A DTN Gateway-Based Architecture for Web Access in Space Internet

Mwamba Kasongo Dahouda[1], Ducsun Lim[1], Kyungrak Lee[2], and Inwhee Joe[1(✉)]

[1] Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul, Korea
{dahouda37,imcoms,iwjoe}@hanyang.ac.kr
[2] ETRI, Gajeong-ro, Yuseong-gu, Daejeon, Korea
krlee@etri.re.kr

Abstract. A Delay/Disruption-Tolerant Network (DTN) has received more attention as an expected solution in situations where it is difficult to provide a stable and continuous Internet connection. The purpose of this research is to design a client–server model for space Internet in order to have access to the web documents hosted by an interplanetary node. One of the DTN implementations named Interplanetary Overlay Network (ION) is introduced to provide the main functions to allow networked communications in deep space. In this paper, we propose an architecture for web document access in space Internet over DTN. Our web architecture is based on the DTN Gateway between terrestrial and space Internet that acts as a web proxy and bundle forwarder according to the type of web documents; it works as a web proxy for static web documents, while it works as bundle forwarder between DTN client and DTN server for other web documents. The HyperText Transfer Protocol (HTTP) requests and responses are formatted onto bundles by using *bpsendfile* and *bprecvfile*, and the DTN Gateway forwards them as a contact basis in space Internet. We have reported the evaluation results from our implementation by measuring the bit error rate and delay according to the size of static web documents.

Keywords: DTN · HTTP · Space Internet · Web architecture

1 Introduction

Delay/Disruption-Tolerant Network (DTN) was introduced as a solution for interplanetary communication [1], where normal inter-networking protocols, dependent on low end-to-end delay, low packet loss rates, and a stable connection would not suffice. However, in addition to the interplanetary communication, the DTN technologies can be applied to a variety of challenged networks as presented in reference [2], e.g., terrestrial mobile networks, military ad-hoc networks, vehicular networks, sensor networks, etc. To provide the communication quality that applications demand, these networks need to consider various network characteristics like high latency, limited bandwidth, high packet-loss rate, path instability,

R. Silhavy (Ed.): CSOC 2020, AISC 1224, pp. 156–164, 2020.
https://doi.org/10.1007/978-3-030-51965-0_13

etc. In the Space Internet, the network architecture reduces intermittent communication issues by addressing technical problems in heterogeneous networks that lack continuous connectivity. It takes a different approach by relying exclusively on asynchronous communications.

Throughout the world, one of the most routinely used sets of protocols is the Transmission Control Protocol/Internet Protocol (TCP/IP) which include commands and facilities that allow, for example, transfer files between systems; but Space Internet uses the bundle protocol/ Licklider Transmission Protocol (BP/LTP) as underlying protocol; allowing the file transmission from one node to another. A web page is a document commonly written in HyperText Markup Language that is accessible through the Internet using an Internet browser[1]. The HyperText Transfer Protocol is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted; browsing a web page is an example of request–response on synchronous communication, our study aims to retrieve a static web document in space Internet.

This paper presents a one-to-one model of HTTP requests and responses onto bundles; the HTTP request is formatted and then inserted into the payload of the bundles then forwarded by DTN Gateway [4] as illustrated in Fig. 1. If the DTN node client wants to retrieve the web document, it sends a request; all the resources required to display a web page are retrieved upon a single request and returned in bundled format, ideally a single response [4] by the DTN node server; and the web page is displayed on the client node's internet browser. The DTN node server is expected to have locally stored knowledge; that identifies all the associated objects for each resource so that all objects making up a web page can be returned in a single response [4] for reducing the round trip time problem.

