Delay-Tolerant Networking
Internet Draft
Intended status: Informational
Expires: September 30, 2018

Taixin Li
Guanwen Li
Huachun Zhou
Beijing Jiaotong University
March 30, 2018

A Hybrid Integrity Assurance Strategy for Bundle Protocol
draft-li-dtn-hybrid-integrity-05.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html

This Internet-Draft will expire on September 30, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

Delay/Disruption Tolerant Networking (DTN) is designed for a severe environment where communication quality is not guaranteed. It works as an overlay network associated with Bundle Protocol (BP) and some convergence layer protocols like Licklider Transmission Protocol (LTP). However, there is no mechanism in both BP and LTP Protocol to ensure integrity of a packet with the granularity of bit. Since the integrity is crucial for packet transmission and necessary metadata consumes extra costs, there should be a strategy to decide which packets and how the packets are required to conduct integrity assurance based on network resources and user requirements. Hence, in this document, a hybrid integrity assurance strategy is proposed to ensure the different levels of integrity of bundles based on the status of network resources and the need of users.

Table of Contents

1. Introduction

Delay/Disruption Tolerant Networking (DTN) [RFC4838] is designed for a severe environment where connectivity of network is intermittent and communication quality is not guaranteed. It works as an overlay network associated with Bundle Protocol (BP) [RFC5050] and convergence layer protocols like Licklider Transmission Protocol (LTP) [RFC5325] [RFC5326]. BP, which is an application layer protocol, is based on a custody transfer mechanism and defines how to forward bundles in DTN, while LTP ensures the reliability of bundle transmission with the granularity of packet. However, there is no mechanism in both BP and LTP Protocol to ensure integrity of a packet with the granularity of bit. Integrity is crucial for packet transmission since errors in the header leads to some unexpected results while errors in the payload results in end-to-end retransmission and waste of limited storing and link resources.

BPSEC [I-D.ietf-dtn-bpsec-06] defines a security protocol providing end to end data integrity and confidentiality services for the Bundle Protocol. However, necessary checksum metadata consumes costs, so there should be a strategy to decide which packets and how the packets are required to conduct integrity assurance based on the network resources, such as buffer utilization rate, bandwidth, and packet loss rate.

In this document, we define a new type of extension block to carry the checksum field. Furthermore, we propose a hybrid integrity assurance strategy to ensure the different levels of integrity of bundles based on the status of network resources and the need of users. The intermediate nodes make a hop-by-hop integrity detection strategy according to network status while the end user makes an end-to-end integrity detection strategy.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Checksum Block Format

There are three parts in the bundle packet, primary block, payload block, and extension block. Extension Block is designed to carry additional information that DTN nodes can use to make processing decisions that are related to bundles.

We define a new type of extension block and use it to carry the checksum information in this document, and the basic format is based on [RFC6258], which defines DTN metadata extension block.

The format of checksum block is as follows:

Checksum Block Format:

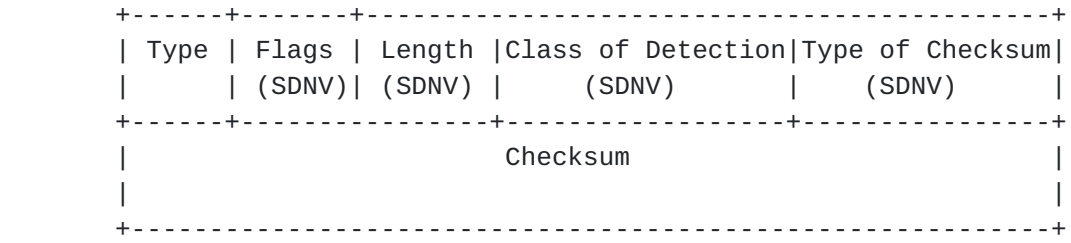

Figure 1 Checksum Block Format

o Block type code (1 byte) - defined in all bundle protocol blocks except the primary bundle block (as described in the Bundle Protocol). The block-type code for checksum is 0x20.

o Block processing control flags (SDNV) - defined in all bundle protocol blocks except the primary bundle block. SDNV encoding is described in the Bundle Protocol. The following block processing control flag MUST be set "4 - Discard block if it can't be processed", which means that if a bundle node receives a bundle with a checksum block and it is not capable of supporting the checksum block, it just discards this block without processing it.

o Block data length (SDNV) - defined in all bundle protocol blocks except the primary bundle block. SDNV encoding is described in the Bundle Protocol.

- Class of detection (SDNV) - (CoD) indicates the bundle's class of detection, which decides whether a bundle packet should conduct an integrity assurance and which part should be detected. CoD is decided by two factors, network resources status and user requirements. For now, it contains three types: 00 = inadequate resources, 01 = adequate resources, 10 = mandatory detection. Here, 00 and 01 are determined by the network status, and 10 is decided by the user. The users have a higher priority, which means that if CoD = 10, the bundle packets should be detected no matter what the network status is.

- Type of Checksum (SDNV) - (ToC) indicates the type of checksum data. For now, it contains four types: 00 = checksum of primary block, 01 = checksum of payload block, 10 = checksum of primary block and payload block, 11 = no checksum of either primary block or payload block.

- Checksum data - contains the raw checksum data itself, which is generated by some algorithms.

