Transmission Protocols for Delay-Tolerant Networks

Adrian Rumpold
Betreuer: Dr. Nils Kammenhuber
Seminar Innovative Internettechnologien und Mobilkommunikation SS2011
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: rumpold@in.tum.de

ABSTRACT

In this paper, we provide an overview of important transmission protocols for a certain class of challenged heterogeneous networks commonly termed *delay-* or *disruption-tolerant networks* (DTN). We outline basic requirements and limitations for these protocols, including the Bundle Protocol and the LTP and Saratoga convergence layer protocols, sketch their respective solution approaches and provide and overall comparison. Finally, we discuss some inherent shortcomings and operational challenges of the Bundle Protocol specification regarding areas of reliability, routing and time synchronization as well as schematic examples for future improvements.

Keywords

DTN, delay-tolerant networks, disruption-tolerant networks, network protocols, Bundle Protocol, convergence layer protocols, LTP, LTP-T, Saratoga

1. INTRODUCTION

Wide-area communication today is far from being limited to the Internet only, but rather includes challenging environments such as near- and deep-space satellite links, underwater equipment and various forms of wireless sensor networks. These usage scenarios differ significantly in their characteristics from the well-known Internet architecture in terms of link availability, signal quality and communication channel bandwidth. As network connectivity might only be available sporadically, the foremost goal is maximal utilization of transmission opportunities, even if this might imply decreased reliability. A framework for such scenarios is given by delay- or disruption-tolerant networks (DTN), omitting a traditional end-to-end conversation paradigm in favor of a store-and-forward architecture.

Work on DTN technologies commenced in the late 1990s following the vision of an *Interplanetary Internet* (IPN), where communication delays of multiple hours and frequent link outages had to be considered and expected [4, p. 3]. Given their aptitude for highly dynamic and resilient communication architectures with ad-hoc connectivity, disruption-tolerant network protocols also elicited interest in military circles as an enabler for wireless integrated battlefield networks [12, p. 2].

In their reference specification, Cerf et al. [4] (first drafted in 2003) propose an end-to-end message-oriented overlay protocol, the *Bundle Protocol*; this architecture was subsequently further refined by Scott and Burleigh [8]. We briefly describe the Bundle layer protocol's structure and important aspects in section 2. This protocol can either be used directly on top of a regular TCP/IP network connection or use specialized *convergence layer protocols*, bridging the gap between the abstract bundle overlay structure and the underlying network connection.

In section 3 we introduce two important convergence layer protocols, the *Licklider Transport Protocol* (LTP) and the *Saratoga* protocol. Section 4 provides a discussion of some major challenges and obstacles frequently encountered in DTN applications. In section 5 we present an outlook onto possible future developments in DTN research as well as a technology for delay-tolerant communication without the Bundle Protocol.

Figure 1 demonstrates how the protocols discussed can be integrated into the stack of preexistent Internet protocols for use in DTN applications. The Bundle Protocol exposes the abstracted interface to the application layer, the Bundle layer in turn accesses underlying convergence layer protocols. Besides the specialized convergence layer protocols presented in this paper, the TCP/IP protocol can also be used for reduced configuration complexity in testing environments [4, p. 29].

An in-depth discussion of standard protocols like *TCP*, *UDP* and the Internet Protocol *IP* as well as various data link layer protocols is beyond the scope of this work; interested readers can find further reading regarding their properties for example in Tanenbaum [11].

2. BUNDLE PROTOCOL

In the following subsections, we describe the basic concepts and properties behind the Bundle Protocol first proposed by Internet pioneer Vint Cerf in 2003 as a communication protocol for the envisioned Interplanetary Internet. Shortly after the initial drafts, the IRTF Delay-Tolerant Networking Working Group published two RFCs 4838 and 5050 [4, 8], defining the general architecture for DTN environments as well as the Bundle Protocol itself.

