiFi, Bluetooth, USB, and 3G cellular networks. Usually these devices have specialized and/or proprietary operating systems and general-purpose DTN implementations cannot be used. Thus, a development effort is required to provide DTN functionality for them.

Bytewalla is a DTN Bundle Protocol implementation in Java by KTH, Sweden for the Android devices [24]. The initial concept scenario for Bytewalla is that people carrying an Android mobile phone travel between African rural villages and act as “data mules”. Bytewalla was later ported in pure Java for Microsoft Windows and Linux systems by the University of Wisconsin as JavaDTN [25].

DASM is a DTN implementation for Symbian phones [26]. It was tested with Nokia Communicators 9300i and 9500 and its development has been superseded by DTNS60. Although DTNS60 is a follow-up work on DASM, it is a complete redesign. DTNS60 implementation targets Symbian S60 smartphones (Maemo tablets). The implementation was tested on a Nokia N95 8 GB and a Nokia E90 [27]. DTNS60 supports bundles up to 1 MB and does not provide support for fragmentation and reassembly functionality [27].

DTNS60 includes “DT-Talkie”, a DTN-enabled application[3]. DT-Talkie was tested on Nokia N800 and N810 Linux-based Internet Tablet OS 2008 (Maemo 4.0).

IBR-DTN was ported on the OpenMoko platform [28] to support open-source smartphones like the NEO FreeRunner[4].

Finally, an iPhone implementation in Objective-C has been reported but is not publicly available[5].

E. Sensor networks

Wireless sensor networks (WSNs) are considered as good application of DTN; a WSN can be a “region” in DTN terminology using DTN gateways to transfer through the Internet sensed information and alerts. DTNLite provides a reliable transfer mechanism for sensor networks [29]. The architecture implementation for TinyOS is realized on Mica motes. Given the severely constrained environment of a Mica mote, the implementation follows the DTN architecture but does not implement the heavy-weighted bundle protocol.

Contiki is a lightweight, portable, multi-tasking operating system with an event driven kernel and includes a TCP/IP stack. Contiki is targeted for tiny devices that have severe memory and other resource constraints. ContikiDTN is an implementation of DTN for sensors running the Contiki operating system [30]. It was demonstrated to interoperate with a DTN2 implementation running on a personal computer using the TCP convergence layer.

An iMote2 sensor was hacked to run the Linux operating system (OpenEmbedded) and then the IBR-DTN was ported on top of it [23]. It is the (physically) smallest device to run IBR-DTN and supports only the IEEE 802.15.4 (LowPAN) as a convergence layer.

F. POSTELLATION for embedded systems

POSTELLATION is a DTN implementation written in C for embedded systems. It runs on Windows, Mac OS X, Linux, and RTEMS. It is packaged for easy installation and includes an HTTP proxy enabling HTTP/HTTPS browsing over DTN.

POSTELLATION can run on a real-time operating system, such as RTEMS, for flight software requirements[6]. It supports TCP, UDP, and TCP-TLS convergence layers over both IPv4 and IPv6. The software is available under license.

---

[3] `http://www.netlab.tkk.fi/tutkimus/dtn/dttalkie/`.

[4] On 2 April 2009, OpenMoko canceled planned phones and will probably concentrate on other kinds of hardware, but will still support and sell the current Neo FreeRunner.

[5] `http://www.ietf.org/mail-archive/web/dtn-users/current/msg00275.html`

[6] `http://postellation.viagenie.ca/`

G. Other implementations

There have been some special-purpose implementations for DTN, like RDTN that is based on Ruby but has no support for a convergence layer [31]; pyDTN, a DTN simulator in Python and C++ by the University of Maryland, USA[7]; and BP-RI v1.0.1, a Java implementation compatible with version 4 of the bundle protocol Internet Draft that is no longer developed[8].

IV. PERFORMANCE EVALUATION

The DTN implementations must cope well with large delays and disruptions. Given the hostile network conditions that they face, it is important that an implementation takes full advantage of available system resources and network bandwidth during connectivity events in order to maximize the network performance. The efficient routing decisions is an important aspect and previous surveys have covered the topic exhaustively [6], [7], [32].

From a systems point of view, DTNperf_2 is a performance evaluation tool for DTN [13]. It is built in accordance to the Iperf network performance analysis tool and supports many DTN options like: custody transfer, sent window, and experiment setups based on time, volume, and file transfers. It produces useful statistics and log files that can be used to analyze and correlate the behavior of the system. DTNperf_2 started as an independent tool but it is now an integral part of the DTN2 source code.

A network traffic analysis tool can be of significant help for a developer, for example when porting DTN software on a different architecture, integrating a new convergence layer, or testing for interoperability. A Bundle Protocol dissector is contributed in the Wireshark network protocol analyzer[9]. It can decode bundle layer traffic over TCP and UDP convergence layers. An LTP decoder is also available.

There have been to day efforts regarding the performance evaluation of DTN systems. We review these in the following.

A. Performance of DTN2, ION, and IBR-DTN

The IBR-DTN software was developed with embedded systems in mind. As such, it has a very small footprint [22], [23]: a default build of DTN2 on a Linux machine results in a DTN daemon of almost 22 MB, while IBR-DTN occupies only 114 KB. The RAM usage for DTN2 is more than 40 MB, while IBR-DTN can accommodate as little as 4 MB. For comparison, a customized build of ION used on the GGBA chamber of the International Space Station (ISS) has size of only 1.5 MB (524 KB in compressed form) [33].

One important operation parameter of a DTN system is the use of a storage backend for saving and retrieving the bundles while in transit. A set of experiments was held comparing the performance of different backends; the main findings are summarized in the next [22], [23]:

- In disk-based storage and small payload sizes, IBR-DTN significantly outperforms DTN2; in fact its performance is similar to the one of memory-based storage.
- In memory-based storage, DTN2 outperforms IBR-DTN for very small bundle sizes. For bigger sizes, they are about the same. Both IBR-DTN and DTN2's performance are limited by the bundle frequency i.e., the processing per bundle is the limiting factor rather than the storage bandwidth limit.
- Neither IBR-DTN nor DTN2 succeed in saturating a 480 Mbps link when using bundle sizes from 5 bytes up to 500 KB even with memory-based storage. IBR-DTN tops at about 310 Mbps while DTN2 tops lower: 245 Mbps (memory) and 98 Mbps (disk). This clearly indicates a performance issue, especially for scenarios where large amount of data must be moved fast during high-speed short-lived contacts, like in the case of vehicular nodes joining a WiFi network.
- Experiments with OpenWRT RSPro boards with an SD storage (which is very slow) peaked at 251 Mbps for plain TCP connections. IBR-DTN achieved a throughput of about 34 Mbps using bundles of 1 MB.
- Replacing the embedded system with a more resourceful personal computer did not improve dramatically the performance: IBR-DTN topped at about 45 Mbps.

A second evaluation scenario involved testing IBR-DTN 0.6.3, DTN2 2.7, and ION 2.4.0 on an Athlon II X4 2.8 GHz Linux PC with 4 GB RAM running Ubuntu 11.04 with 1 Gbps Ethernet NIC [34], [35]. The scenario doubled as an interoperability test among the three implementations; all possible source/destination combinations were tested successfully. The main performance findings are: a) Raw TCP: 940 Mbps; b) IBR-DTN and DTN2 achieve more than 600 Mbps (memory-based), while ION achieves only up to 449 Mbps; c) the fastest combination is ION sender and IBR-DTN receiver; and d) the underlying storage system effectively limits the achievable throughput. We further note that version 2.8.0 of DTN2 implements transactions for sustaining operation among restarts. These transactions are continuously stored in the backend for each received bundle and thus, the performance of DTN2 may have worsen.

A third experiment analyzes the performance of IBR-DTN with a IEEE 802.15.4 convergence layer. The IEEE 802.15.4 standard allows a maximum packet size of 128 bytes. Deducting the header, it can transfer 115 bytes. Just a bundle protocol header can be bigger than this size and thus, it may be needed to split it in segments and reassembled it at the destination. Accounting for the smallest BP header, the maximum bundle payload can be a mere 40 bytes. So, two thirds of each packet are lost in headers and only one third is payload (40/115). It is clear that BP header compression can be helpful in such a scenario. The raw IEEE 802.15.4 data rate in the iMote2 with Linux was 4.6 Kbps. This is way lower than the theoretical maximum of 250 Kbps [36].

