EURECOM
Department of Mobile Communications
Campus Sophia Tech
Les Templiers
450 route des Chappes
B.P. 193
06410 Biot
FRANCE

Research Report RR-13-285

Transport Layer Protocols and Strategies for Delay and Disruption Tolerant Networks

July 30$^{\text{th}}$, 2013
Last update July 30$^{\text{th}}$, 2013

Panagiotis Matzakos, and Christian Bonnet

Tel : (+33) 4 93 00 81 00
Fax : (+33) 4 93 00 82 00
Email : {Panagiotis.Matzakos,Christian Bonnet}@eurecom.fr

[1]EURECOM's research is partially supported by its industrial members: BMW Group, Cisco, Monaco Telecom, Orange, SAP, SFR, STEricsson, Swisscom, Symantec.

Panagiotis Matzakos, and Christian Bonnet

Abstract

Delay Tolerant Network Architecture was initially designed to deal with communication challenges in Interplanetary Deep Space Networks. Such challenges can be for example the lack of end-to-end connectivity, communication through error-prone asymmetric channels, as well as long propagation delays and connectivity disruptions. In this context, the application of traditional TCP/IP stack can be very problematic and DTN architecture aims to overcome the communication obstacles by offering a transparent and flexible scheme, capable of providing data transfer services which can "survive" the challenging conditions and at the same time they are applicable throughout heterogeneous Network Environments. DTN architecture and its transport layer techniques have been more recently considered suitable for terrestrial applications as well, in environments which can have similar properties with Deep Space. In this survey we aim to show the similarities as well as the particularities of each application environment and present analytically the solutions offered in the literature, trying to highlight their possible applicability in environments different from the ones for which they were initially proposed.

Index Terms

Delay/Disruption Tolerant Networks, Transport Layer

Contents

List of Figures

1 Introduction

2 Common Characteristics and Requirements

Among the application domains that we focus on, we can find a lot of common basic guidelines regarding their characteristics.

2.1 Common Characteristics

- **Lack of End-to-End Connections:** In the Internet world, end-to-end connection capability is taken for granted, which is the reason of TCP's dominance with respect to a wide range of applications in this context. In the environments that we are going to deal with in this survey, however, the lack of end-to-end connections is rather the rule than the exception, constituting their most crucial difference from Internet domain environments and dictating the need to come up with alternative transport-layer mechanisms and architectures in order to be able to provide the required services (e.g. Reliability, minimum possible latency, addressing etc.) which are still needed, regardless of the environment's particularities.

- **Frequent channel errors:** Both Interplanetary and terrestrial networks operating in distant, link error-prone environments suffer from high packet-loss rates. These events are treated falsely as congestion events by TCP, triggering mechanisms which limit the transmission rate and as a result they lead to non-optimal usage of the available bandwidth.

- **Limited transmission opportunities:** Interplanetary communications are characterized by limited, often periodical, Line of Sight (LOS) transmission opportunities, which means that when these opportunities appear, it is crucial that the bandwidth is exploited in the best way. In the case of terrestrial disruption tolerant networks the same effect can be observed in disconnected areas where the communicating nodes should take maximum advantage of opportunistic contacts (e.g. Message ferries). A different terrestrial environment would be a Vehicular Network where, due to the limited coverage of the WLAN infrastructure as well as the high speed of the vehicles, the transmission opportunities are once again limited.

- **High Asymmetric Links:** Another attribute of the studied environments is the asymmetric nature of the links (uplink and downlink directions) which can lead to buffer congestion, abortion of message transfers due to lack of updated routing information, additional retransmissions and slow increase of the congestion window, in case we use TCP - Internet based solutions, which do not assume these conditions.

- **Low data rates - High delivery delay:** It is obvious that a consequence of all the above are much lower data rates and higher delivery delay than the ones of the Internet world. Beyond that, as explained in [4], for a constrained system (energy and/or storage), we could improve the performance with respect to data rates and delivery delay but on the price of lower reliability (delivery ratio). In other words, for a constrained system, there will always be a trade-off between those two attributes and the application-specific requirements should be the ones which will show the way towards the best suited policy.

- **Heterogeneous Network Environments:** We can say that both space and terrestrial DTN solutions have to be functioning within Heterogeneous Networks in order to provide end-to-end message delivery. More specifically, there is the common need for some node to act as the gateway between the structure-less intermittently connected Network component and the Internet-based structured Network component. For this operation to be accomplished, some appropriate architecture has to be applied, which will ensure the interoperability between the two Network components.