2.1 Protocol overview

The Bundle Protocol constitutes a message-oriented overlay atop various transport protocols. It may be used over TCP/IP connections as well as convergence layer protocols, such as the Licklider Transmission Protocol (LTP) and the

doi: 10.2313/NET-2011-07-2_18

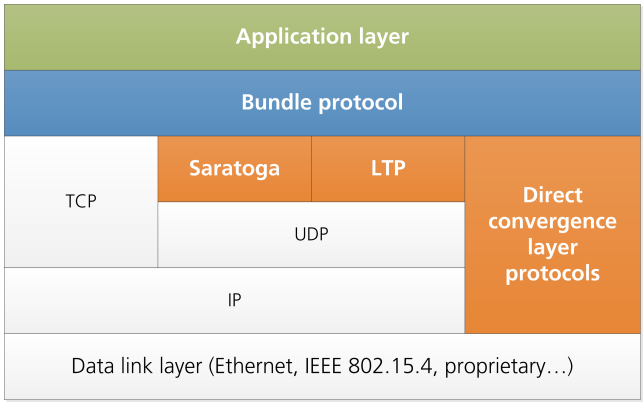

Figure 1: Classification of DTN protocols in the Internet protocol hierarchy

Saratoga protocol discussed later in section 3. The following brief description of the protocol is based on the DTN architecture specification [4] and the Bundle Protocol specification [8] where not stated otherwise.

Applications using the Bundle Protocol may send messages of arbitrary length. These application data units (ADU) are encapsulated into protocol data units (PDU) when passed to the bundle layer; a single ADU is usually transmitted completely by a PDU, but may also be segmented by the bundle layer. PDUs in the Bundle Protocol are referred to as *bundles* and are transmitted between nodes in a store-and-forward fashion. This communication paradigm distinguishes delay-tolerant networks from traditional IP-based local- and wide-area networks, where packets are usually stored in transmission buffers only for short periods of time during inter-node routing. Rather, en-route bundles may be stored for indefinite amounts of time by their current hop until a suitable connection for forwarding towards its final destination is available. Such connections may present themselves for example in the form of a scheduled communication contact with a space vessel or opportunistic contacts in wireless sensor networks.

Bundle sources and destinations are identified by means of *Endpoint identifiers* (EID). Multiple naming schemes for EIDs are proposed by the Bundle Protocol specification, each conforming to a common Uniform Resource Identifier (URI) format and consist of a scheme name[1] and a scheme-specific part (SSP). Therefore, the URI `ipn:rover.mars` is one of the numerous possible examples of such an endpoint identifier with `ipn` as the scheme name and `rover.mars` as the SSP.

Associations between endpoint identifiers and node addresses are not established at creation time of a bundle. Instead rather a late binding occurs, so a specific EID might be reinterpreted several times during bundle delivery. This late binding facilitates bundle delivery in cases of changes in the network topology of which the originating node is not yet aware or if the transit duration of a message exceed validity times of EID registration bindings [4, p. 9].

A simplistic quality-of-service approach is included in the Bundle Protocol by class of service specifiers in a bundle's header information. A message may take a relative priority indicator of *bulk*, *normal* or *expedited*, however the specification does not prescribe any detailed handling policy for forwarding nodes [8, pp. 13 f.].

To increase transmission efficiency during phases of intermittent connectivity, the DTN architecture includes considerations for proactive as well as reactive fragmentation. Proactive fragmentation precludes transmission of messages larger than a scheduled contact between two nodes by breaking up larger bundles into smaller fragments suitable for the scheduled connection and its total data volume. Reactive fragmentation on the other hand is employed as a reaction to interrupted connections, where only partial content has been transmitted successfully. The forwarding node is permitted to break the incomplete message into two fragments, so the content already received correctly may instantaneously be transmitted further without the need for retransmitting the complete bundle [3, p. 18].

Figure 2 illustrates how the Bundle Protocol can be integrated with heterogeneous network architectures comprising different technology stacks. Heterogeneity with respect to communication protocols can occur on any of the protocol layers underneath the Bundle Protocol. Such configurations may be found in operations where DTN traffic is subsequently forwarded across Internet TCP/IP connections to its final destination. Bundle protocol routers along the communication path are capable of bridging between different network layer and convergence layer protocols (see section 3).

2.2 Reliable transmission support in the Bundle Protocol

TCP provides end-to-end conversational reliability by means of retransmission of segments not acknowledged by the destination host [11, pp. 532 ff.]. This technique cannot be directly applied to delay-tolerant network architectures, however, where a continuous end-to-end connectivity is oftentimes not available. Therefore, the Bundle Protocol supports node-to-node retransmission of incorrectly transferred data. As a single DTN may incorporate a number of incompatible transport layer protocols, end-to-end reliability in such inhomogeneous environments can only be achieved at the bundle layer [12, p. 17].