A performance analysis on a DTN2 testbed for low-power, low-cost computers also disclosed a correlation between the size of the bundle and the attainable throughput [37]. The authors used a computer with 266 MHz processor and 256 MB of SDRAM and experimented with ATA-6 notebook HDD (40 GB; 5,400 rpm; transfer rate 100 MB/s) and a Compact Flash card (4 GB; 19.4 MB/s). They conclude that CPU is incapable of coping with NIC interrupts (IEEE 802.11a at 5.8 GHz) and hypothesize the maximum average throughput to be 3 MB/s (24 Mb/s). The CPU was used at full capacity and workloads of 50 KB (the maximum allowed bundle size for in-memory storage in DTN2) were transferred on average 23.8 times faster than bundles of 2 KB. This is an effect of reduced per-bundle overhead processing. Testing with larger bundle sizes requires additional disk accesses. Compact flash exhibited poorer performance, yet it may be beneficial for

---

[7] `http://www.umiacs.umd.edu/~mmarsh/pydtn/`

[8] `http://irg.cs.ohiou.edu/ocp/bundling.html`

[9] `http://www.wireshark.org`

energy-limited environments. The recommendations are i) use largest possible bundle size for in-memory storage; ii) increase parallelism; and iii) invest more on CPU rather than faster disk or more memory. Finally, they propose to allow for bundle receipt aggregation because each bundle receipt is tiny, only 54 bytes, and a lot of overhead processing is required for them.

B. Effect of bundle security

The bundle security protocol (BSP) has only recently reached a final status, as RFC 6257. It introduces four new blocks for bundle protection, namely Bundle Authentication (BAB), Payload Integrity (PIB), Payload Confidentiality (PCB), and Extension Security (ESB). BAB is a point-to-point authentication block, PIB and PCB protect the payload in transit from a security-source to a security-destination, and ESB is used to protect any extension blocks that may be present in a bundle.

The effects and overheads of security in performance were studied in one work [24]. In this work, an implementation of the `PCB-RSA-AES128-PAYLOAB` BSP block is described for the Android 1.6 platform. The author ported the DTN2 C++ code on the Android platform and used the BouncyCastle library for cryptographic operations. He implemented the required AES-GCM encryption algorithm, since it is not provided by the library.

The performance analysis on the smartphone platform indicated a small transmission overhead of only 0.007 seconds, because the program had an average transmission bandwidth of only 50 KB/sec and a constant size overhead of 372 bytes independent of payload block size. However, there was a large battery overhead for encryption and decryption operations (40% and 76%) due to cryptography. The increase can be accredited to BouncyCastle specific implementation of cryptography and to the capabilities of the Android 1.6 device. The less the bundle size, the more the increase percentage, probably due to the contribution of public-key encrypt/decrypt operations to the overall processing. It was also observed that encryption costs less than decryption, which can be accredited to the fact that encryption operations use a small public exponent of 17 while a large private exponent is used for decryption operations. The conclusion is that bundle security despite requiring a small transmission overhead, introduces large processing overheads, at least for PCB and possibly PIB and ESB blocks too.

C. Effect of convergence layers on performance

The bundle protocol allows transmission of bundles with variable size. The effect of bundle size, fragmentation, and selection of convergence layer parameters on the performance of a DTN system is attracting the interest of the research community for both space and terrestrial applications.

There is not yet a commonly acceptable bundle size for space DTN: Ivancic *et al.* used bundles of 160 MB while Burleigh suggests small bundles, less than 64 KB long, to enable partial data delivery at application [38]. An investigation of the packet size impact on DTN performance is described in [17]. The scenario involves a 10 MB file transfer from Mars to Earth through an error-prone channel with propagation delay of 10 minutes using the LTP convergence layer. The authors experimented with bundle size, LTP block size, and LTP segment size on an ION-based implementation. They show that the segment size influences overhead greatly, while bundle size has a major impact on memory occupancy and release (smaller bundle released memory faster). The transmission time was not significantly influenced by either segment or bundle size.

The effect of cross-layer PDU size adjustment was studied in [39]. The authors simulated a space link in the `ns-2` tool and use the DTTP convergence layer [40]. They further assume that each DTTP packet fits within one data link layer frame. In general, the best performance (memory-wise) is achieved for bundles of 5 KB with 750 byte DTTP packets. For big bit error rates of 10^{-5} the "best" size seems to be between 500 and 750 bytes. Worse performance is shown for bigger (as expected) and lower (not expected) sizes.

In a vehicular DTN environment (VDTN) with sporadic contacts, fragmentation is shown to allow the transfer of large messages that do not fit in the short contact duration [41]. Reactive fragmentation achieves real-time adaptation to the duration of contact but requires more processing. Proactive fragmentation is harder to adjust a priori but may perform slightly better than reactive if the size is properly adjusted. Proactive fragmentation is inefficient compared to reactive for short fragments due to increased overheads (numerous fragments and per-fragment overhead) affecting delivery latency.

An IBR implementation using HTTP as a convergence layer is described in [42]. The performance analysis shows 72.9 and 23.5 Mbit/s (downstream and upstream). Although a performance penalty is incurred by the bundle layer and the heavy HTTP convergence layer, the huge performance drop is actually due to the need to store large bundles in system's disk. The plain IP throughput is 90.4 Mbit/s (measured with Iperf) and in RTT experiments, any payload less than 100 KB had no influence on the achieved performance.

D. Performance on AX.25 networks

AX.25 is a link layer protocol for packet radio networking over HF, VHF and UHF links of 1,200 bps. It is used primarily by radio amateurs. The AX.25 Connected Mode Convergence Layer (AX25CM-CL) is a convergence layer implementation for DTN2, using `AF_AX25 SOCK_SEQPACKET` sockets on Linux platforms [43], [44], [45].

A set of experiments compares performance of DTN with TCP-CL and NORM-CL to IPv4 TCP/IP in different network configurations: control, point-to-point, single hop, and double hop [43], [46], [47], [44], [45]. The experiments showed that there is a significant difference in throughput compared to the theoretical model and this can be accredited to the necessary buffering and host-to-TNC (Terminal Node Controller) data transfers [47]. In any case, AX.25CM-CL exhibits better performance characteristics compared to TCP-CL in such challenged environments.

In the performed experiments, NORM exhibited quite robust behavior, even under hostile conditions (severe winds) [44]. The experiments showed that better performance is achieved upon configuring NORM's transmission rate to 84 Kbps instead of the default rate of 128 Kbps [44]. The DTN TCP-CL average throughput is slightly less than IPv4, but DTN NORM-CL performs significantly (almost 15%) better in point-to-point connections, despite using the unreliable UDP as transport. In the single-hop setup, the throughput for all CL dropped in about half. Again, NORM-CL outperformed TCP-CL by 40%. In the double-hop setup, the throughput of all dropped an additional 35-55%. In this case, NORM-CL suffered the less total degradation (68% compared to 82% of

Iperf and 78% of TCP-CL) and clearly outperformed TCP-CL by about 55% [45].

V. EMULATION AND SIMULATION

The simulation and emulation of DTN is a necessary activity for testing and validating new ideas and concepts in such challenged environments of operation. In the following, we review simulation and emulation environment reported in the literature. There is a vast variation of capabilities and focus of each environment and given the size and complexity of such networks, one must carefully select the appropriate tool to fit their needs.

A. DTN routing simulators

There are four mainstream simulators for DTN routing: NS2, OMNET++, DTNSim, and ONE [48]. NS2 and OMNET++ are general-purpose simulators that have been extended to a degree for supporting DTN. Similar claims hold for OMNET++. DTNSim supports only two scenarios, routing in a remote village and a network of city bus. Although ONE is superior in supporting DTN compared to the other three, it is still deficient in topics such as low-degree accuracy of time slots, inadequate processing capability causing performance degradation on large simulations, and lack of support for lower-level protocols.

B. VDTNsim

VDTN is a new disruptive network architecture where vehicles act as the communication infrastructure [49]. A VDTN simulator was developed by extending the ONE network simulator [50]. Then, a laboratory VDTN testbed, called VDTN@Lab which features laptops (as terminal and relay nodes) and a Lego Mindstorms NXT[10] coupled with an Asus PDA phone P527 that supports Bluetooth and IEEE 802.11 technologies (as mobile nodes) [51].

C. Terrestrial DTN testbeds

DOME is the testbed for experimentation with DieselNet in urban areas [52]. Its focus is on networks with moving nodes on scheduled routes, such as a buses. The N4C project has installed a terrestrial DTN testbed for experimentation in remote areas [53].

