Computer Networks 59 (2014) 244–263

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Delay Tolerant Payload Conditioning protocol ☆

Giorgos Papastergiou [a,*], Ioannis Alexiadis [a], Scott Burleigh [b], Vassilis Tsaoussidis [a]

[a] *Space Internetworking Center, Democritus University of Thrace, School of Engineering, Department of Electrical and Computer Engineering, Building A, Panepistimioupoli Xanthis, Kimmeria, 67100 Xanthi, Greece*
[b] *Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, La Cañada Flintridge, CA 91011, USA*

ARTICLE INFO

Article history:
Received 9 November 2012
Received in revised form 2 October 2013
Accepted 4 November 2013
Available online 9 November 2013

Keywords:
Delay Tolerant Networking
Aggregation
End-to-end reliability
Transport protocol
In-order delivery
Duplicate suppression

ABSTRACT

Delay Tolerant Networking (DTN) architecture is a new communication architecture developed to provide network connectivity in challenging environments. It forms a store-and-forward overlay network that employs persistent storage to deal with link disconnections. Consistent with its store-and-forward requirements, the design of the transport function of the DTN architecture was primarily based on hop-by-hop operations in preference to the traditional end-to-end communication model. As a result, pure end-to-end functionality is absent from the current DTN architecture and thus has been shifted towards applications. In this study, we highlight the benefits of having an additional layer of application-independent protocol offering transparent application data conditioning services end-to-end and we introduce Delay Tolerant Payload Conditioning (DTPC) protocol, a novel protocol that realizes this layer. DTPC protocol is an expandable, connectionless, reliable, sequenced transport protocol which extends the DTN architecture in a fashion that accords with the end-to-end principle, enabling the following services: (a) application data aggregation, (b) application-level reliability, (c) in-order delivery, and (d) duplicate suppression. DTPC was integrated into the JPL's Interplanetary Overlay Network (ION) DTN reference implementation, and its functionality was evaluated/validated through real-time experiments in a DTN testbed.

1. Introduction

The DTN architecture [1] and the accompanying Bundle Protocol (BP) [2] specification propose a means for data communication on potentially heterogeneous networks characterized by high propagation delays, frequent link disruptions and disconnections. Examples of such networks include deep-space networks, sensor-based networks, terrestrial wireless networks that cannot ordinarily maintain end-to-end connectivity, satellite networks with moderate delays and periodic connectivity, and underwater acoustic networks with moderate delays and frequent interruptions due to environmental factors. BP operates as an end-to-end message-oriented overlay placed above the transport layers of the networks that it interconnects and below applications, employing store-and-forward message (bundle) routing coupled with persistent storage to combat link disruptions. It uses a flexible naming scheme with late binding and offers a prioritization service based on three relative priority classes: bulk, normal and expedited. End-to-end transportation of bundles relies on the hop-by-hop operation of underlying

☆ The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007–2013_FP7-REGPOT-2010-1, SP4 Capacities, Coordination and Support Actions) under grant agreement n° 264226 (project title: Space Internetworking Center-SPICE). This paper reflects only the authors' views and the Community is not liable for any use that may be made of the information contained therein. Part of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

* Corresponding author. Address: Democritus University of Thrace, Department of Electrical and Computer Engineering, Panepistimioupoli Xanthis, Kimmeria, Building A, Office 1, Ground Floor, 67100 Xanthi, Greece. Tel.: +30 25410 79554; fax: +30 25410 79553.

E-mail addresses: gpapaste@ee.duth.gr (G. Papastergiou), ialex@ee.duth.gr (I. Alexiadis), scott.c.burleigh@jpl.nasa.gov (S. Burleigh), vtsaousi@ee.duth.gr (V. Tsaoussidis).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.bjp.2013.11.003

transport (or other) protocols (known as convergence layer protocols), such as TCP, UDP and Licklider Transmission Protocol (LTP, [3]). Reliable bundle delivery depends mainly on the reliability mechanisms offered by these protocols. Additionally, BP employs a custody transfer mechanism for enhancing reliability further. This mechanism allows the transfer of retransmission responsibility to successive downstream nodes as a bundle advances towards the destination in a hop-by-hop manner.