Node-to-node reliability in the bundle layer is provided by *custody transfers* between two neighboring nodes. Intending such a hand-off, a bundle's current custodian transmits the bundle to the next node and starts a retransmission timer. If the receiving peer does not complete the custody transfer by acknowledging correct reception, the transmission is repeated until successful acknowledgment by the designated custodian. After completing a custody transfer, the previous custodian may safely delete the bundle from its long-term storage, as the current custodian is now responsible for reliable forwarding of the bundle to its destination or subsequent custodians.

[1] IANA assigned URI schemes are published at `http://www.iana.org/assignments/uri-schemes.html`

doi: 10.2313/NET-2011-07-2_18

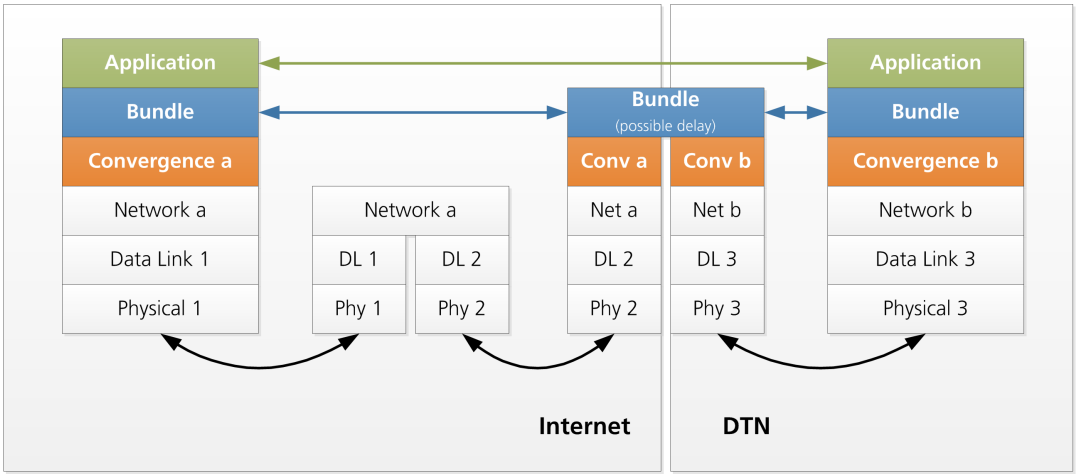

Figure 2: End-to-end bundle routing across heterogeneous protocol stacks and network architectures (based on [3])

To achieve end-to-end reliability, transmission of *return receipts* is required besides enforcement of node-to-node custody transfers. This additional confirmation is required, since a bundle may have been discarded by an en-route custodian after expiry of its time-to-live timestamp before arriving at its designated destination node.

3. CONVERGENCE LAYER PROTOCOLS

While the Bundle Protocol in theory can be operated over arbitrary transport protocols, oftentimes the need for specialized protocols arises, which address the various dissimilarities in heterogeneous network environments found in DTN setups. These class of protocols is subsumed under the term *convergence layer protocols*. Their main task is to provide an abstraction over underlying network protocol layers and permit best possible link utilization. Convergence layer protocols may themselves either operated on top of transmission protocols such as UDP or directly access the data link layer if desired [15, p. 4].

3.1 Licklider Transmission Protocol (LTP) & LTP-T

The Licklider Transmission Protocol (LTP), named after Internet pioneer J.C.R. Licklider, provides selective-reliability communication over high-latency links. It is purposed as a block-oriented convergence-layer protocol for use in interplanetary communication segments in delay-tolerant multihop networks [6, pp. 1 f.].

LTP operations between two communication nodes consist of two distinct parts: a reliably transmitted red part and following green part segments, for which no reliability is assured by the protocol. On start of transmission a red part segment and one or more checkpoints (including a special red part segment – end-of-red-part [`EORP`]) are transferred to the receiver, which in turn sends out report segments containing information about successfully received segments. Reception of these reports is subsequently acknowledged by the original sender. If a timeout occurs while waiting for an acknowledgment, the respective segment is retransmitted until its reception has been successfully confirmed [7, section 6].

Following the `EORP` message one or more green segments are transmitted without acknowledgment of successful reception to maximize communications throughput during limited phases of episodic connectivity. After conclusion of a conversation by means of a end-of-block (`EOB`) message, all transmitted data is passed to the application layer. [6, p. 1].