D. SPICE testbed

SPICE testbed is an emulation environment for testing space communications [54]. The architecture is described in [55]. Its design goals are the dynamic control of network parameters; scalability; transparency; and flexibility.

The testbed consists of 12 nodes installed in three different locations (two in Greece and one in the USA). It includes emulated links, real and modeled protocol implementations, and enhanced with actual satellite link interfaces offered by HellasSAT. Network parameters can be adjusted by a Central Management System through a GUI. Network emulation, in terms of bandwidth, PER (Packet Error Rate), corruption, duplication, re-ordering, and delay is realized using two tools: `netem`, the Linux network emulator module, and `tc`, the Linux traffic control tool, which in turn is part of the IProute2 package of tools.

The protocol stack in the testbed includes Ethernet, IP, and Space Packet Protocol (SPP) below DTN-specific protocols; CFDP (ack mode) over UDP, CFDP (unack mode) over UDP; the BP runs over LTP, TCP, and UDP; CFDP (unack mode), AMS, or third-party applications can run above the BP. The BP nodes support CGR, DTLSR, and PRoPHET routing. All these support a rich environment for emulating and testing various communication setups with different link types.

E. UTMesh

The UTMesh is a testbed for wireless mesh networking and delay tolerant networking developed by the University of Tokyo, Japan [56]. The testbed includes 51 nodes (September 2010) that are based on Armadillo-220 with 8 MB program memory and 32 MB working memory. The node has an ARM9 200 MHz CPU and runs Linux operating system and authors added storage for logging (USB stick) and a USB WiFi or Bluetooth interface for ad hoc networking. The rechargeable battery capacity (2100 mAh or 10 Ah) provides enough capacity for 24-hour experiments. The PEAR algorithm implementation footprint is about 3.000 lines of C code occupying about 51 KB of executable code [57]

F. TATPA, UCIT, and EGGS

The TATPA testbed[11] was built to evaluate performance of new TCP variants and new transport layer architecture, including DTN, over heterogeneous networks with at least one wireless section. TATPA is based on 11 Linux PC and can be accessed via a remote web interface to setup and conduct experiments. It was used for experiments with DTN performance on Nokia N900 smartphone [58].

In the course of the SatNEx project, TATPA was integrated in UCIT (University of Bologna - CNIT Integrated Testbed) to run experiments utilizing a real satellite link [59].

EGGS is another integrated testbed developed within the SatNEx project [59]. It consists of a proximity link (remote planet and orbiting satellite) and a long-haul link (orbiter to Earth communication) that connect a remote sensor network in another planet to an Earth collection station. ACE is used as a long-haul satellite link emulator [60]. DUMMYNET[12] is used as a proximity satellite link emulator.

In the simulation, the sensor network connects via TCP to the orbiter which in turn sends data to Earth using DTN2 over UDP. Performance problems are reported for the DTN stack in coping with high rates (10 measurements per second of 25 bytes each). Emulated links have throughput of 64 Kbps, the long-haul link propagation delay is between 125 ms and 200 seconds, and the proximity link propagation delays is 40 seconds.

G. ESA GSTVi

The European Space Agency (ESA) Ground Segment Test and Validation Infrastructure (GSTVi) integrates several simulators that are already used to test and validate ESA spacecraft operations [61].

GSTVi was used to study the CCSDS File Delivery Protocol (CFDP) and DTN architectures for future ESA missions [62].

---

[10] `http://en.wikipedia.org/wiki/Lego_Mindstorms_NXT`

[11] `http://tatpa.deis.unibo.it/auth/home.php?content=Description.html`

[12] `http://info.iet.unipi.it/~luigi/dummynet/`, similar to NetEm

In the short to medium-term time frame CFDP without DTN seems to provide the easier way to adoption due to its simplicity and maturity. The additional features of DTN, like dynamic routing or reactive fragmentation may only be required in the long-term. However, DTN does not seem to provide a solution in the long-term when more complex networks of interoperating assets in space and on ground (Earth and other planets) exist.

The simulation parameters were: uplink 10 Kbps Earth-orbiter and 500 Kbps for the orbiter to lander link. Latency of 1,200 sec (20 minutes) between Earth and orbiter and 1 sec between orbiter and lander. CFDP PDU 220/1,024 (uplink/downlink) and 10,240 bytes for CFDP/DTN uplink and downlink Earth-orbiter. Experiments with 100 KB files failed because the emulated Proximity-1 link based on UDP does not support bundle size greater than UDP datagram.

H. *xLuna*

There is a significant gap between the services offered by existing space-qualified Real-Time Operating Systems (RTOS) and those required by the most demanding future space applications. High-level services such as file systems and POSIX-compliant interfaces are required [63].

The xLuna project was carried out by Critical Software on behalf of ESA in the frame of the Portuguese task force. The xLuna project offers a reliable RTEMS and Linux kernel that could be used for both payload and avionics applications. The xLuna is based on RTEMS 4.6.6 for the privileged mode supporting hard real-time tasks (HRT) and a Snapgear Embedded Linux as a user-mode Linux for general-purpose applications, including DTN2, and speeding up the development.

The performance of the RTEMS is slightly lower when Linux is running. The performance of user-mode Linux is currently poor [63].

VI. SPACEFLIGHT TESTS AND EXPERIMENTS

DTN, as a descendant of IPN, is architected mainly for deep space communications. Efforts have been put to develop software and hardware that will operate in a space environment and demonstrate the validity of the technology.

The first reported results from testing DTN into space date back in 2008. In these tests, a bundle node sent images from a Low Earth Satellite (LEO) belonging to the UK Disaster Monitoring Constellation (UK-DMC) [15], [38]. Due to space limitations, the node onboard the UK-DMC satellite implemented only the bundle-forwarding functionality (cannot receive bundles) and used as a convergence layer the already deployed Saratoga protocol[21]. The ground segment consisted of two full DTN bundle agents based on the DTN2 software. The initial tests were successful and demonstrated the validity of the technology. The tests also revealed the need for reliability at the bundle layer and the problems that arise when clocks at different nodes start to drift. These architectural problems remain more or less unsolved until today.

A follow-up experiment occurred in 2009. This time, multiple ground terminals were involved, in the UK, the USA, Australia, and Japan [64]. This experiment was a successful demonstration of proactive and (by accident) reactive fragmentation: bundle fragments were received from Alaska and Hawaii ground stations and were then reassembled at a bundle agent in Ohio, USA. All ground agents ran DTN2 version 2.3.0 and the space agent ran the ION software.

The Deep Impact Network Experiment (DINET) used the EPOXI (formerly DI) spacecraft and Earth nodes to simulate an Earth-Mars network in October 2008 [65], [66]. The DINET technology experiment represents the first deep space implementation of DTN in flight and ground software [67]. In the course of the experiment, some 300 images were sent from the JPL nodes to the spacecraft and then returned back over a period of 27 days. The DINET experiment demonstrated the readiness of DTN for operational use in space missions. The flight software was a customized JPL ION version.

An essential component of DINET was the Experiment Operations Center (EOC), which acted as a central point of control for the DTN experiments [67]. The DINET EOC was located in JPL Protocol Technology Lab (PTL) and is part of the larger DTN Experiment Network (DEN). The EOC is using exclusively the ION software. DTN management actions were required to update the ION clock aboard EPOXI on a semiweekly basis to compensate for a drifting spacecraft clock, as well as to insert new routes after an unplanned power outage across the EOC network.

Two more spaceflight tests have taken place by NASA but no further information about the specifics of each test is available[13]. In both tests, the ION software is used. These tests are:

- Earth Observing One (EO-1) spacecraft in polar Earth orbit.
- DTN-over-IP via Cisco's Internet Router In Space (IRIS) operating on Intelsat 14 in geosynchronous orbit.

The University of Colorado in cooperation with NASA has designed, implemented, and tested DTN-on-ISS i.e., operation of DTN on the International Space Station [33]. The bundle router is installed on GGBA-5 which provides an embedded platform with a 1 GHz Intel Celeron 32-bit processor; 1 GB DDR SDRAM and 4 GB solid-state disk; and Debian Etch operating system based on Linux 2.6.21 kernel. The DTN node runs a custom JPL ION implementation. The first experiments completed successfully on June 2009. The specific application (transmission of telemetry files) enjoyed a clear benefit from using DTN; the previous scheme required on average 3504 redundant receptions per file while now this number dropped to just 0.06. Since no error detection was available in the bundle layer, the authors implemented a thin error detection layer based on CRC16 checksums beneath the bundle and atop the CCSDS layers. At the application layer, each file is accompanied by its `md5sum` hash which is checked for correctness upon file reception.