Practically, BP constitutes a network layer responsible for source to destination bundle delivery, possibly over heterogeneous networks, which resides directly below applications. Thus, unlike the traditional terrestrial protocol stack model, current DTN architecture does not include a transport layer for providing end-to-end services. Such services are left to the application. For example, in recognition of the potentially disconnected nature of DTNs, where an end-to-end closed-loop retransmission tactic is in most cases inefficient, retransmission of lost or corrupted data in the DTN architecture is performed only on a hop-by-hop basis, by the BP optional custody transfer mechanism along with any reliability mechanisms provided by the convergence layer protocols, such as LTP and TCP. As a result, applications requiring typical end-to-end reliability must implement their own end-to-end message reliability mechanisms.

In this study, we argue that even in these challenged environments, where an end-to-end path may never exist, it is both possible and desirable to provide end-to-end transport services. We detail the absence of significant end-to-end services from the DTN architecture and we investigate the benefits that can be offered by adding to DTN a true end-to-end transport layer of protocol for application data conditioning. This layer is placed architecturally above the "bundle" layer and below the application layer, offering end-to-end services that are absent from current DTN architecture.

In that context, we introduce Delay Tolerant Payload Conditioning (DTPC) protocol, a novel protocol that extends DTN architecture in a fashion that accords with the end-to-end principle. Being an end-to-end protocol, DTPC protocol needs to operate only at the endpoints of the communication system. DTPC protocol is an expandable, connectionless, reliable, sequenced transport protocol designed to be used on top of the BP offering the following services:

- Controlled aggregation of application data units (ADUs) with application-specific elision

In order to regulate the overhead introduced by BP protocol when large volumes of small ADUs are transmitted, DTPC protocol offers an aggregation service that aggregates ADUs, possibly from different applications residing in the same node, which have the same destination and require the same quality of service (class-of-service, custody transfer, etc.). The aggregation service is controlled by two limits: a length limit sets a maximum bound on the total size of aggregated ADUs and a time limit prevents undue delay before transmission of data during periods of low activity. Additionally, the aggregation service is coupled with an optional elision service that enables applications to remove obsolete or redundant ADUs from aggregated data units before transmission, based on application-specific criteria.

- End-to-end reliability

DTPC protocol provides an additional degree of assurance in the delivery of application data when lower layer reliability mechanisms fail, by using an end-to-end ARQ mechanism that is based on positive acknowledgments. Retransmission timeout intervals are worst-case values based on data lifetime and limits on the number of retransmissions, rather than on round-trip-time estimations.

- In-order delivery

Each DTPC protocol data unit is uniquely identified by a sequence number, enabling delivery of the contained application data units in transmission order. Additionally, DTPC protocol offers a "latest delivery" service that allows for in-order delivery of application data with relaxed completeness constraints, such that holes in the receiving sequence are permitted as long as the missing data units are considered expired. This "latest delivery" service assures that out-of-order data units never get stranded or expire at the receiver waiting for the reception order to be restored.

- Duplicate suppression

DTPC protocol suppresses the delivery of application data units (ADUs) that have already been received or that are considered expired, ensuring that duplicate or expired ADUs are never received by an application.

Due to its flexible design, DTPC protocol can be easily adjusted to unforeseen usage scenarios. In that sense, DTPC protocol can be viewed as an application-independent framework for injecting end-to-end characteristics into the DTN architecture.

The remainder of the paper is organized as follows: In Section 2 we present the necessary background and discuss the issues in the current DTN architecture that motivated this work. In Section 3 the design concepts and operation of DTPC protocol are described in detail. In Section 4 we elaborate on the experimental methodology, metrics and evaluation cases. We present the results of our experimental analysis in Section 5. Finally, Section 6 concludes this paper.

2. Motivation and background

We begin our discussion with the observation that BP introduces significant overhead to applications transmitting large volumes of small-sized data units, such as DTN-based continuous situation-awareness applications and telemetry applications. This overhead consumes a significant share of the available bandwidth and decreases the utilization of the communication channel. Even when Compressed Bundle Header Encoding (CBHE, [4]) is