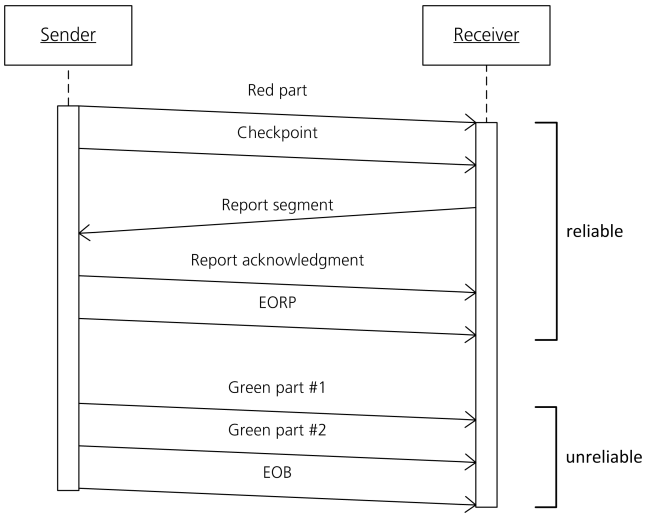

Figure 3: LTP transmission sequence

Figure 3 shows a complete LTP communication session between two node consisting of one red part and two green part segments without any transmission errors. Only the red part segment is transmitted in a reliable fashion, whereas reception of the green part segments is not acknowledged by the remote peer and the communication is terminated by the sender after the `EOB` segment. The depicted communication session can be envisioned during transmission of satellite imaging data: Here, meta-information about transmitted records have to be transmitted reliably in order to successfully process the following image data. Such metadata is therefore a candidate for inclusion into the red part segment at the beginning of the conversation. Errors with regard

doi: 10.2313/NET-2011-07-2_18

to the image data itself, however, can be tolerated. Thus, such information is suitable for bulk transmission within the subsequent green part segments.

As the LTP protocol was primarily developed for single-hop connections only, it was subsequently extended under the name of LTP-T to form a transport protocol and accommodate multi-hop architectures. During normal operation, LTP-T generally performs as if a single LTP connection was used between every two nodes. Major differences when compared to the LTP protocol can be found in case of transmission errors: Correctly received segments are forwarded to the next hop, while only corrupted or lost segments are requested to be retransmitted from the original sender. Because of this mechanism, the ordering of transmitted segments is possibly changed, so re-ordering of the segment sequence and checkpoint scheduling become important issues and are subjects of ongoing research work [6].

3.2 Saratoga Protocol

The Saratoga protocol, named after United States World War II aircraft carrier USS Saratoga, was originally designed as a IP-based store-and-forward file transmission protocol for small satellites in a near-Earth low orbit [9, p. 4]. As opposed to regular Internet communication, connectivity in such scenarios is only intermittent in nature and highly asymmetrical in terms of up- and downstream bandwidth, a configuration commonly found in delay-tolerant networks. The UK-DMC imaging satellite, for example, features a downstream bandwidth of 8.1 Mbps opposed to an upstream of only 9.6 kbps [ibid.] resulting in a bandwidth asymmetry of around 844 : 1 – compared to commercial ADSL2 lines featuring a factor of merely 8 : 1. The TCP protocol performs increasingly worse for links exhibiting asymmetries greater than 50 : 1, as the return path to the sender is congested by acknowledgment segments [1].

If the TCP protocol were to be used for communication, every segment sent over the network link would have to be acknowledged by the receiver, leading to reduced bandwidth utilization and impact on overall transfer performance. Moreover, the TCP slow start mechanism further hinders effective link utilization [13, pp. 3 f.]. Additional challenges are constituted by high round-trip-times (RTT), bit error rates (BER) and frame loss rates (FLR) on interplanetary communication links.

Therefore, the lightweight Saratoga protocol does not rely on TCP as a transport protocol, but instead uses the connectionless UDP and UDP Lite protocols, which do not require acknowledgment of transmitted datagrams and therefore do not provide a reliable service [11, pp. 525 f.]. By omitting the premise of fair line contention and focusing on communication scenarios with only two participants, Saratoga strives to achieve complete utilization of the communications link and maximization of data throughput [13, p. 1].