In parallel with ISS development, DTN nodes have been deployed in the ground segment, at the Huntsville Operations Support Center (HOSC) since 2009 [68]. The nodes are integrated in HOSC operating environment and are based on the DTN2 implementation. This setup allows to test for interoperability with the ION-based nodes in ISS. Also, it allows to test for multiple ground nodes at HOSC and Colarado University at Boulder, from where the GGBA on ISS is controlled.

As part of its METERON project for rover exploration in Mars, ESA explores the use of DTN for non-real-time communications through ESA's Columbus payload on ISS [61]. In this setting, the ESA/ESOC DTN nodes are virtual machines running Linux. The DTN nodes connect via VPN to NASA's

---

[13] `http://www.ietf.org/mail-archive/web/dtn-interest/current/msg04170.html`

DEN, which provides a link to the ISS and then a laptop within Columbus payload. ION is the software chosen by the METERON project.

VII. EARTH APPLICATIONS

A. Connectivity in developing areas

Provision of network and Internet connectivity for human-oriented communications in isolated areas includes many challenges, mostly concerning infrastructure availability and costs associated with installation and maintenance of the necessary equipment. The DakNet project is a DTN approach for providing network access to rural villages in India and Cambodia [69]. It was the first to introduce the concept of "data mule" i.e., an offline means, like a bus, a motorcycle, or even a bicycle, to move data between disconnected points. The idea was later extended to use "mechanical backhauls" and the DTN architecture in order to provide a solution with services such as naming, addressing, routing, and security [70]. KioskNet implemented these concepts into a working system and a pilot deployment provided valuable input to re-design of the initial prototype [71].

Electronic health services have a great potential to improve the delivery of health care in developing areas. The DTN approach can be beneficial for specific type of communications, like transmission of high-bandwidth, non-real-time information. A teleconsultation service based on DTN is already demonstrated in Ghana, utilizing Diaspora professionals [72], [73], [74]. A survey among health workers in low-resource settings indicated that there is enough interest in DTN-supported services to motivate further investigations [75]. The results suggest that there are at least some early adopters interested in experimenting with DTN in medical informatics systems context. Services such as access to email, notification of lab results, backup of Electronic Health Records (EHR), and teleconsultation seem to be the most promising to utilize DTN.

B. Disconnected areas

The Padjelanta national park in Laponia, Sweden is the home of the Saami population of semi-nomadic reindeer herders. The park lacks infrastructure and it is a UNESCO World Heritage site. Due to this, large antenna towers or other invasive infrastructure cannot be installed [76]. The Saami Network Connectivity (SNC) project explored the idea of DTN for providing basic Internet services to the herders [77]. The SNC project led to the N4C project[14] and similar experiments in isolated areas such as at Dharamsala in Indian Himalayan are held lately in the context of the ExtremeComm workshops [76].

Mines are another example where deployment of DTN can be beneficial. In the mining sites, it is cheaper to deliver messages down the hole carrying the mobile devices rather than building an infrastructure network [27]. A pilot DTN demonstration was held in ore mines of Finland using ALIX.3D boards and mobile phones [78].

C. Environment and wildlife monitoring

Environment and wildlife monitoring requires non-intrusive methods that require infrequent visits for maintenance and power charging for both interference and cost reasons. The ZebraNet project collected information about social behavior and movement of zebras within an area of 100 sq. km. in Kenya [79]. The ZebraNet sensors were installed in 2004 and provided valuable information on the hardware design for long-term, unmonitored operations [80].

The Environmental Monitoring in Metropolitan Areas (EMMA) project develops a decentralized and cost-efficient architecture for area-wide measurement of air pollutants in urban settings[15]. The project uses the public transport vehicles, such as buses and trams, to continuously collect environmental data; this information is exchanged among the vehicles using DTN techniques in order to minimize transmission costs. The IBR-DTN software spawned from the EMMA project [23].

Experiments with lake pollution monitoring in Ireland exploited DTN using boats as data mules [81]. This was a cost-effective approach compared to transmitting the collected information using cellular networks.

The MANA project tackled the problem of deploying a sensor network for year-round lake monitoring in North-East Greenland [82]. The authors provide valuable experiences on the harsh conditions met, the inability for operator activities, and the applicability of a DTN approach for such settings.

Sensors are deployed in agricultural areas for monitoring the weather and crop conditions. Cellular phones can be used to transmit the information to a central position but the operation cost is not always affordable. Tractors, vehicles, and farmers can be used as data mules to transfer collected information to a remote station. A field experiment was performed in the University of Tokyo, emulating an agriculturar scenario [57]. The experiment involved 39 embedded Linux nodes with a custom , non-bundle-protocol DTN build: an implementation of the Potential-based Entropy Adaptive Routing (PEAR) [83]. A success rate of 99.8% in data collection was observed with delays spanning between 10 and 75 minutes.

LUSTER (Light Under Shrub Thicket for Environmental Research) is a system that meets the challenges of Environmental Wireless Sensor Network (EWSN) systems using a hierarchical architecture that includes distributed reliable storage, delay-tolerant networking, and deployment-time validation techniques [84]. LUSTER has been evaluated in the laboratory, in a forested area, and in a deployment on Hog Island of Eastern Shore of Virginia. It uses a DTN component for in-network storage and bridges two wireless networks: IEEE 802.11 and IEEE 802.15.4. It does not implement the bundle protocol but rather it utilizes a query language and node storage using SD flash cards in order to recover missing measurements from the field.

D. Urban areas

Numerous projects and applications have been developed for demonstrating the applicability of DTN in urban settings. Despite the rich connectivity options, there are scenarios were it is not cost-effective (financially or power-wise) or no sufficient network coverage exists. In these cases, a DTN approach can be preferable.

User-provided Networks (UPN) can provide service to mobile users while sharing a broadband connection. Although the Internet connection is "always-on", connection sharing may be provided only in a "best-effort" approach e.g., when the user-provider is away or does not use it heavily. Home Access Points (HAPs) equipped with storage capabilities can provide

---

[14] `http://www.n4c.eu/`

[15] http://www.ibr.cs.tu-bs.de/projects/emma/

an elegant solution to this problem by exploiting the delay-tolerant networking principle [85].

DieselNet is a testbed of 40 buses with DTN nodes that cover an area of 150 sq. miles around urban area of Amherst, MA, USA [86]. DieselNet allows researchers to collect valuable information on connectivity patterns and to improve routing algorithms for DTN.

Content sharing and content distribution to specific target groups, such as commuters, is a target application for some DTN proposals [87], [88]. Examples include accessing Twitter [89] and performing web search from a bus [90]. BikeNet is another example application for collecting information from bicycles routes and rides [91]. BikeNet uses a dual mode of networking: DTN for normal operation and 3G connectivity for transmitting urgent signals. While one could claim that such functionality (traffic categorization) can be managed in the application layer instead of injecting another networking layer (the bundle), the benefits become clear once more than one applications with similar characteristics are used.

Smart Caching builds atop the DTN concept in order to provide continuous service during disconnection periods [92]. While DTN accommodates for delays (no timeouts), Smart Caching tries to retain service provision but proactive content caching during high-speed connectivity episodes. As such, it can work over existing network infrastructure installations (e.g., UMTS at motorways) without requiring to upgrade the network capacity and technology (e.g., WiMax).

Maritime communication environments can also benefit from a DTN approach [93]. The authors model the communication environment in the Strait of Singapore using WiMax links and actual trace data. The performed simulations show that DTN routing (Epidemic and Spray-and-Wait) outperforms classical ad hoc routing protocols (AODV and OLSR) in packet delivery rates on the expense of greater delay that DTN can cope with. Thus, DTN can be a better communication alternative for such a crowded environment.

E. Undersea communications

An Underwater Convergence Layer (UCL) was developed for the DTN2 Reference Implementation supporting the WHOI Micro-Modem [20]. It allows communication via acoustic signals (1,500 m/sec) compared to RF signals (300,000 km/sec). The implementation is based on DTN2's External Convergence Layer (ECL) that allows for the implementation of CLAs outside of the DTN2 code base. CLAs implemented using ECL interface exchange XML-encoded messages with the ECL via a TCP/IP socket.