- **High RTT:** Due to the large amount of disconnection periods and the large propagation delays (for the case of interplanetary communications), a common characteristic of both space and terrestrial DTNs are the much higher RTTs comparing to the Internet-based terrestrial scenarios. This clearly has an important impact on the design of transport layer protocols such as appropriate adjustments in retransmission strategies and timeout durations, avoidance of slow increase of the transmission window and exploitation of the channel bandwidth when the transmission opportunity appears.

2.2 Requirements

- **Reliability:** Although, as we said before, it is often impossible to maintain end-to-end connection for Delay and Disruption Tolerant Networks in most cases, we still need to assure somehow the reliability of message reception. As we will show later, the way that the proposed DTN architecture [15] deals with this is by assuring reliability in a hop-by-hop basis on the way to the final destination.

- **Congestion and Rate control:** These two transport layer services are necessary in order to secure the fine operation of the overall network and the fine operation of the receiver in terms of avoiding buffer overflows respectively. In the error-prone, frequently disconnected and/or large propagation delayed environments that we investigate, providing these two services requires dealing with a wide range of issues like for example the difficulty of distinguishing losses due to congestion from losses due to channel error or large RTTs, as we described above. Additionally, for a store-and-forward DTN architecture as the one that we will describe in the next section [15], storing (large) amounts of data at gateways or intermediate nodes can easily lead to buffer overflows in the cases of low outgoing data rate links or connectivity loss respectively (when more and more data is gathered at a node who should wait until connectivity is restored before he can forward them).

- **In order delivery:** Due to the specific nature of the environments we are investigating, unordered data delivery is another expected event to happen. Re-transmitting large amounts of data flows due to this is a luxury that we cannot really afford for DTNs in contrast with Internet-based environments once again. As a result, we need to come up with ordering mechanisms within the transport and/or application layers, which should either happen at the intermediate hops of the path or the destination node itself.

3 Requirements and Characteristics per Environment

Apart from the common characteristics and requirements described above, we will describe in this section the additional environment-specific attributes and requirements.

3.1 Interplanetary Networks

The basic difference between space and terrestrial DTNs is that, in space, the contacts (i.e. Time intervals when LOS conditions appear between the two connection ends) are mostly **scheduled and predetermined** and as a result routing can be static and we could even exploit end-to-end paths in some cases between the source and the destination. Based on that, the authors of [3] suggest that space oriented DTNs could benefit from sophisticated transport protocols, running end-to-end, in order to exploit alternative communication paths in cases of nodes' failure or links' disruptions*. This strategy would be beneficial in cases where timely delivery of data is important and we cannot afford waiting for a bundle to be delivered, when the node will recover from a failure or when the next transmission opportunity will appear.

Another characteristic of the Interplanetary Networks is that, due to the limited contact periods, we usually use **bulk data transfer** when transmission opportunities exist. In [5], Bezirgiannidis et al present an evaluation of the trade-offs regarding the packet size decisions in Interplanetary Networks. Finally, given the preciousness of space links, the protocols used should keep provision and security of the application data.

3.2 Terrestrial Networks

3.2.1 Disconnected Areas and Wireless Sensor Networks

We classify these two environments together because we consider that they have major similarities and analogies in terms of attributes and requirements. In [6], A.Seth et Al. describe the setting of Rural Internet Kiosks in disconnected areas, which aim to achieve inter-connectivity by the combination of multiple communication-capable relays (e.g. Message ferries, proxy gateways etc.). In [7,8,9,10] we can find some DTN-based deployment examples of WSNs, which are based on similar inter-connectivity strategies.

There is quite a large range of solutions provided for Transport Layer Protocols in Wireless Sensor Networks [11,12,13] and, although most of them are not in the context of DTN architecture, we believe that studying the way these protocols address particular aspects of data transfer in WSNs could be beneficial for dealing with the respective aspects within a DTN-based architecture for both WSNs and Disconnected Areas. Let us now highlight the most special properties of these environments.

- **Lack of Infrastructure:** In both environments it should be feasible to achieve communication with the respective communication ends, despite the lack of some core Network Infrastructure nearby. As a result, we have to make sure that we take advantage of opportunistic contacts (e.g. Randomly appearing message ferries) and of course follow appropriate route discovery strategies.
- **Limited resources:** Energy consumption and storage can be important constraints for WSNs. Given that the Round-trip delays are usually large for those intermittently connected environments, these constraints should be taken into account when designing our routing and retransmission strategies (e.g. Large scale of message replications and unnecessary retransmissions are not really energy-efficient).