As a file transmission protocol, Saratoga provides support for transactions operating on files and directories as their underlying basic entities. Accordingly, the protocol implements operations similar in nature to the well-known FTP protocol, including instructions for fetching or deleting files (`_get_`, `_delete_`), transferring files to a remote peer (`_put_`)

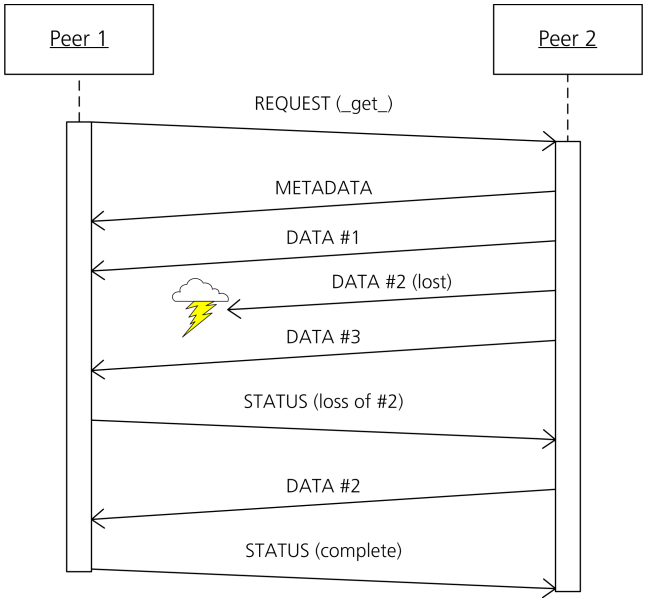

Figure 4: Retransmission during _get_ request

and querying remote directory listings (`_getdir_`) (nomenclature as described in [13, pp. 6 f.]).

Given that the underlying UDP protocol does not provide an indication of correctly transmitted datagrams, the Saratoga specification introduces a selective negative-acknowledgment (SNACK) status message. Communicating peers may exchange `STATUS` packages indicating ranges of successful and failed `DATA` packages during the active conversation, permitting for a selective retransmission of lost segments to be initiated. Figure 4 illustrates this technique with a exemplary `_get_` request consisting of three distinct `DATA` packages, one of which is lost during transmission. This condition is indicated in the client's `STATUS` packet, resulting in the subsequent repetition of `DATA` package #2.

To increase data throughput in deployments with high round-trip latencies, speculative `_get_` requests with an empty file path are permitted. As an answer to such a request the remote Saratoga peer may send any file it deems relevant for the client and immediately commence its transmission. Analogous procedures are also specified for optimistic `_put_` operations, where the sending peer will begin its data transfer without waiting for a corresponding `STATUS` response by the receiving peer [13, p. 9].

In addition to being used in standalone applications, Wood et al. [15] demonstrate how the Saratoga protocol can be used as a convergence layer with the Bundle Protocol. The bundle agent can cooperate with the Saratoga peer by using a shared directory, wherein the DTN bundle agent stores completed bundles as files. Those bundle files can subsequently be requested via the `_get_` transaction or pushed to another Saratoga peer using the `_put_` operation.

4. CHALLENGES AND OBSTACLES

As a fairly recent area of research, many issues in the field of delay-tolerant networking have yet to be solved. In the

doi: 10.2313/NET-2011-07-2_18

following section we identify and present three distinct challenges to the Bundle Protocol; in particular regarding insurance of bundle integrity, routing and time synchronization across intermittently connected networks. These issues and several others were discussed by Wood et al. [14].

4.1 Bundle integrity

Lacking support for checksums, the Bundle Protocol is unable to detect errors in transmitted application data or bundle header metadata. Custody transfers on the other hand can only provide protection against loss of complete bundles and facilitate fast retransmission in such cases. Although error detection and correction could be implemented at the application level, this technique still does not guard against corruption of header information which is not propagated up to the application. Furthermore such an application-provided integrity mechanism introduces a tighter coupling between sending and receiving applications and is therefore not a viable remedy. Resulting from these shortcomings, the Bundle Protocol in its basic form as specified in [8] does not sufficiently support reliable end-to-end transmissions with integrity guarantees.