A field test was held in Italy's Tuscan Archipelago in September 2010 by NATO Undersea Research Centre (NURC) in collaboration with University of Porto [94]. The suitability of the reference implementation to create networks composed of heterogeneous links (radio and acoustic) spanning above- and below-water domains was evaluated and the experiments were successful.

F. Military applications

Airborne Networks have a dynamic nature of topology that fits well with DTN [95]. Naval networks include multiple and heterogeneous links [96]. This is an ideal setting for a DTN approach, since the bundle layer can act as the unifying layer on top of link-specific convergence layers and networking technologies.

A proof-of-concept demo for Marine Corps CONDOR system is described in [97]. The payload of the CONDOR Gateway prototype unit consists of a Cisco 3725 router, an EPLRS radio, a link encryption module, a Cisco access router, and a satellite communication terminal. The DTN2 software is ported to CONDOR and loaded into the Cisco IDS module (a single-board computer with an Intel Pentium III processor, 512 MB RAM, 20 GB hard disk, and an Ethernet interface). CONDOR also added support for Space Communication Protocol Standard Transport Protocol (SCPS-TP) into the module, and implemented a DTN-enabled web proxy.

G. VDTN

VDTN is a new disruptive network architecture where vehicles act as the communication infrastructure. The VDTN architecture places the Bundle layer between the network and the data link layers (IP-over-VDTN) rather than between the transport and application layer as in conventional DTN architecture [49]. VDTN also separates the data and control planes; signaling messages carry information about node type and its speed, physical link data rate and range, energy constraints, storage capacity constraints, delivery options, and security requirements among others.

VDTN signaling at the control plane is exchanged via Bluetooth. This plane drives the creation of WiFi ad hoc networks by activating the wireless network interface for exchanging information at the data plane. A real-world VDTN experiment was later performed around of the Brazilian Fiat Automobile manufacturing plant [98].

A Content Storage and Retrieval (CSR) mechanism was implemented in VDTN [99]. The CSR mechanism defines the exchange of content labels on the control plane at each contact opportunity. These labels describe storage, forward, and retrieval policies of bundles exchanged on the data plane. The performance of CSR was evaluated in VDTN@Lab using the Epidemic and Spray-and-Wait routing algorithms. The metrics considered were *average bundle delivery probability* (percentage) and *bundle average delay* (in seconds). CSR provided better performance for both routing algorithms.

A Contact Prediction (CP) algorithm and a Contact Duration (CD) scheduling policy were tested in the VDTN simulator [100]. The idea is to exchange information via the long-range, low-bandwidth control plane in order to predict future contacts and their duration as to harness them for efficient transfers via the short-range, high-bandwidth data plane. In the simulated experiments, the ranges where 90 and 30 meters respectively.

The innovation of VDTN is that for non-TCP networks, it can implement a storage service **under** IP thus, it can allow traditional, IP-based networking approaches **over** delay-tolerant networks. Furthermore, it is the first architecture to separate the data (WiFi) from the control (Bluetooth) plane.

H. IP over DTN

Similar to VDTN efforts, IP over DTN is also proposed, mainly based on the PEAR framework [101], [83]. The architecture supports only small IP packet transfer (less than 1,500 bytes) and requires UDP as an upper layer, since it introduces great delays that TCP cannot cope with. Given that it operates at the data link layer, complex tasks such as fragmentation and reassembly and network routing are left to upper layers. The UTMesh testbed presented earlier is a demonstrator of this concept. Adopting a multi-path approach, even for static nodes, is shown to achieve scalable message propagation that increases as redundancy-level increases [102].

I. Bridging space and Earth

Emulation of satellite communications (which are between terrestrial systems where TCP/IP runs well and deep space systems where DTN runs well) shows that DTN with sliding window $W > 1$ over TCP and double connections outperforms TCP NewReno/SACK and TCP Hybla (and TCP-only approaches in general) [103].

Land Mobile Satellites (LMS) systems exhibit long round-trip delays and frequent disruptions due to obstructions, like tunnels or buildings [104]. This is also an ideal setting for applying DTN.

J. Traffic engineering

DTN can be used to automate very large-volume data transfers between two places. Although high-speed networks are available, in some cases it is even faster to physically move the storage media from one place to another rather than transmitting the stored information. The TrainNet approach explores this idea by placing racks of portable hard disks in trains and stations and then use the vehicles to transport latency insensitive data between stations [105]. A similar idea was earlier explored in a global scale, using air carriers and airports; there also the reachability problem is explored using the flight schedules among all cities [106].

NetStitcher is a system for bulk transfers between datacenters that are geographically dispersed across the globe [107]. An innovative characteristic of NetStitcher is that it utilizes prediction for future bandwidth availability among datacenters in different timezones, in order to efficiently move the data from one to another until the final destination, closing resembling the ideas of DTN. Both simulations and live deployment of NetStitcher on a real content delivery network (CDN) show an impressive fivefold bandwidth savings compared to other mechanisms.

K. Emergency relief operations

The ability of DTN to interface with multiple networking technologies, ranging from space to underwater acoustic communications renders it a viable choice for emergency relief operations. Currently, it can interface both classical TCP/IP networks and AX.25 links. DTN over AX.25 is proposed as an alternative that may allow for a more ad hoc, self-configuring network formation that is necessary on an emergency [47], [46]. The applicability and suitability of DTN on providing situational awareness using citizens' available infrastructure in emergencies is further explored in [108].

VIII. CONCLUSIONS

In this survey, we reviewed recent advances in developing DTN software, applications, and services. Since its inception more than decade ago for deep space communications, DTN has evolved in an architecture for any challenged networking environment. Despite the wide diversity of operating conditions, the DTN implementations achieve in providing sufficient performance for most scenarios. Network throughput is an area of future improvement, provided that applications needing wire speeds do emerge.

A rich ecosystem of software, systems, simulators, emulators, and testbeds is now available and researchers can experiment with more complex scenarios and pursue even harder problems. At the same time, new application domains are explored and innovative solutions are invented as more people are exposed to the technology. Although a killer application for DTN is yet to be found, more and more applications must cope with disruptions and intermittent connectivity even in the conventional Internet domain. It may be the case that not a single application will drive a wider adoption of DTN but rather the need for every application to continue operation in case the network disappears.

REFERENCES

[1] R. Durst, P. Feighery, and K. Scott, "Why not use the standard Internet suite for the Interplanetary Internet," Interplanetary Internet Study Seminar, California Institute of Technology, 1999.

[2] K. Fall, "A Delay-Tolerant Network Architecture for challenged internets," in *Proceedings of the 2003 conference on applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03)*. ACM, 2003, pp. 27–34.

[3] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges," *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, 2011, pp. 1–34.

[4] O. Gnawali, M. Polyakovt, P. Bose, and R. Govindan, "Data centric, position-based routing in space networks," in *Aerospace Conference, 2005 IEEE*. IEEE, Mar. 2005, pp. 1322–1334.

[5] A. Lindgren and P. Hui, "The quest for a killer app for opportunistic and delay tolerant networks: (invited paper)," in *Proceedings of the 4th ACM workshop on Challenged networks (CHANTS '09)*. New York, NY, USA: ACM, 2009, pp. 59–66.

[6] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 1, quarter 2006, pp. 24–37.

[7] Z. Zhang and Q. Zhang, "Delay/disruption tolerant mobile ad hoc networks: latest developments: Research articles," *Wirel. Commun. Mob. Comput.*, vol. 7, no. 10, Dec. 2007, pp. 1219–1232.

[8] P. Pereira, A. Casaca, J. Rodrigues, V. Soares, J. Triay, and C. Cervello-Pastor, "From delay-tolerant networks to vehicular delay-tolerant networks," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, 2011, pp. 1 –17.

[9] J. Crowcroft, E. Yoneki, P. Hui, and T. Henderson, "Promoting tolerance for delay tolerant network research," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 5, Sept. 2008, pp. 63–68.

[10] F. Warthman, "Delay-Tolerant Networks (DTNs): A tutorial," 2003, version 1.1, 3/5/03.

[11] L. Wood, P. Holliday, D. Floreani, and W. Eddy, "Sharing the dream: The consensual networking hallucination offered by the Bundle Protocol," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2009 International Congress on*. IEEE, 2009, pp. 1–2.

