Computer Networks 87 (2015) 59–77

Contents lists available at ScienceDirect

Computer Networks

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

A mobile code bundle extension for application-defined routing in delay and disruption tolerant networking

Carlos Borrego*, Sergi Robles, Angela Fabregues, Adrián Sánchez-Carmona

Departament d'Enginyeria de la Informació i de les Comunicacions, Universitat Autònoma de Barcelona, Barcelona, Spain

ARTICLE INFO

Article history:
Received 15 October 2014
Revised 21 April 2015
Accepted 28 May 2015
Available online 9 June 2015

Keywords:
Active-DTN
DTN
Routing
Bundle Protocol
Mobile-code
Disaster recovery

ABSTRACT

In this paper, we introduce software code to improve Delay and Disruption Tolerant Networking (DTN) performance. DTN is extremely useful when source and destination nodes are intermittently connected. DTN implementations use application-specific routing algorithms to overcome those limitations. However, current implementations do not support the concurrent execution of several routing algorithms. In this paper, we contribute to this issue providing a solution that consists on extending the messages being communicated by incorporating software code for forwarding, lifetime control and prioritisation purposes. Our proposal stems from the idea of moving the routing algorithms from the host to the message. This solution is compatible with Bundle Protocol (BP) and facilitates the deployment of applications with new routing needs. A real case study based on an emergency scenario is presented to provide details of a real implementation. Several simulations are presented to prove the feasibility and usability of the system and to analyse its performance in comparison to state-of-the-art approaches.

1. Introduction

Portable devices such as mobile phones or tablets are widely used in daily life. They are generally equipped with wireless-enabled communication, GPS receivers and/or touch screens. The existence of these devices has improved outdoor applications in a great variety of situations. Particularly, they can directly connect with each other. The most popular network configurations for this kind of connections are Ad hoc and Mobile Ad hoc (MANET) [4]. This kind of configuration does not require other infrastructure than the connected devices themselves. New communication paradigms are emerging to fill the void for some specific settings not covered by Ad hoc and MANET. This is the case of Delay and Disruption Tolerant Networking (DTN) RFC 4838 [15].

DTN implementations are extremely useful when no concomitant network links connect the source and the destination nodes at transmission time. This is typical in emergency and disaster recovery scenarios where the conventional network infrastructure collapses. It is also common in the space and in undeveloped areas where no conventional network is available. Ad hoc communications are defined when nodes keep connected solely during message transmission. This is also the case of MANET. The capability of DTN to work with intermittently connected nodes makes it suitable for scenarios like the previously described. The lack of infrastructure requirements makes DTN applicable to restoring network connectivity even coexisting with other networking solutions. DTN approaches provide a cheap, easy and ready-to-use deployment.

DTN has strong foundations such as the Bundle Protocol (BP), RFC 5050 [46]. Many groups have been working on their formalities for several years [22,38]. Moreover, NASA is using DTN in the International Space Station [16]. However, there are still a number of issues to be solved, with routing being

* Corresponding author. Tel.: +34 93 5814505.
E-mail addresses: cborrego@deic.uab.cat, cborrego@deic.uab.es (C. Borrego), sergi.robles@uab.cat (S. Robles), fabregues@deic.uab.cat (A. Fabregues), adria.sanchez@deic.uab.cat (A. Sánchez-Carmona).

http://dx.doi.org/10.1016/j.comnet.2015.05.017
1389-1286/

one of the most problematic. We consider routing as the process of selecting *which* messages are to be transmitted (prioritisation and lifetime control) and *where* to (forwarding). These issues need innovative solutions that have not been normally used on the Internet. The rationale for this is that applications running on such poorly connected networks require different routing algorithms for their specific problems. In contrast to what happens on the Internet, no general purpose routing algorithms exist which satisfy the requirements of all applications at once. One of the design principles of the Delay-Tolerant Networking Architecture, as defined in [15], is to:

> *"Provide coarse-grained classes of service, delivery options, and a way to express the useful lifetime of data to allow the network to better deliver data in serving the needs of applications."*

This principle can be summarised as "DTN routing algorithms must be application-defined". The problem lies in the fact that with current specifications, any DTN implementation must be devoted to a single application or at least to applications with similar routing needs. This often implies requiring to establish a network per application. With the aim of following this principle of allowing application diversity, we propose a solution without the need for any software deployment or maintenance. Our proposal incorporates the application-specific routing algorithms as software code to be carried by the messages. By this way, messages can be forwarded, lifetime controlled and prioritised using algorithms suitable for the applications to which they belong. The concept underneath is mobile code [42], that is, a well-known technology designed for this purpose.