3.2.2 Vehicular Disruption Tolerant Networks

Vehicular Disruption Tolerant Networks represent a different type of terrestrial DTNs, in the following manners:

- **Infrastructure is there:** We can assume that, for Vehicular types of networks, the Infrastructure is there and can be used (e.g. Roadside Wifi hotspots and especially cellular coverage for data transfer). However, the aim is to maintain its usage in low levels, in order to avoid creating huge overhead for the dissemination of information which is probably not as critical and urgent as safety warnings for example. As a result, a good strategy would be to use the infrastructure for disseminating control information (e.g. ACKs, feedback regarding the congestion of neighbor Roadside Units, signaling for data prefetch [16,17] etc.), which could help in the adequacy and accuracy of the decisions regarding the right time and place of (bulk) data transfers.
- **High Mobility:** It's easy to understand that, for a vehicular network within a highway scenario, the level of user's mobility is usually supposed to be much higher than the mobility within a Wireless sensor network or a disconnected area. This obviously has an impact on the contact duration with the

Infrastructure nodes. Indeed in [14] Simon Schutz et al consider connection periods with the Roadside Units (RSUs) in the scale of some seconds for their simulations within a vehicular environment, while in [8] opportunistic communication sessions duration within disconnected areas is expected to be in the range of 20 seconds to 5 minutes. As a result, we can assume that avoiding the delay of connection establishment is of even more critical importance for vehicular DTNs.

4 Transport layer Mechanisms

As we will show in the following sections, there are a lot of transport layer protocols based on different types of architectures which have been implemented to deal with the specific characteristics and requirements that we described above. We can classify the offered solutions in the following categories:

- TCP Extensions
- DTN - based
- Proxy - assisted
- Network Infrastructure - assisted.
- Combinations of the above.

In the following, we will present analytically the principles of the proposed architectures as well as representative protocols of each category, by highlighting which of the environment-specific requirements they can capture. In this analysis we will also try to point out the cases where protocols designed for particular types of Networks (e.g. Interplanetary communication) can conditionally also be applicable for other environments (e.g. Vehicular).

4.1 DTN Architecture

4.1.1 Strategies

To overcome the obstacle of lack of end-to-end connectivity and the long periods of disruptions and/or the long propagation delays, DTN-architecture [15] relies on a store-and-forward, hob-by-hop message delivery strategy, accompanied by the transfer of reliable delivery responsibility (custody transfer) among the nodes through the path to the destination node or the network gateway. By transferring the custody together with the messages, providing reliability is supposed to cost much less in terms of re-transmissions, delivery delay, data rate and energy consumption. Indeed, imagine a transfer failure occuring in some node close to the destination.

	IPNs	Disconnected Areas and WSNs	VDTNs
No End-to-End Connections	+	+	+
Frequent Channel Errors	+	+	+
Limited transmission opportunities	+	+	+
High Asymmetric Links	+	+	+
Low data rates - High delivery delay	+	+	+
Heterogeneous Network Environments	+	+	+
Scheduled/ Predetermined Contacts	+	+	
Bulk data transfer	+	+	+
Lack of Infrastructure	+	+	
Limited Resources		+	
Infrastructure exists			+
High Mobility			+
Reliability	+	+	+
Congestion Control	+	+	+
In order Delivery	+	+	+

Table 1: Characteristics and Requirements per environment (+ means dependency on the specific application)

Would it be less costly to re-transfer the failed data from the source node of the initial message or from a neighbor node? Additionally, it is expected that a node close to the one who failed will have a better image of alternative routes to forward the data than the source node. However, we should note that a hop-by-hop reliability strategy does not guarantee end-to-end reliability, as stated in [18].

For this reason a special naming scheme and compatibility with Network specific convergence layers (e.g. [19,20]) is supported. DTN architecture aims also in providing the necessary interoperability to support connectivity among different Networking environments.

4.1.2 The Bundle and Convergence Layers

The Bundle Layer is the most basic novelty introduced by the DTN architecture. It constitutes a new sublayer within the application layer of the protocol stack as you can see in figure 1, which is responsible to provide the majority of DTN-specific operations.