[12] L. Wood, W. Eddy, and P. Holliday, "A bundle of problems," in *2009 IEEE Aerospace Conference Proceedings*. Piscataway, NJ, USA: IEEE, 2009, pp. 1–17.

[13] C. Caini, P. Cornice, R. Firrincieli, and M. Livini, "DTNperf_2: A performance evaluation tool for delay/disruption tolerant networking," in *Proceedings of the International Conference on Ultra Modern Telecommunications, ICUMT 2009, 12-14 October 2009, St. Petersburg, Russia*. IEEE, 2009, pp. 1–6.

[14] E. Koutsogiannis, F. Tsapeli, and V. Tsaoussidis, "Bundle layer end-to-end retransmission mechanism," in *Proceedings of the 2011 Baltic Congress of Future Internet Communications (BCFIC 2011)*. IEEE, 2011, pp. 109–115.

[15] L. Wood, W. Ivancic, W. Eddy, D. Stewart, J. Northam, C. Jackson, and A. da Silva Curiel, "Use of the delay-tolerant networking bundle protocol from space," in *Proceedings of the 59th Astronautical Congress (IAC '08)*, 2008, paper IAC-08-B2.3.10. [Online]. Available: http://www.iafastro.net/iac/archive/browse/IAC-08/B2/3/1920/

[16] W. D. Ivancic, "Security analysis of DTN architecture and bundle protocol specification for space-based networks," in *2010 IEEE Aerospace Conference*. Piscataway, NJ, USA: IEEE, 2010, pp. 1–12.

[17] N. Bezirgiannidis and V. Tsaoussidis, "Packet size and DTN transport service: Evaluation on a DTN testbed," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on*. IEEE, 2010, pp. 1198–1205.

[18] M. Demmer, "The DTN reference implementation," presentation at IETF DTNRG Meeting - March 9, 2005. [Online]. Available: http://www.dtnrg.org/docs/presentations/IETF62/dtn-impl-ietf-mar05-demmer.pdf

[19] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '07)*. New York, NY, USA: ACM, 2007, pp. 373–384.

[20] D. Merani, A. Berni, J. Potter, and R. Martins, "An underwater convergence layer for disruption tolerant networking," in *Proceedings of the 2011 Baltic Congress of Future Internet Communications (BCFIC 2011)*. IEEE, 2011, pp. 103–108.

[21] L. Wood, W. Eddy, W. Ivancic, J. McKim, and C. Jackson, "Saratoga: a delay-tolerant networking convergence layer with efficient link utilization," in *Satellite and Space Communications, 2007. IWSSC '07. International Workshop on*, Sept. 2007, pp. 168 –172.

[22] M. Doering, S. Lahde, J. Morgenroth, and L. Wolf, "IBR-DTN: an efficient implementation for embedded systems," in *Proceedings of the third ACM workshop on Challenged networks (CHANTS '08)*. New York, NY, USA: ACM, 2008, pp. 117–120.

[23] S. Schildt, J. Morgenroth, W.-B. Pöttner, and L. Wolf, "IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation," *Electronic Communications of the EASST*, vol. 37, Jan. 2011, pp. 1–11.

[24] S. Domancich, "Security in delay tolerant networks for the Android platform," Master's thesis, Royal Institute of Technology (KTH), Sweden, June 2010.

[25] "JavaDTN: A java port for bytewalla DNT." [Online]. Available: https://wiki.umiacs.umd.edu/VirtualMeshTest/index.php/JavaDTN

[26] O. Mukhtar and J. Ott, "Backup and bypass: Introducing DTN-based ad-hoc networking to mobile phones," in *Proceedings of the 2nd International Workshop on Multi-hop Ad hoc Networks: From Theory to Reality (REALMAN '06)*. New York, NY, USA: ACM, 2006, pp. 107–109.

[27] A. Turkulainen, "Delay-Tolerant Networking for the open mobile software platform: An application case study," Master's thesis, Aalto University, Finland, Jan. 2010.

[28] O. Community, "The OpenMoko project homepage." [Online]. Available: http://www.openmoko.org/

[29] S. Nedenshi and R. Patra, "DTNLite: A reliable data transfer architecture for sensor networks," in *8th International Conference on Intelligent Engineering Systems*, 2004.

[30] M. Loubser, "Delay tolerant networking for sensor networks," Swedish Institute of Computer Science, Kista, Sweden, Tech. Rep. T2006:01, 2006.

[31] J. Greifenberg and D. Kutscher, "RDTN: An agile DTN research platform and bundle protocol agent," in *Wired/Wireless Internet Communications: 7th International Conference, WWIC 2009, Enschede, the Netherlands, May 27-29 2009, Proceedings*, ser. Lecture Notes in Computer Science, H. van den Berg, G. Heijenk, E. Osipov, and D. Staehle, Eds. Springer Berlin/Heidelberg, 2009, vol. 5546, pp. 97–108.

[32] M. Liu, Y. Yang, and Z. Qin, "A survey of routing protocols and simulations in delay-tolerant networks," in *Proceedings of the 6th international conference on Wireless algorithms, systems, and applications (WASA'11)*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 243–253.

[33] A. Jenkins, S. Kuzminsky, K. Gifford, R. Pitts, and K. Nichols, "Delay/disruption-tolerant networking: Flight test results from the International Space Station," in *2010 IEEE Aerospace Conference Proceedings*. Piscataway, NJ, USA: IEEE, 2010, pp. 1–8.

[34] W.-B. Pöttner, J. Morgenroth, S. Schildt, and L. Wolf, "Performance comparison of DTN bundle protocol implementations," in *Proceedings of the 6th ACM workshop on Challenged networks (CHANTS '11)*. New York, NY, USA: ACM, 2011, pp. 61–64.

[35] ——, "An empirical performance comparison of DTN bundle protocol implementations," Technische Universität Braunschweig, Informatikbericht 2011-08, 2011.

[36] S. Schmidt, "Delay Tolerant Networking on embedded Linux hardware," Master's thesis, Technische Universität Braunschweig, Germany, Aug. 2010.

[37] E. Oliver and H. Falaki, "Performance evaluation and analysis of delay tolerant networking," in *Proceedings of the 1st international workshop on System evaluation for mobile platforms (MobiEval '07)*. New York, NY, USA: ACM, 2007, pp. 1–6.

[38] W. Ivancic, W. Eddy, L. Wood, D. Stewart, C. Jackson, J. Northam, and A. da Silva Curiel, "Delay/disruption-tolerant network testing using a LEO satellite," in *8th Annual NASA Earth Science Technology Conference (ESTC 2008)*, 2008, pp. 1–7, paper A4P2, Proceedings available at http://esto.nasa.gov/conferences/estc2008/index.html.

[39] C. V. Samaras and V. Tsaoussidis, "Adjusting transport segmentation policy of DTN Bundle Protocol under synergy with lower layers," *The Journal of Systems and Software*, vol. 84, 2011, pp. 226–237.

[40] C. Samaras and V. Tsaoussidis, "Design of delay-tolerant transport protocol (DTTP) and its evaluation for Mars," *Acta Astronautica*, vol. 67, no. 7-8, 2010, pp. 863–880.

[41] N. Magaia, P. Pereira, A. Casaca, J. Rodrigues, J. Dias, J. Isento, C. Cervellló-Pastor, and J. Gallego, "Bundles fragmentation in vehicular delay-tolerant networks," in *7th Euro-NF Conference on Next Generation Networks (NGI 2011)*. IEEE, 2011, pp. 27–29.

[42] J. Morgenroth, T. Poegel, R. Heitz, and L. Wolf, "Delay-tolerant networking in restricted networks," in *Proceedings of the 6th ACM workshop on Challenged networks (CHANTS '11)*. New York, NY, USA: ACM, 2011, pp. 53–56.

[43] J. Ronan, K. Walsh, and L. Darren, "Experiments with delay and disruption tolerant networking in AX.25 and D-Star networks," in *28th ARRL and TAPR Digital Communications Conference*. 225 Main Street, Newington, CT 06111-1494, USA: ARRL, 2008, pp. 97–106.

[44] J. Ronan and C. O'Connor, "A comparison of different TCP/IP and DTN protocols over the D-Star Digital Data Mode," in *29th ARRL and TAPR Digital Communications Conference*. 225 Main Street, Newington, CT 06111-1494, USA: ARRL, 2010, pp. 134–138.