Instead of introducing header checksums or similar integrity checking measures, Wood et al. [14] propose a different approach making use of the optional security extensions of the Bundle Protocol [10]. Usually the problem of efficient and secure key distribution arises when using either symmetrical or asymmetrical cryptographic systems. However, if only integrity of transport is to be assured, use of a common ciphersuite and a well-known key shared between all participating communication nodes is sufficient. This cryptographic system permits wrapping immutable header information as well as application data in order to defend against incidental errors introduced during transmission, reassembly of fragments or in-memory storage of bundles.

This approach may even be used in conjunction with end-to-end encryption to enhance performance by permitting intermediary routers to check for possible modification of the bundle's content, although they are unable to decrypt the encapsulated ADU. In contrast to a mere end-to-end encryption, corrupted bundles may be retransmitted faster, thus leading to increased overall network performance.

4.2 Name Resolution and Routing

The issue of routing in delay-tolerant networks is closely connected to the set of problems revolving around common naming schemes for endpoint identifiers in the Bundle Protocol. Since no name-resolution protocols such as the Domain Name System (DNS) have been defined for DTNs yet, resolving a bundle's destination EID and deriving routing information is inherently difficult. Late binding of EIDs onto network addresses further complicates routing in scenarios with multi-homed or mobile peers roaming between several subnetworks.

This problem could be tackled by providing static routing information to each participating node in the network, however this approach poses a stark contrast to the ad-hoc networking architecture found in DTN applications. Another possibility is the usage of source routing, possibly deriving routing information and forwarding rules from the EID itself by postulating a distinct hierarchy within the naming scheme.

Traditional Internet routing protocols, for example the Routing Information Protocol (RIP) and the Border Gateway Protocol (BGP), are not suitable for long-delay networks with intermittent connectivity, as they require excessive exchange of routing information metadata to accommodate for changing network topologies. Candidates for routing in DTN bundle architectures have yet to be fully specified and should be independent from underlying convergence layer protocols to facilitate integration and interconnection of various heterogeneous networks.

The use of replication-based routing protocols instead of forwarding strategies with a singular transmission path is a recent but promising area of development. These protocols enable a DTN router to propagate multiple instances of a single Bundle to its neighboring peers for further transmission. While being more resource-intensive, these approaches help to address and alleviate the problematic issues raised above. Examples for replication-based routing protocols for delay-tolerant networks include the simplistic epidemic routing and its more complex enhancement, the *PRoPHET* protocol (Probabilistic Routing Protocol using History of Encounters and Transitivity) [5].

4.3 Time synchronization

Another pressing issue in DTN architectures presents itself in the form of time synchronization. Timestamps are employed in several places in the Bundle Protocol, including the lifetime and creation header fields. The lifetime works analogously to the time-to-live (TTL) header field of the IP protocol and helps to prevent messages in the network from looping indefinitely in case of routing malfunctions. Moreover, this approach facilitates discarding of packets which carry data of limited usefulness after certain expiration times. Messages may be uniquely identified by their creation timestamp, which is used for example during fragment reassembly – its semantics are similar to the identification field of the IPv4 header.

For the aforementioned reasons a common notion of time as well as mechanisms for time synchronization are required across a delay-tolerant network using the Bundle Protocol overlay. Time synchronization in those scenarios cannot be achieved by the Bundle Protocol itself, as correct time information must be acquired first for production of valid bundles. Wood et al. [14] propose an additional simple time exchange protocol, where network nodes claim different confidence levels for themselves regarding internal clock accuracy. Peers with low confidence levels can subsequently improve their clock precision by acquiring current time from nodes with higher confidence levels.

In addition to operational difficulties, security considerations must also be taken into account when designing strategies and protocols for network time synchronization. Only authorized and authenticated nodes should be able to distribute time information to other clients in the network and adequate measures must be taken to assure their identity. Failure to comply with this requirement enables a malicious entity to isolate peers from the network using a denial-of-

doi: 10.2313/NET-2011-07-2_18

service attack, effectively preventing them from communicating with other Bundle Protocol hosts.

5. OUTLOOK AND CONCLUSION

As we discussed in section 4, the young family of protocols for delay-tolerant networks contains room for enhancements and ongoing research, especially considering security, communication integrity and routing issues. Most of these topics are still work in progress, therefore no final statement can be given regarding their future development. It seems plausible, however, that with continuously increasing application scenarios involving the need for delay-tolerant networking, the protocols involved will rapidly reach a more mature status by successive improvements.