Bundle Construction and Fragmentation

Bundles are the basic data units within the DTN architecture and they are constructed out of the Application Data Units (ADUs) within the Bundle Layer, which is, then, the responsible entity for providing end-to-end reliable data transfer to the application. ADUs, forwarded to the Bundle Layer, are of arbitrary length [15] and it is then the responsibility of the Bundle Agents to construct Bundles out of them constituted by two or more blocks. Through the Network-spesific convergence layer (figure 1), Bundles can be further split into more fragments in order to fit in the Network specific requirements (e.g. Maximum Transmission Unit (MTU), expected connection duration, channel state indications etc.). Additionally, the Convergence Layer may offer the service of reactive fragmentation in case it is informed that only part of the previously transmitted Bundle was successfully received by the next hop. This has the obvious benefit of avoiding to retransmit whole Bundles when disconnection appears during some Bundle transfer.

Store-and-forward

Through this mechanism the Bundle Protocol aims to provide a message transfer service and an optional virtual end-to-end reliability service (through custody transfer). Thus, the Bundle Protocol Agent of each Bundle node is responsible for storing the Bundles in its local secondary memory, awaiting for forwarding them when connectivity is available.

Addressing and Late-Binding

An Endpoint Identifier is used to identify a DTN endpoint. An Endpoint refers to a set of one or more DTN Nodes. According to [15], using an EID, a node should be able to determine the Minimum Reception Group (i.e. The minimum amount

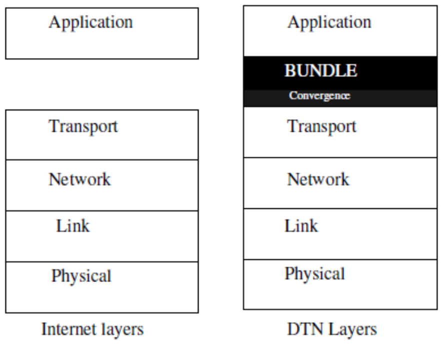

Figure 1: Internet and DTN stacks

of nodes of an endpoint which have to receive a Bundle in order for the transfer to be considered successful). An EID has the form of the example shown in figure 2. Late Binding introduced by the DTN Architecture allows for the translation of the Scheme Specific Part (SSP) of the destination EID to some specific EID or some lower layer address, late within the path to the destination Endpoint, instead of doing it at the source node as in the traditional DNS Internet-based schemes. If we imagine a frequently disrupted network environment, this strategy can be an important asset in the sense that it can avoid utilizing obsolete binding information due to the transit time being higher than the actual validity of a binding [15].

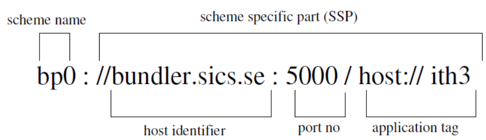

Figure 2: An example of EID

Custody transfer

This is the mechanism that allows reliability assurance service to move closer to the destination Bundle Node, following the transfer of the respective Bundles. In this way, the responsibility for Bundles' re-transmission doesn't have to stay within the source of the Bundle but it can be transferred to other nodes closer to the destination. Not all of the nodes throughout a route to the destination have to be custodians and the choice should be based on the amount of available resources that the candidate nodes possess (e.g. Buffer space, Energy level).

Custody Signaling

Parameters of Bundles' transmission such as the originating timestamp, useful life indicator, class of service designator and the Bundle length are included within custody signaling and they can be beneficial for scheduling and routing decisions [15]. More details regarding custody signaling can be found in [19].

4.2 DTN-based Protocols

After having presented the basic principles and novelties introduced by the DTN architecture we will continue now by presenting analytically transport, convergence and application layer protocols which operate within the Bundling, store-and-forward concept.

TCP Convergence Layer Protocol

TCP-based convergence Layer protocol (TCPCL) uses the well known TCP to provide reliable communication services between DTN nodes. Based on its specification [30], there should be an establishment of a TCPCL connection between two Bundle nodes, separately from the establishment of the TCP connection. This connection is terminated when the TCP connection is also terminated for any reason. As a convergence layer, TCPCL is the interface through which the communicating Bundle nodes exchange their EIDs. It supports proactive and (optionally) reactive fragmentation. Also, through the periodically transmitted KEEPALIVE messages, each sender aims to retain the liveness of the connection during inactivity periods.

In [31], Ruhai Wang et Al make an evaluation of TCPCL for Long-Delay Cislunar Communications, where they verify its problematic behavior for environments with long propagation delays, large Bit Error Rates (BERs) and/or frequent link disruptions. This behavior is due to TCP's nature which suffers from the particular weaknesses that we have already described. For their evaluation, they compare TCPCL/TCP implementation to an LTPCL/UDP implementation (we will describe LTP in the next section) as well as an hybrid implementation, which uses TCPCL/TCP for the segments of the end-to-end path with shorter propagation delays and/or lower BERs and LTPCL/UDP for the rest segments of the path. An interesting result is that, for an environment with a mix of long and shorter-delay links, the Hybrid performs better than pure LTPCL/UDP.