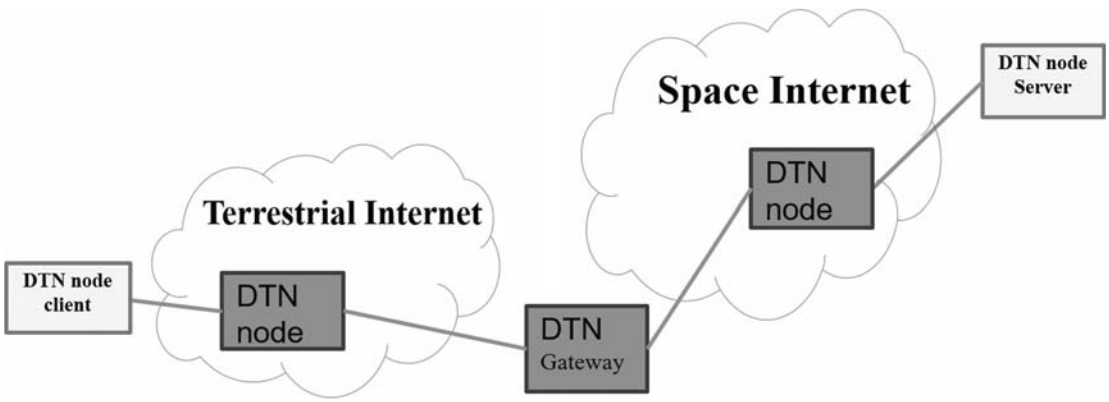

Fig. 1. Client-Server Architecture for Space Internet. DTN gateways interconnect regions running potentially dissimilar protocol stacks

The HTTP request and response must be encapsulated into the bundle protocol to allow communication between DTN nodes [9]. In our client-server model;

[1] https://www.computerhope.com/jargon/w/webpage.htm.

the web page is retrieved by using the wget; a utility for non-interactive download of files from the web, together with *bpsendfile* and *bprecvfile*, ION–DTN file transmission utilities; allows us to realize a bidirectional communication for retrieving a web page hosted by a remote web server in a space communication with intermittent connectivity.

The paper is organized as follows: Sect. 2 presents related work, and Sect. 3 describes our architecture and the following Sect. 4 the implementation. In Sect. 5, we give our conclusion and future work.

2 Related Work

We use client-server model communications together with offline operation and apply the concept of delay-tolerant networking to enable offline static web document access. The concepts presented in this thesis are largely based on the work by Ott and Kutscher in [4,5]. They introduced the ideas of sending aggregates of HTTP resources in bundles and describing resource dependencies with metadata files on the server. A similar approach to enabling communications under intermittent connectivity, based on proactive data retrieval and bundling, has been discussed in [6].

Another concept is the usage of the Bundle Protocol to transport HTTP-requests and -responses. In [7], adapt HTTP to the DTN environment by introducing the concept of resource bundling and present an approach for a conversion of the different message types and the required components. With the limitation of the HTTP protocol in some parts of the network, additional gateways for the conversion between HTTP and Bundle Protocol are always necessary. In [8], describes the application technology of Delay Tolerant Network (DTN) using Space Internet Protocol; The schedule model is based on the file transmission/reception, message transmission, and video streaming service models, which are services targeted by the space Internet. The bundle with the highest priority was sent first. In our case; we define the priority high while sending the HTTP request and response message. In [3], present the design and evaluation of a convergence layer for the DTN implementation IBR-DTN using HTTP as underlying protocol; long-polling was used to realize a bidirectional communication.

3 Architecture

This section presents our client–server model architecture for space Internet; the three-node networks were designed. It is shown in Fig. 2. Node 2 acts as a DTN Gateway between Node 1 and Node 3. This is achieved in the configuration files by making node 2 the gateway for node 1 and node 3. We have designed our model to allow us the follow of the network traffic and node 2 acted as DTN gateways which are bundle forwarders. In our model; node 1 (node client) requests a URL from a node 3 (node webserver) via HTTP. That node web server responds with an HTML page or a raw response (HTML, style CSS, images, etc.).

The architecture is subdivided into 3 main components as depicted in Fig. 2. In DTN, the end-to-end connectivity between source nodes and destination nodes rarely exists. Due to this disruption connectivity, messages should be forwarded to intermediate participating nodes by using the "store-carry-forward" mechanism, if the source node intends to deliver messages to the destination node [10].