DTN applications are not limited to the Bundle Protocol however: Wood et al. [14] propose an alternative approach using an extension of the HTTP protocol (*HTTP-DTN*) directly atop of a convergence layer protocol. Instead of passing Bundle PDUs, application data is encapsulated into HTTP requests, newly defined `Content-*` headers carry accompanying delivery metadata. `Content-MD5` headers are capable of providing end-to-end reliability by inclusion of a payload checksum.

HTTP clients unable to understand the additionally defined headers are required to disregard the respective request, so the proposed HTTP-DTN architecture should not interfere with ordinary Internet HTTP traffic. Albeit sharing some of the shortcomings of the Bundle Protocol, its integrated support for integrity checks as well as higher familiarity of developers with HTTP implementations might make the HTTP-DTN protocol attractive for certain application cases.

References

[1] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz. The effects of asymmetry on TCP performance. *Mobile Networks and Applications*, 4:219–241, 1999. ISSN 1383-469X. URL `http://dx.doi.org/10.1023/A:1019155000496`.

[2] S. Burleigh, M. Ramadas, and S. Farrell. RFC 5325, Licklider Transmission Protocol Motivation. IRTF Delay-Tolerant Networking Research Group, September 2008. URL `http://tools.ietf.org/pdf/rfc5325.pdf`.

[3] V. Cerf. InterPlaNetary Internet – presentation at DARPA Proposers' Day, 21 January 2004.

[4] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, and R. Durst. RFC 4838, Delay-Tolerant Networking Architecture. IRTF Delay-Tolerant Networking Research Group, April 2007. URL `http://tools.ietf.org/pdf/rfc4838.pdf`.

[5] A. Lindgren, A. Doria, E. Davies, and S. Grasic. Probabilistic Routing Protocol for Intermittently Connected Networks. IRTF Delay-Tolerant Networking Research Group, 2011. URL `http://tools.ietf.org/pdf/draft-irtf-dtnrg-prophet-09.pdf`.

[6] F. S. Muhammad, L. Franck, and S. Farrell. Transmission protocols for challenging networks: LTP and LTP-T, Sept. 2007. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4409406`.

[7] M. Ramadas, S. Burleigh, and S. Farrell. RFC 5326, Licklider Transmission Protocol Specification. IRTF Delay-Tolerant Networking Research Group, September 2008. URL `http://tools.ietf.org/pdf/rfc5326.pdf`.

[8] K. Scott and S. Burleigh. RFC 5050, Bundle Protocol Specifications. IRTF Delay-Tolerant Networking Research Group, November 2007. URL `http://tools.ietf.org/pdf/rfc5050.pdf`.

[9] C. Smith, C. Jackson, W. Eddy, L. Wood, and W. Ivancic. Saratoga: A Scalable File Transfer Protocol. Transport Area Working Group, 2010. URL `http://tools.ietf.org/html/draft-wood-tsvwg-saratoga-08`.

[10] S. Symington, S. Farrell, H. Weiss, and P. Lovell. RFC 6257, Bundle Security Protocol Specification. IRTF Delay-Tolerant Networking Research Group, May 2011. URL `http://tools.ietf.org/pdf/rfc6257.pdf`.

[11] A. S. Tanenbaum. *Computer Networks*. Prentice Hall PTR, 4th edition, 2003. ISBN 9780130661029.

[12] F. Warthman. Delay-tolerant networks (DTNs): A tutorial, 2003. URL `http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf`.

[13] L. Wood, W. M. Eddy, W. Ivancic, J. McKim, and C. Jackson. Saratoga: a Delay-Tolerant Networking convergence layer with efficient link utilization. In *2007 International Workshop on Satellite and Space Communications*, pages 168–172. IEEE, Sept. 2007. ISBN 978-1-4244-0938-9. doi: 10.1109/IWSSC.2007.4409410. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4409410`.

[14] L. Wood, W. Eddy, and P. Holliday. A bundle of problems. *IEEE Aerospace conference*, pages 1–17, 2009. doi: 10.1109/AERO.2009.4839384. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4839384`.

[15] L. Wood, J. McKim, W. M. Eddy, and W. Ivancic. Using Saratoga with a bundle agent as a convergence layer for delay-tolerant networking. IRTF Delay-Tolerant Networking Research Group, May 2011.