UDP Convergence Layer Protocol (UDPCL) over UDP

As we know, UDP is an unreliable protocol without congestion control support. For this reason, as stated in [32], "UDP can only be used on a local network, or in cases where the DTN node implements explicit congestion control". An additional concern when using UDP as a transport protocol is with respect to the UDP segment size. In [32], it is explained why it is crucial to make sure that the respective UDPCL protocol divides the initial Bundle into segments, each of which should be sufficiently small to fit into a Datagram.

In order to be able to function outside the scope of a local network, the authors of [32] suggest that Datagram Congestion Control Protocol (DCCP) could be used as a transport layer protocol.

Licklider Transmission Protocol (LTP) and LTP-T

LTP is a Link-Convergence Layer Protocol, which was initially designed for Interplanetary Deep Space Communication. However it can also operate on top of UDP aiming for local network communication scope. LTP provides point-to-point reliability and it was designed to be able to handle with long delay and/or disruption periods within local host-to-host communication. For this reason, it is based on lower layer indications regarding the link state and the presence of transmission opportunities (contacts). Since it is operating within a local link scope, it does not need to apply any routing and transport layer congestion control mechanisms. Thus, as stated in [20], it basically decouples Automatic Repeat Requests (ARQs) from the choice of parameters critical for the Network performance (timestamps of transmission, amount of data to be transmitted or received etc.). Additionally, thanks to its link layer nature, LTP is not a chatty protocol (i.e. it does not consume additional initial round trips on connection establishment, user authentication etc.). Instead, it is a stateful protocol in the sense that each session's parameters are predetermined and as a result the data transfer can begin immediately upon encountering the next node. Moreover, through the support for multiple parallel data block transfers (sessions), it can guarantee the avoidance of underutilization of the link. This means that in the case of some block transfer we won't need to wait until its ACK is received before we can transmit the next block. In this sense, the sessions may not arrive in order at the receiver's LTP Layer.

Regarding the retransmission timeout periods, we understand that LTP needed a mechanism to support an accurate calculation for them. As stated in [21], due to the unidirectional nature of LTP sessions and the great difference in RTTs between a segment transfer suffering from a disconnection and another one which does not, statistical analysis of round-trip history cannot be applied as a means of predicting round-trip time. Instead, LTP applies a deterministic approach for the approximation of the timer components which are more or less known (i.e. propagation time for the two transmission directions + processing times, queuing delays at sender and receiver sides + a safety margin to make sure to avoid unnecessary retransmissions). For the part of timer calculation which is more random and refers to the disconnection periods, LTP takes advantage of the link state cues provided, in order to suspend its timer when it receives indications for disconnections.

LTP provides another important flexibility feature, through its support for both reliable and unreliable data transfer. Specifically, each data block can be comprised of a reliable (red) part which needs to be acknowledged and an unreliable (green) part which doesn't need to be acknowledged. In that sense, LTP can provide both TCP and UDP based data transfer even within the same session. For the reliable part of each block, a Selective Acknowledgment mechanism is used (SACK) in order to verify the successful transfer of a block at the receiver. Specifically, the receiver generates a reception report upon reading the last segment of the red part (end of red-part (EORP)) of each block.

Terrestrial Use

Although LTP was designed for being used for Deep Space IPNs as a convergence layer within the DTN concept, it would be interesting to examine in which types of terrestrial networks it could be applied to as well. In [20] Stephen Farell et al suggest its eligibility for Wireless Sensor Networks, which use data mules. There, LTP can support scheduled communication with data mules very well by exploiting the lower-layer cues.

In [21,22,23] you can find the documents for LTP's motivation, specification and security issues respectively. In [24], you can find an analytical comparison between LTP and TCP as well as UDP convergence Layers.

LTP-T

LTP-T is a transport layer protocol, as an extension to LTP. In [25] you can find the reasoning behind this transport layer approach. It is supported, among the arguments, that if LTP is suitable for communication among each individual set of "hops", within an homogeneous network environment, why not utilizing LTP in a form that could support "multi-hop" communication as well. Other arguments refer to the ease-of-use of applications sitting upon a transport layer protocol, which is designed to support IP addressing and routing where possible. Such an approach appears to be more straightforward and compatible with the terrestrial Internet and in this concept it is contrasted to the more complex naming scheme adopted by the Bundle Protocol. The basic operation differences (e.g. Faster forwarding capability) and additions in order to support necessary transport layer mechanisms (e.g. End-to-end authentication, congestion notification, block fragmentation) are also described in [25]. In [26] Stephen Farell et al make an evaluation of an LTP-T implementation.