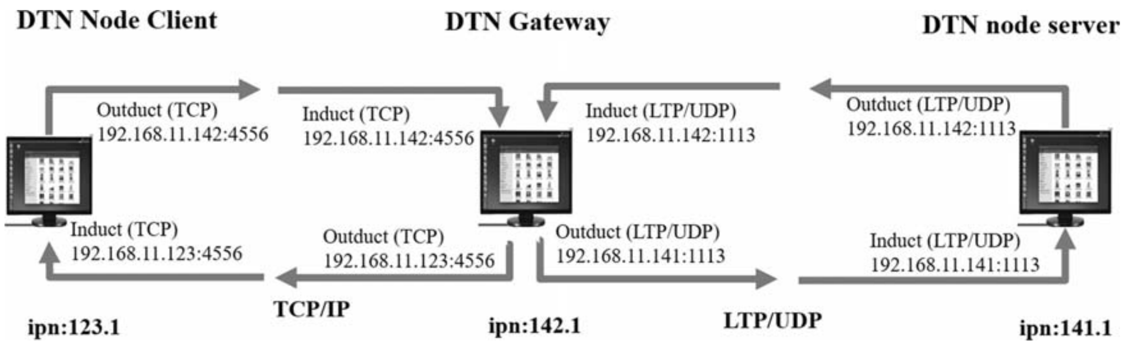

Fig. 2. Client-Server DTN experimental setup

3.1 DTN Node Client

The node client includes IP Neighbor Discovery functionality; when the HTTP GET request is sent to the webserver, it is first inserted into the bundle named as request bundle, and then transmitted, forwarded by the node 2 through disruption network by the bundle protocol to the destination node 3.

3.2 DTN Gateway

The DTN node Gateway forwards all bundles from client to server and from server to client; and rely on the bundle protocol (BP), Bundle Protocol truncates application data into bundles of different size and Licklider Transmission Protocol is used as the convergence layer protocol, encapsulating bundles into blocks and fragmenting blocks into segments [3]. LTP is one of the most known protocols that are used in the convergence layer under Bundle Protocol [3] and allow file transmission in a disruption network or bundle transmission from source to destination in a space Internet.

3.3 DTN Node Server

The node server includes the same functionalities as the node client; besides, apache tomcat and also a Java web Application implementing a Servlet; allowing the webserver to operate which provides the resource or the web page requested by the client. The DTN node Server support resource collection [4] for DTN-based HTTP access: as (static) HTTP resources are stored in files (e.g., response.html). In our implementation which is described in the next section; we have used the *wget* that performs the recursive downloading which can convert all links for offline viewing of local HTML.

The *wget* can follow links in HTML and XHTML pages and create local versions of remote websites, fully recreating the directory structure of the original site.[2] The node server uses the *wget* to recursively obtain all the resources that are necessary to display the web document; as described in Fig. 3, we presented the flow diagram describing the communication between node client and node sever; the resources returned by *wget* are retrieved from the file system and packed into DTN bundle for forwarding.