In this paper, we define Active-DTN, that is, a DTN where messages have an active decision behaviour instead of being pure data wrappers. This behaviour makes possible the use of application-specific routing allowing, at the same time, the concurrent execution of applications with different routing needs. Active-DTN follows BP in order to be compatible with current implementations. It defines the Mobile code Metadata Extension Block (MMEB) to allow code to be carried, and three code types are proposed: the forwarding, the lifetime control and the priority of messages. This paper also provides the description of aDTN, the first Active-DTN implementation and a discussion about how can it be applied to a disaster recovery scenario with several rescue teams sharing the same network. The simulation experiments performed in the context of disaster recovery show the soundness of this proposal evaluating several components of Active-DTN.

The paper starts with all the relevant state of the art information, in Section 2, paying special attention to the BP. Next, we provide a full description of Active-DTN in Section 3 and its block definition as MMEB in Section 4. Section 5 defines three code types for forwarding, lifetime control and priority. In Section 6, some security considerations are discussed. Then, aDTN, the first Active-DTN implementation, is explained in Section 7. The paper follows with a discussion about its applicability to disaster recovery, in Section 8, followed by Section 9 with the simulation experiments. Finally, Section 10 contains the conclusions.

2. Related work

There are two main developing paradigms related to networks characterised by intermittent connectivity, asymmetric bandwidths, long and variable latency and ambiguous mobility patterns. The most relevant to this study is the paradigm of the IRTF Delay Tolerant Network Research group.[1] They defined the DTN architecture, RFC 4838 [15], and BP, RFC 5050 [46], which are an abstract service description for the exchange of what they denoted by bundles in DTN. A bundle is a series of contiguous data blocks containing enough semantic information to allow the application to make progress where an individual block may not. These bundles carry the application information from a source to a destination following the store-carry-and-forward paradigm. That is: each node stores application data that can forward whenever the node contacts another node. The bundle architecture behaves as an overlay network.

The second paradigm belongs to the Haggle project, [45]. Haggle is a one-way communication architecture which main purpose is to take advantage of brief connection opportunities. As in the BP, Haggle proposes solutions to scenarios with intermittent network availability suffering from long delays by switching messages and performing opportunity-oriented behaviours. Haggle allows the application messages on every DTN node to choose among a limited number of routing protocols. These routing protocols must be previously deployed. However, in intermittently connected scenarios, this deployment is not easy to conduct. In DTN, due to the idiosyncrasy of the network, performing such deployments does not guarantee that when a message arrives at an intermediate node their optimal routing protocol will be available. Additionally, these nodes are normally hardware-limited and consequently the local maintenance of the different routing protocols is not easy to perform, as pointed out in studies like [37].

The two previously introduced paradigms accept disruptions as the idiosyncrasy of the problem. Contrarily, studies like [1] and [29] propose different ways of linking the existing partitioned networks. These proposals may be useful in some situations. Unfortunately, they are essentially based on adding infrastructure elements to the network. This is not always feasible due to the complexity added to the system, the economic cost of the solution, or the difficulty of finding the best location for the links. Furthermore, most of these proposals fail to consider networks of mobile elements such as those required by a practical application such as disaster recovery [34].

Interesting proposals, such as [52], suggest the possible use of the software-defined network technology (SDN) [48] in DTN and in particular in emergency scenarios. SDN is an open standards-based and vendor-neutral network approach that allows the applications to centrally control the behaviour of the network in a very agile way. By means of a logically centralised software program, SDN separates the network control from the forwarding functions. However,

[1] The main site of the IRTF Delay Tolerant Network Research group is http://www.dtnrg.org.