In [27] Syed Muhammad et al. evaluate and compare the performance of LTP and LTP-T in two different environments (i.e. Deep Space and Sensor Networks). The parameters of the simulations are: the amount of requested reliability (number of red segments), the frame loss ratio and the checkpoint ratios (i.e. how often ACK triggering indications appear within the received fragment sequence). The performance metrics are: the end-to-end delay and the goodput. They conclude that LTP-T performs better for both metrics when the control loops (i.e. duration of sending data and receiving report segments) are short or the challenging hops are isolated.

Saratoga

Saratoga is a transport layer protocol which primarily aims in local network file transfers (up to very large sizes), where central management of communication resources allows for scheduling their distribution among the network nodes. Its initial use was for earth-to-satellite communication (not Deep Space) within a Disaster Monitoring Constellation (DMC) scheme. In such a scheme, a connectivity pass between a satellite and a ground station lasts for 8 - 12 minutes with highly asymmetric links (i.e. 8.1 Mbps downlink, 9.6 Kbps uplink).

By using scheduling, Saratoga makes sure that the only source of packet loss is channel errors and as a result no need for congestion control is needed. Based on this scheduling policy, each application round can take full use of the dedicated links and multiplexing of flows can also be applied. In that sense Saratoga takes maximum advantage of the appearing transmission opportunities. However, if some particular application requires for interconnection with the public Internet or the link has to be shared among traffic flows, congestion control is needed and, for this reason, extensions for TCP-Friendly Rate Control (TFRC) or other congestion control mechanisms can be applied within Saratoga to support it [27]. Saratoga was, also, designed to provide the desired support for highly asymmetric links as well as file transfers ranging from relatively small to very large sizes.

Saratoga is a UDP-based protocol, supporting a Selective Negative Acknowledgment (SNACK) with holestofill strategy [27] for reliable retransmissions. Additionally, for reliable transfer, Data packets are sent using UDP with checksums turned on. Alternatively, for a more "error-tolerant" implementation, UDP-Lite [28] can be used instead. This policy minimizes the amount of payload checksumming required and lets the error detection and handling to the application. It can be useful in scenarios where we want to avoid discarding entire packets for some bit errors [29]. Along with its UDP nature, Saratoga can provide support for both IPv4 and IPv6 addressing, which assures the connectivity within a wide variety of link types.

The type of transactions provided by Saratoga are the following: Data download, Data Upload, Directory contents retrieval and delete content.

For being able to provide the services described, Saratoga introduces some particular packet types:

- BEACONS which aim to advertise the presence of Saratoga nodes among a local link by being sent on an IPv4 reserved, unforwardable multicast or an IPv6 link local multicast address [27]. In this sense Beacons act as a connectivity indicator with each advertising node.

- STATUS packets provide the feedback for the file transfer process and (optionally) a transaction establishment. Regarding the former, a STATUS packet can include a "holestofill" indication containing either a SNACK with the amount of data which was (possibly) not transferred successfully during the transaction and need to be retransmitted, or necessary information for the efficient resume of a previously interrupted transaction.

- REQUEST packets are used to initiate either a Data Download, or a Directory listing or a delete transaction.

- METADATA packets are (optional) parts of the data transfer and they provide info regarding the file characteristics (i.e. name, size of file, size of file descriptor).

In [27] you can find the analytical specification of Saratoga protocol. In [29], the authors present the notes on the use of Saratoga as transfer convergence layer protocol within the Bundle architecture. They initially highlight the differences with other popular convergence layer protocols (i.e. TCP, UDP and LTP), providing more insight on the comparison with LTP. As they state, although LTP and Saratoga have the same design base (both suitable for local link scope), their most significant difference is with respect to data flows' naming. In particular LTP's data blocks are unnamed and it requires a higher layer protocol to deal with naming (e.g. Bundle Protocol's EIDs). On the other hand Saratoga is able to perform named file transfers without any need from higher layers on this issue.

L. Wood et al in [34] provide some insight on the integration of Saratoga within the Bundle Architecture. They highlight the superiority points of Saratoga, operating as a convergence layer within the DTN model, over the already existing convergence layer protocols. In particular, they emphasize, among others, the fact that Saratoga can co-ordinate with IP (like UDP and TCP based solutions) and at the same time it is a reliable protocol without any segment size restrictions (advantages over UDP), not suffering from suboptimal performance due to Congestion Control strategies (advantage over TCP).