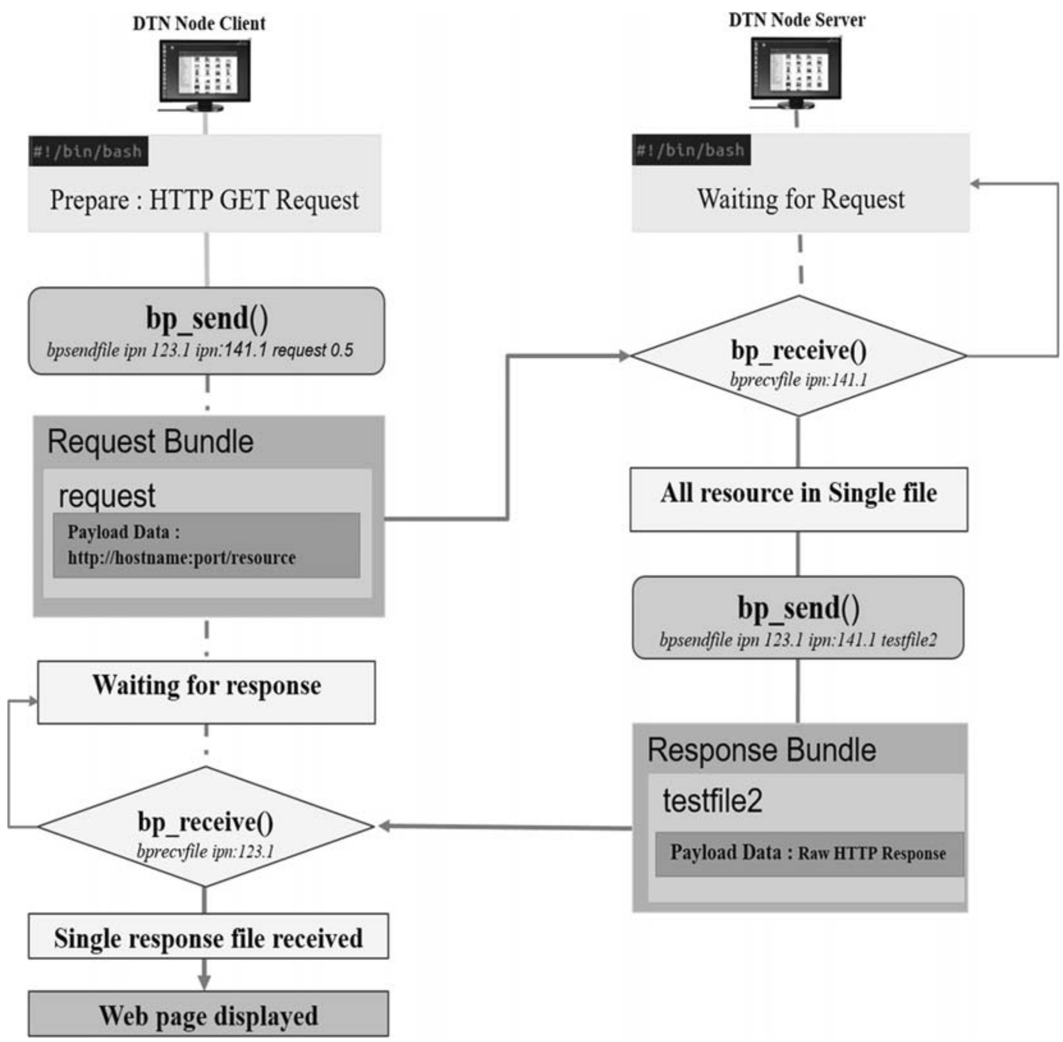

Fig. 3. Client–Server flow diagram

2 https://www.gnu.org/software/wget/.

The server web hosts the web pages which typically contain multiple resources, such as images, stylesheets and JavaScript; all the resources will be sent in a single file.

4 Implementation and Experiments

The following two subsections describe the essential components of our experimental methodology, the implementation and the experiment results.

4.1 Implementation

The node client was a virtual machine; the node 2 was equipped with a 1.9 GHz Intel Celeron(R) CPU 1005M, 4 GB RAM, 893.4 GB as a Capacity of Hard Drive, and 802.11 WiFi Adapter. The connection between nodes was supported by a Wireless-N INFINITI Gigabit High Power Broadband and Access point; the Server node was equipped with a 2.2 GHz Intel(R) Core(TM)i7-2675QM CPU, 4 GB RAM, 500 GB of Hard Drive Disk with a Broadcom 802.11n Network Adapter.