[45] ——, "A further comparison of different TCP/IP and DTN protocols over the D-Star Digital Data Mode," in *30th ARRL and TAPR Digital Communications Conference*. 225 Main Street, Newington, CT 06111-1494, USA: ARRL, 2011, pp. 72–79.

[46] J. Ronan, K. Walsh, and D. Long, "Evaluation of a DTN convergence layer for the AX.25 network protocol," in *Proceedings of the Second International Workshop on Mobile Opportunistic Networking (MobiOpp '10)*. New York, NY, USA: ACM, 2010, pp. 72–78.

[47] J. Ronan, L. Darren, and K. Walsh, "More experiments with delay and disruption tolerant networking over AX.25 networks," in *30th ARRL and TAPR Digital Communications Conference*. 225 Main Street, Newington, CT 06111-1494, USA: ARRL, 2011, pp. 65–71.

[48] M. Liu, Y. Yang, and Z. Qin, "A survey of routing protocols and simulations in delay-tolerant networks," in *Proceedings of the 6th international conference on Wireless algorithms, systems, and applications (WASA '11)*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 243–253.

[49] V. Soares, F. Farahmand, and J. Rodrigues, "A layered architecture for vehicular delay-tolerant networks," in *IEEE Symposium on Computers and Communications (ISCC 2099)*. IEEE, 2009, pp. 122 –127.

[50] ——, "VDTNsim: A simulation tool for vehicular delay-tolerant networks," in *15th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD 2010)*. IEEE, 2010, pp. 101–105.

[51] J. Dias, J. Isento, B. Silva, V. Soares, P. Ferreira, A. Nogueira, and J. Rodrigues, "Creation of a vehicular delay-tolerant network prototype," in *Conferlncia de Engenharia, 5, Covilh, 25 Novembro 2009 - Inovao e Desenvolvimento*. Covilh, Portugal: Universidade da Beira Interior, 2009, pp. 1–5.

[52] H. Soroush, N. Banerjee, A. Balasubramanian, M. Corner, B. Levine, and B. Lynn, "DOME: A diverse outdoor mobile testbed," in *Proceedings of the 1st ACM International Workshop on Hot Topics of Planet-Scale Mobility Measurements*. New York, NY, USA: ACM, 2009, pp. 2:1–2:6.

[53] S. Farrell, "Security in the wild," *IEEE Internet Computing*, vol. 15, no. 3, May 2011, pp. 86–91.

[54] E. Koutsogiannis, S. Diamantopoulos, G. Papastergiou, I. Komnios, A. Aggelis, and N. Peccia, "Experiences from architecting a DTN testbed," *Journal of Internet Engineering*, vol. 2, no. 1, Dec. 2009, pp. 219–229.

[55] C. Samaras, I. Komnios, S. Diamantopoulos, E. Koutsogiannis, V. Tsaoussidis, G. Papastergiou, and N. Peccia, "Extending internet into space – ESA DTN testbed implementation and evaluation," in *Mobile Lightweight Wireless Systems*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, F. Granelli, C. Skianis, P. Chatzimisios, Y. Xiao, S. Redana, O. Akan, P. Bellavista, J. Cao, F. Dressler, D. Ferrari, M. Gerla, H. Kobayashi, S. Palazzo, S. Sahni, X. S. Shen, M. Stan, J. Xiaohua, A. Zomaya, and G. Coulson, Eds. Springer Berlin Heidelberg, 2009, vol. 13, pp. 397–404.

[56] H. Ochiai, K. Matsuo, S. Matsuura, and H. Esaki, "A case study of UTMesh: Design and impact of real world experiments with Wi-Fi and Bluetooth devices," in *Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on*. Washington, DC, USA: IEEE Computer Society, july 2011, pp. 433–438.

[57] H. Ochiai, H. Ishizuka, Y. Kawakami, and H. Esaki, "A DTN-based sensor data gathering for agricultural applications," *IEEE Sensors Journal*, vol. 11, no. 11, Nov. 2010, pp. 2861–2868.

[58] C. Caini, P. Cornice, R. Firrincieli, M. Livini, and D. Lacamera, "DTN meets smartphones: Future prospects and tests," in *Proceedings of the 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC 2010)*. Piscataway, NJ, USA: IEEE Press, 2010, pp. 355–360.

[59] T. de Cola, L. Ronga, T. Pecorella, P. Barsocchi, S. Chessa, E. Ferro, A. Gotta, G. Oligeri, F. Potortì, R. Secchi, A. Sathiaseelan, V. Castro, R. Peral, C. Caini, and R. Firrincieli, "Communications and networking

over satellites: SatNEx experimental activities and testbeds," *International Journal of Satellite Communications and Networking*, vol. 27, no. 1, 2009, pp. 1–33.

[60] G. Albertengo and S. Petroianni, "ACE/2: a scalable modular SATCOM system emulator," in *Distributed Simulation and Real-Time Applications, 2002. Proceedings. Sixth IEEE International Workshop on.* IEEE, 2002, pp. 119–126.

[61] F. B. de Frescheville, S. Martin, N. Policella, D. Patterson, M. Aiple, and P. Steele, "Set-up and validation of METERON end-to-end network for robotic experiments," 2011, eSA/ESTEC, Noordwijk, The Netherlands, April 12–14, 2011.

[62] F. Flentge, "Study on CFDP and DTN architectures for ESA space missions," in *SPACOMM 2011, The Third International Conference on Advances in Satellite and Space Communications*. IARIA, 2011, pp. 75–80, iSBN: 978-1-61208-128-1.

[63] P. Braga, L. Henriques, B. Carvalho, P. Chevalley, and M. Zulianello, "xluna - demonstrator on ESA Mars rover," in *DASIA 2008 - Data Systems in Aerospace*, L. Ouwehand, Ed. ESA, 2008, sP-665 Proceedings. A more detailed version is available on http://asd.criticalsoftware.com/upload_product/wp/Real-Time_Dependable_Kernel_for_Embedded_Systems.pdf.

[64] W. Ivancic, P. Paulsen, D. Stewart, W. Eddy, J. McKim, J. Taylor, S. Lynch, J. Heberle, J. Northam, C. Jackson, and L. Wood, "Large file transfers from space using multiple ground terminals and delay-tolerant networking," in *IEEE Global Telecommunications Conference (GLOBECOM 2010)*. Piscataway, NJ, USA: IEEE, 2010, pp. 1–6.

[65] R. M. Jones, "Disruption tolerant network technology flight validation report," NASA JPL, Pasadena, California, Tech. Rep. JPL 9-2, Feb. 2009.

[66] J. Wyatt, S. Burleigh, R. Jones, L. Torgerson, and S. Wissler, "Disruption tolerant networking flight validation experiment on nasa's epoxi mission," in *Advances in Satellite and Space Communications, 2009. SPACOMM 2009. First International Conference on*, July 2009, pp. 187–196.

[67] J. Torgerson, L. Clare, S.-Y. Wang, and J. Schoolcraft, "The deep impact network experiment operations center," in *Aerospace conference, 2009 IEEE*, Mar. 2009, pp. 1–12.

[68] K. Nichols, M. Holbrook, R. L. Pitts, K. K. Gifford, A. Jenkins, and S. Kuzminsky, "DTN implementation and utilization options on the International Space Station," in *SpaceOps 2010 Conference*, ser. Conference Proceeding Series. AIAA, 2010.

[69] A. S. Pentland, R. Fletcher, and A. Hasson, "DakNet: Rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, Jan. 2004, pp. 78–83.

[70] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost communication for rural Internet kiosks using mechanical backhaul," in *Proceedings of the 12th annual international conference on Mobile computing and networking (MobiCom '06)*. New York, NY, USA: ACM, 2006, pp. 334–345.

[71] S. Guo, M. Derakhshani, M. H. Falaki, U. Ismail, R. Luk, E. A. Oliver, S. U. Rahman, A. Seth, M. A. Zaharia, and S. Keshav, "Design and implementation of the KioskNet system," *Comput. Netw.*, vol. 55, no. 1, Jan. 2011, pp. 264–281.

[72] R. Luk, M. Ho, and P. M. Aoki, "Asynchronous remote medical consultation for Ghana," in *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI '08)*. New York, NY, USA: ACM, 2008, pp. 743–752.

[73] R. Luk, M. Zaharia, M. Ho, B. Levine, and P. Aoki, "ICTD for healthcare in Ghana: Two parallel case studies," in *2009 International Conference on Information and Communication Technologies and Development (ICTD)*, Apr. 2009, pp. 118–128.