They suggest that a simple mechanism for the integration is to provide a shared directory between Saratoga and the Bundle Protocol, where the Bundle Agent can place its Bundles into files which can, then, either be uploaded by Saratoga to other peers or allow other peers to download them. Other necessary integrations for the interoperability with the Bundle Protocol are: advertising this capability through their BEACONs and provide an additional EID field within the BEACONs. Additionally, they suggest that the STATUS packets of a transfer, that Saratoga provides, can be used to trigger reactive fragmentation in the case of interrupted transfers.

	TCPCL/TCP	UDPCL/UDP	LTP	LTP-T	Saratoga
End-to-End Connections	+			+	
Frequent Channel Errors			+	+	+
Limited transmission opportunities		+	+	+	+
High Asymmetric Links		+	+	+	+
Low data rates - High delivery delay			+	+	+
Heterogeneous Network Environments				+	+
Scheduled/ Predetermined Contacts		+	+	+	+
Bulk data transfer	+		+	+	+
Lack of Infrastructure		+	+	+	+
Limited Resources			+		
High Mobility					
Reliability	End-to-End		Point-to-Point	End-to-End	Point-to-Point
Congestion Control	+			+	
In order Delivery	+		Yes, within the same session but not guaranteed in order delivery of sessions		
Operational Environments	Relatively short delays, less error-prone links		Long haul, local point-to-point, Deep Space		Local point-to-point, IP, earth-to-satellite.
Layer	Transport Convergence		Link Layer Convergence	Transport Layer	Transport Convergence

Table 2: Characteristics and Requirements per environment (+ means dependency on the specific application)

References

[1] Mohancur Sarkar, K.K. Shukla, K.S. Dasgupta, “A Survey of Transport Protocols for Deep Space Communication Network”. International Gernal of Computer Applications October 2011.

[2] Carlo Caini, Rosario Firrinceli, Mario Marchese, Tomaso de Cola, Michele Luglio, Cesare Roseti, Nedo Celandroni, Francesco Potorti, “Transport layer Protocols and architectures for satellite network”. International Journal of Satellite Communications and Networking, July 2005.

[3] Giorgos Papastergiou, Christos V. Samaras, Vassilis Tsaousidis, “Where does Transport Layer Fit into Space DTN Architecture?”. 5th Advanced Satellite Multimedia Systems and the 11th Signal Processing for Space Communications Workshop, 2010.

[4] Ioannis Psaras, Ning Wang, Rahim Tafazolli , “Six years since first DTN papers: Is there a clear target ?”, In: 1st Extreme Workshop on Communication (ExtremeCom 2009), Laponia, Sweden, 2009.

[5] Nikolaos Bezirgiannidis, Vassilis Tsaousidis, “Packet size and DTN Transport service: Evaluation on a DTN Testbed”, International Congress of Ultra Modern Telecommunications and Control Systems and Workshops, 2010.

[6] A.Seth, D.Kroeker, M. Zaharia, S.Guo, S.Keshav “Low-cost communication for Rural Internet Kiosks Using Mechanical Backhaul”, In: Proceedings of MobiCom, New York 2006.

[7] Melissa Ho, Kevin Fall, “Poster: Delay Tolerant Networking for Sensor Networks”, in In Proceedings of Sensor and Ad Hoc Communications and Networks (SECON), 2004.

[8] P.Juang, H.Oki, Y.Wang et al. “Energy Efficient Computing for Wildlife Tracking: Design Trade-offs and Early Experiences with ZebraNet”. In Proceedings of ASPLOS-X, October 2012.

[9] Wolf-Bastian Pottner, Felix Busching, Georg Von Zengen, Lars Wolf, “Data Elevators: Applying the Bundle Protocol in Delay Tolerant Wireless Sensor Networks”, Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 9th International Conference.

[10] Yu-TeHuang, Yi-Chao-Chen, Jyh-How Huang, Ling-Jyh Chen, Polly Huang, “YushanNet: A Delay Tolerant Wireless Sensor Network for Hiker Tracking in Yushan National Park”, 10th International Conference on Mobile Data Management: Systems, Services and Middleware, 2009.

[11] Chonggang Wang, Mahmoud Daneshmand, Bo Li, Kazem Sohraby, "A Survey of Transport Protocols for Wireless Sensor Networks", Network, IEEE (Volume:20 , Issue: 3), May-June 2006.

[12] Saima Zafar, "A Survey of Transport Layer Protocols for Wireless Sensor Networks", International Journal of Computer Applications (0975 - 8887) Volume 33 No.1, November 2011.