On both machines, Ubuntu 16.04 LTS was installed as the host operating system. As DTN implementations for the experiments, we used the ION 3.6.2 version. We designed and implemented a simple application using the Linux bash shell scripting. The bash script was executed on both nodes. As shown in Fig. 3; the DTN network traffic was managed by a WiFi access point. On both nodes start ION with the command:*ionstart* with the configuration as parameter in order to check the persistent storage of bundles in the case of unavailability of the next DTN hop.

The bash scripts have the following steps for request-response:

1. DTN node client send an HTTP request by using:
   bpsendfile ipn:123.1 ipn:141.1 request class of service
   The transmitted bundle contains the HTTP request in the payload Data; we have set up the class of service to 0.5 that gives the bundle the highest possible priority during transmission to the destination node.
2. The node server receive the request bundle by using:*bprecvfile ipn:141.1*
3. The node server recover the HTTP request, retrieve the web page requested.
4. The node server put the response in testfile2 and send back the Raw HTTP Response (html, style CSS, JavaScript and Images) by using:
   bpsendfile ipn:123.1 ipn:141.1 testfile2
5. The transmitted bundle contains the Raw HTTP response in the payload Data.
6. The node client receive the Raw HTTP response by using:
   bprecvfile ipn:123.1
   All the Raw HTTP response messages will be inserted in a file response.html.
7. Using Firefox; the Web page is displayed in the node client's Web Browser (Offline web access)

The shell scripts for request scenario are shown below.

```
#!/bin/bash

SOURCE_NODE=ipn:123.1
DESTINATION_NODE=ipn:141.1
SERVICE_CLASS=0.1
REQUEST="http://192.168.11.106:8080/mnilabs/"
echo "$REQUEST"
echo "$REQUEST" > request
bpsendfile $SOURCE_NODE $DESTINATION_NODE request $SERVICE_CLASS
bprecvfile $SOURCE_NODE 1
response=$(cat testfile1)
echo $response
echo "$response" >> response.html
kdesudo firefox response.html
exit
```

The shell scripts for response scenario are shown below.

```
#!/bin/bash

SOURCE_NODE=ipn:141.1
DESTINATION_NODE=ipn:123.1
recursive_download="wget -qO- -l10"
while true;
do
    echo "Incoming request's number : $((req=req+1))"
    bprecvfile $SOURCE_NODE 1
    REQUEST=$(cat testfile1)
echo "$REQUEST"
response=$($recursive_download " " $REQUEST)
echo "$response"
echo "$response" >> testfile2
bpsendfile $SOURCE_NODE $DESTINATION_NODE testfile2
done
```

4.2 Experimental Results

In the experiments, bit error rate and delay are investigated as the performance metric. When performing the request–response; experimentally demonstrated in terms of transmission; the bit error rate measurements show performance up to 1×10^{-4}. For all experiments; the bundle storage was based on the Simple Data Recorder (SDR). In the context of DTN application development, various sizes of web document needs to be transmitted. Consequently, as workload, we have used web documents of different size, namely: 48 KB, 62 KB, 190 KB, and 417 KB.

The different web documents contained one or many images; when the client node sends request; according to the size of the static web document, example