[74] R. Luk, M. Ho, and P. M. Aoki, "A framework for designing teleconsultation systems in Africa," in *Proc. Int'l Conf. on Health Informatics in Africa (HELINA), Bamako, Mali, Jan. 2007*, 2007, pp. 1–5.

[75] S. Syed-Abdul, J. Scholl, P. Lee, W.-S. Jian, D.-M. Liou, and Y.-C. Li, "Study on the potential for delay tolerant networks by health workers in low resource settings," *Computer Methods and Programs in Biomedicine*, vol. XXX, no. 0, 2011, p. in press.

[76] A. Lindgren and P. Hui, "ExtremeCom: to boldly go where no one has gone before," *SIGCOMM Computer Communication Review*, vol. 41, no. 1, Jan. 2011, pp. 54–59.

[77] A. Lindgren, A. Doria, J. Lindblom, and M. Ek, "Networking in the land of northern lights: two years of experiences from dtn system deployments," in *Proceedings of the 2008 ACM workshop on Wireless networks and systems for developing regions (WiNS-DR '08)*. New York, NY, USA: ACM, 2008, pp. 1–8.

[78] P. Ginzboorg, T. Krkkinen, A. Ruotsalainen, M. Andersson, and J. Ott, "DTN communication in a mine," in *Extreme Workshop on Communication - The Himalayan Expedition (ExtremeCom 2010)*, 2010, pp. 1–6.

[79] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. 5, Oct. 2002, pp. 96–107.

[80] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware design experiences in ZebraNet," in *Proceedings of the 2nd international conference on Embedded Networked Sensor Systems (SenSys '04)*. New York, NY, USA: ACM, 2004, pp. 227–238.

[81] S. Farrell and V. Cahill, *Delay- and Disruption-Tolerant Networking*. Norwood, MA, USA: Artech House, Inc., 2006.

[82] P. Bonnet and M. Chang, "Monitoring in a high-arctic environment: Some lessons from mana," *IEE Pervasive Computing*, vol. 9, no. 4, Oct. 2010, pp. 16–23.

[83] H. Ochiai and H. Esaki, "Mobility entropy and message routing in community-structured delay tolerant networks," in *Proceedings of the 4th Asian Conference on Internet Engineering (AINTEC '08)*. New York, NY, USA: ACM, 2008, pp. 93–102.

[84] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, and J. Porter, "LUSTER: Wireless sensor network for environmental research," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*. New York, NY, USA: ACM, 2007, pp. 103–116.

[85] E. Koutsogiannis, L. Mamatas, and I. Psaras, "Storage-enabled access points for improved mobile performance: an evaluation study," in *Proceedings of the 9th IFIP TC 6 international conference on Wired/wireless internet communications (WWIC'11)*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 116–127.

[86] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, vol. 6. IEEE, 2006, pp. 1–11.

[87] J. Ott and D. Kutscher, "Drive-thru Internet: IEEE 802.11b for "automobile" users," in *INFOCOM 2004. 23rd IEEE International Conference on Computer Communications. Proceedings*, vol. 1. IEEE, 2004, pp. 4 vol. (xxxv+2866).

[88] L. McNamara, C. Mascolo, and L. Capra, "Media sharing based on colocation prediction in urban transport," in *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2008, pp. 58–69.

[89] K. Zaragoza, N. Thai, and T. Christensen, "An implementation for accessing Twitter across challenged networks," in *Proceedings of the 6th ACM Workshop on Challenged Networks (CHANTS '11)*. New York, NY, USA: ACM, 2011, pp. 71–72.

[90] A. Balasubramanian, Y. Zhou, W. Croft, B. Levine, and A. Venkataramani, "Web search from a bus," in *Proceedings of the Second ACM Workshop on Challenged Networks (CHANTS '07)*. New York, NY, USA: ACM, 2007, pp. 59–66.

[91] S. Eisenman, E. Miluzzo, N. Lane, R. Peterson, G. Ahn, and A. Campbell, "Bikenet: A mobile sensing system for cyclist experience mapping," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 1, Jan. 2009, pp. 6:1–6:39.

[92] S. Goebbels, "Disruption tolerant networking by smart caching," *Int. J. Commun. Syst.*, vol. 23, no. 5, May 2010, pp. 569–595.

[93] H.-M. Lin, Y. Ge, A.-C. Pang, and J. Pathmasuntharam, "Performance study on delay tolerant networks in maritime communication environments," in *IEEE OCEANS 2010*, May 2010, pp. 1–6.

[94] "The centre quartely," Mar. 2011. [Online]. Available: http://www.nurc.nato.int/news/CDozen/201103_e-quarterly.pdf

[95] T. Jonson, J. Pezeshki, V. Chao, K. Smith, and J. Fazio, "Application of delay tolerant networking (dtn) in airborne networks," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, Nov. 2008, pp. 1–7.

[96] C. Rigano, K. Scott, J. Bush, R. Edell, S. Parikh, R. Wade, and B. Adamson, "Mitigating naval network instabilities with disruption tolerant networking," in *Proceedings of the 2008 IEEE Military Communications Conference (MILCOM 2008)*, Nov. 2008, pp. 1–7.

[97] S. Parikh and R. Durst, "Disruption tolerant networking for marine corps CONDOR," in *Military Communications Conference, 2005. MILCOM 2005. IEEE*, vol. 1, Oct. 2005, pp. 325–330.

[98] M. Paula, J. Isento, J. Dias, and J. Rodrigues, "A real-world VDTN testbed for advanced vehicular services and applications," in *16th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD 2011)*. IEEE, 2011, pp. 16–20.

[99] B. Silva, V. Soares, M. Proena, and J. Rodrigues, "Impact of content storage and retrieval mechanisms on the performance of vehicular delay-tolerant networks," in *International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2010)*. Split, Croatia: FESB, University of Split, 2010, pp. 407 –411.

[100] V. N. G. J. Soares, J. J. P. C. Rodrigues, F. Farahmand, and M. Denko, "Exploiting node localization for performance improvement of vehicular delay-tolerant networks," in *2010 IEEE International Conference on Communications (ICC 2010)*. IEEE, 2010, pp. 1–5.

[101] H. Ochiai, K. Shimotada, and H. Esaki, "IP over DTN: Large-delay asynchronous packet delivery in the internet," in *Ultra Modern*

Telecommunications Workshops, 2009. ICUMT '09. International Conference on, Oct. 2009, pp. 1–4.

[102] H. Ochiai, M. Nakayama, and H. Esaki, "Hop-by-hop reliable, parallel message propagation for intermittently-connected mesh networks," in *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2011, pp. 1–9.

[103] C. Caini, P. Cornice, R. Firrincieli, and D. Lacamera, "A DTN approach to satellite communications," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 5, June 2008, pp. 820–827.

[104] C. Caini, R. Firrincieli, and M. Livini, "DTN bundle layer over TCP: Retransmission algorithms in the presence of channel disruptions," *Journal of Communications*, vol. 5, no. 2, 2010, pp. 106–116.

[105] M. Zarafshan-Araki and K. Chin, "Trainnet: A transport system for delivering non real-time data," *Computer Communications*, vol. 33, no. 15, 2010, pp. 1850–1863.

[106] A. Keränen and J. Ott, "Dtn over aerial carriers," in *Proceedings of the 4th ACM Workshop on Challenged Networks (CHANTS '09)*. New York, NY, USA: ACM, 2009, pp. 67–76.

[107] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with NetStitcher," in *Proceedings of the ACM SIGCOMM 2011 conference on SIGCOMM*, ser. SIGCOMM '11. New York, NY, USA: ACM, 2011, pp. 74–85.

[108] K. Fall, G. Iannaccone, J. Kannan, F. Silveira, and N. Taft, "A disruption-tolerant architecture for secure and efficient disaster response communications," in *Proceedings of the 7th International Conference on Information Systems for Crisis Response and Management*, 2010.

Artemios G. Voyiatzis received a B.Sc. in Mathematics from the University of Crete, Greece; a B.Sc. and an M.Sc. in Computer Science from the University of Crete, Greece; and a Ph.D. in Electrical and Computer Engineering from the University of Patras, Greece (January 2009).

Artemios joined in January 2011 the Industrial Systems Institute of the R.C. "Athena", Greece as an Associate Researcher. His research interests include networked embedded systems (cyberphysical) security, network security, system architecture, usable security and privacy, and cryptography applications and performance.