[13] A.J. Dinusha Rathnayaka, Vidyasagar M. Potdar, Atif Sharif, Saeed Sarencheh, Samintha Kuruppu, "Wireless Sensor Network Transport Protocol - A State of the Art", International Conference on Broadband, Wireless Computing, Communication and Applications, 2010.

[14] Simon Schutz, Lars Eggert, Stefan Schmid, Marcus Brunner, "Protocol Enhancements for Intermittently Connected Hosts", ACM SIGCOMM Computer Communication Review July 2005.

[15] V.Kerf et al, "Delay - Tolerant Architecture", IETF RFC 4838, April 2007.

[16] C.Holman, K.A. Harras, K.C. Almeroth, A. Lam, "A proactive data bundling system for intermittent mobile connections", In SECON of the IEEE, vol.1 pp. 216 - 255, Sep.2006.

[17] Bin Bib Chen, Mun Choon Chan "MobTorrent: A framework for Mobile Internet Access from Vehicles", IEEE INFOCOM , pp. 1404-1412, 2009.

[18] Khaled A. Harras and Kevin C. Almeroth, "Transport Layer issues in Delay Tolerant Mobile Networks", IFIP NETWORKING, 2006.

[19] K. Scott, S. Burleigh, "Bundle Protocol Specification", IETF RFC 5050, November 2007.

[20] Stephen Farell, "When TCP Breaks Delay- and Disruption Tolerant Networking", IEEE Internet Computing, July-August 2006.

[21] S. Burleigh, M. Ramadas, S. Farrel, "Licklider Transmission Protocol - Motivation", IETF RFC 5325, September 2008.

[22] M. Ramadas, S. Burleigh, S.Farell "Licklider Transmission Protocol Specification", IETF RFC 5326, September 2008.

[23] S. Farell, M. Ramadas, S. Burleigh, "Licklider Transmission Protocol - Security Extensions", IETF 5327, September 2008.

[24] Ruhai Wang, Scott C. Burleigh, Paavan Parikh, Chen-Jen Lin, Bo Sun "Licklider Transmission Protocol (LTP)-Based DTN for Cislunar Communications", IEEE/ ACM Transactions on Networking, Vol. 19, April 2011.

[25] Stephen Farell, Vinny Cahill, “LTP-T: A Generic Delay Tolerant Transport Protocol”.

[26] Stephen Farell, Vinny Cahill, “A DTN-Friendly Transport Protocol”, Satellite and Space Communications, 2007. IWSSC ’07.

[27] L. Wood, W. Eddy, W. Ivancic, C. Jackson, “Saratoga: A scalable Data Transfer Protocol draft-wood-tsvwg-saratoga-10”, Internet Draft, September 26 2011.

[28] L-A. Larzon, M. Degermark, S. Pink, L-E. Johnson, G. Fairhurst, “The Lightweight User Datagram Protocol (UDP Lite)”, IETF RFC 3828, July 2004.

[29] Lloyd Wood et Al. “Taking Saratoga from Space-Based Ground Sensors to Ground-Based Sensors”, Aerospace Conference, 2011 IEEE.

[30] M. Demmer, J. Ott, S. Perreault, “Delay Tolerant Networking TCP Convergence Layer Protocol ”, IETF, November 3 2008.

[31] Ruhai Wang et Al. “TCP Convergence Layer-Based Operation of DTN for Long-Delay Cislunar Communications”, IEEE Systems Journal, Vol. 4, NO.3, September 2010.

[32] H. Kruse, S. Jero, S. Ostermann, “Datagram Convergence Layers for DTN Bundle and LTP Protocols”, IETF, Sep. 2012.

[33] L. Eggert, G. Fairhurst, “Unicast UDP Usage Guidelines for Application Designers”, IETF RFC 5405, November 2008.

[34] L. Wood, J. McKim, W. M. Eddy, W. Ivancic, C. Jackson, “Using Saratoga with a Bundle Agent as a Convergence Layer for Delay- Tolerant Networking”, Internet Draft, March 26, 2012.

[35] L. Wood, Wesley M. Eddy et Al. “Saratoga: a Delay - Tolerant Networking convergence layer with efficient link utilization”, Delay Tolerant Networking session, Third International Workshop on Satellite and Space Communications, September 2007.

[36] L. Wood, Charles Smith et Al “Taking Saratoga from Space - Based Ground Sensors to Ground - Based Space Sensors”, IEEE Aerospace 2011 conference, March 2011.