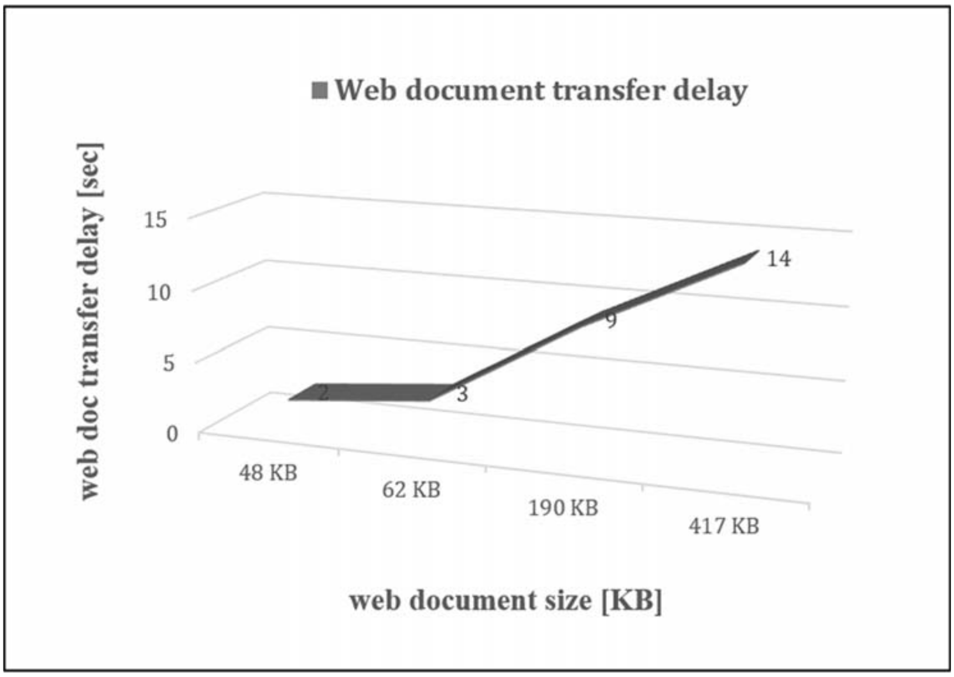

Fig. 4. Web document transfer delay

for; a document with 48 KB, after 2 s the web page will be displayed in the web browser of the client node, as shown in the Fig. 4; more we have a big size more the delay increases.

5 Conclusion

This paper designed a client-server architecture for Space Internet and presented an efficiency analysis of the static web document transmission in an intermittent network by using one of DTN implementations, namely ION-DTN. The testbed consisted of three nodes which exchanged bundles over an underlying WiFi environment, we have designed and implemented two simple bash-shell scripts, allowing HTTP requests and responses onto bundles. As future work we are planning to extend the number of nodes in the scenario and introduce intermittent–Opportunistic Contacts, and using the same approach, we can plan a dynamic web page access in a space Internet.

Acknowledgments. This work was supported by the Electronics and Telecommunication Research Institute [Development of Space Internet Technology for Korea Lunar Exploration] project in 2019.

References

1. Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., Travis, E., Weiss, H.: Delay-tolerant network architecture: the evolving interplanetary internet. Internet Draft draft-irtf-dtnrg-arch-01.txt, Expires, February 2003

2. Fall, K.: A delay-tolerant network architecture for challenged internets. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM 2003, pp. 27–34. ACM, New York (2003)
3. Bezirgiannidis, N., Tsaoussidis, V.: Packet size and DTN transport service: evaluation on a DTN testbed. In: Proceedings of the IEEE Conference, November 2010
4. Ott, J., Kutscher, D.: Bundling the Web: HTTP over DTN (2006)
5. Ott, J., Kutscher, D.: Applying DTN to Mobile Internet Access: An Experiment with HTTP (2005)
6. Holman, C., Harras, K.A., Almeroth, K.C., Lam, A.: A Proactive Data Bundling System for Intermittent Mobile Connections (2006)
7. Peltola, L.: Enabling DTN-based web access: the server side. Master's thesis, Helsinky University of Technology, April 2008
8. Kyung, D., Lim, D., Joe, I., Lee, K.: A streaming scheme based on contact scheduling for space internet. Korean Inst. Commun. Inf. Sci. **1**, 421–422 (2019)
9. Morgenroth, J., Pögel, T., Heitz, R., Wol, L.: Delay-Tolerant Networking in Restricted Networks. ACM, New York (2011)
10. Mao, Y., Zhou, C., Ling, Y., Lloret, J.: An Optimized Probabilistic Delay Tolerant Network (DTN) Routing Protocol Based on Scheduling Mechanism for Internet of Things (IoT), Sensors (Basel) **19**(2), 243 (2019). https://doi.org/10.3390/s19020243
11. The Interplanetary Overlay Network (ION). https://samsclass.info/129S/proj/ion-dtn.htm