4. Processing Rules of Integrity Detection

As is discussed in [WOOD09] and [I-D.templin-dtnhiaps-00], integrity detection is required on intermediate nodes in addition to destination nodes. In order to make full use of the limited resources in the severe environments, both the source nodes and the intermediate nodes should monitor the usage rate of their resources such as the storage and link. Then different integrity assurance strategies will be made according to network resources status and user requirements. The integrity detection is called if the network resources are inadequate, the custody is needed, or it is demanded by end users. Besides, intermediate nodes detect the header/primary block or the payload block according to the Type of Checksum field carried in the checksum block. If there are errors in the packet data, forwarding process is stopped and retransmission is called. When the destination nodes receive packets, they detect the checksum block and if there are errors in the packet data, retransmission will be called.

4.1. Processing Rules in Source Nodes

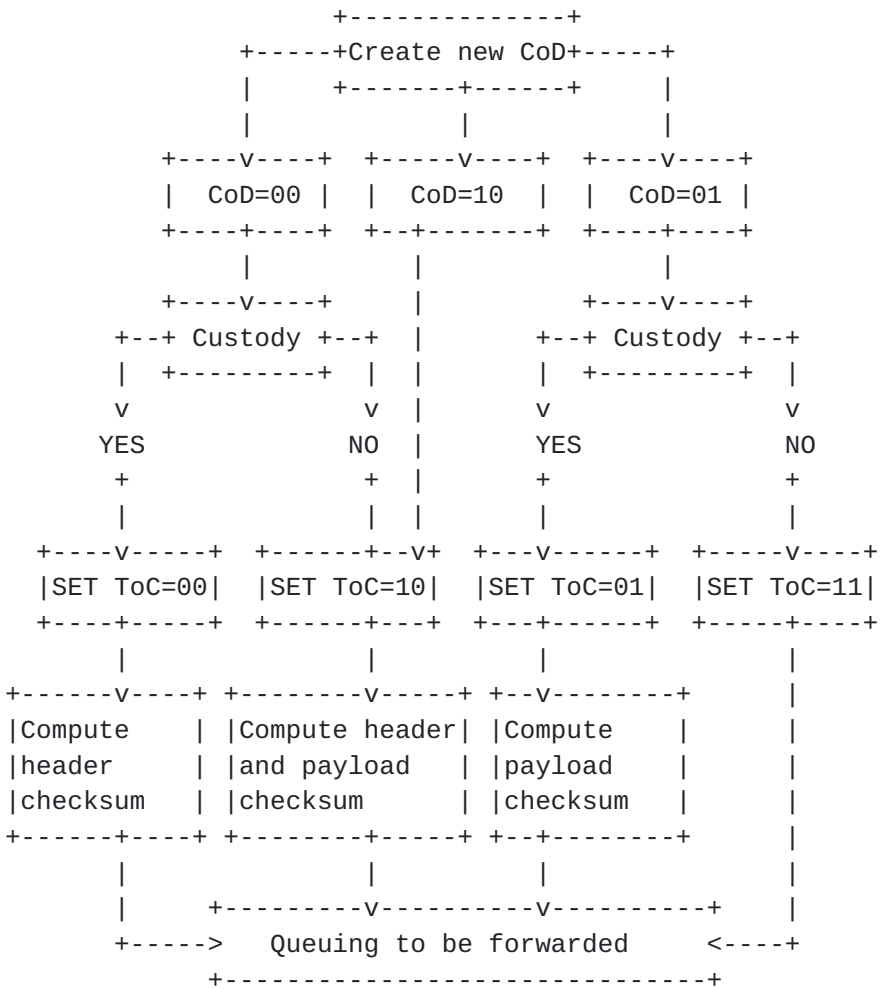

Figure 2 Processing Rules in Source Nodes

The processing rules in source nodes are shown in Figure 2. The source nodes collect the network link status, such as bandwidth and packet loss rate, and create Class of Resource (CoD). The algorithm of creating CoD is not discussed here.

If CoD=00 (inadequate resources), it means the network environment is severe and error prone. The source nodes read Bundle Processing Control Flags (defined in RFC5050). If custody is needed, Type of Checksum (ToC) will be set 10 (checksum of primary block and payload block), and the checksum of primary block and payload block will be computed by a designated algorithm. The algorithm is not discussed here. Then the Checksum data field will be filled. If custody is not needed, ToC will be set 00 (checksum of primary block), and the

checksum of primary block will be computed. Then the Checksum data field will be filled.

If CoD=01 (adequate resources), it means the network resources are relatively adequate. If custody is needed, ToC will be set 01 (checksum of payload block), and the checksum of payload block will be computed. Then the Checksum data field will be filled. If custody is not needed, ToC will be set 11 (no checksum of either primary block or payload block), no checksum calculation actions will be triggered. At last, the processed packets will queue and wait to be forwarded.

If CoD=10 (mandatory detection), it means that end users at the source node want their packets to be detected no matter what the network status is. So ToC is set 10 (checksum of primary block and payload block), and the checksum of primary block and payload block will be computed. Then the checksum data field will be filled. At last, the processed packets will queue and wait to be forwarded.

4.2. Processing Rules in Intermediate Nodes

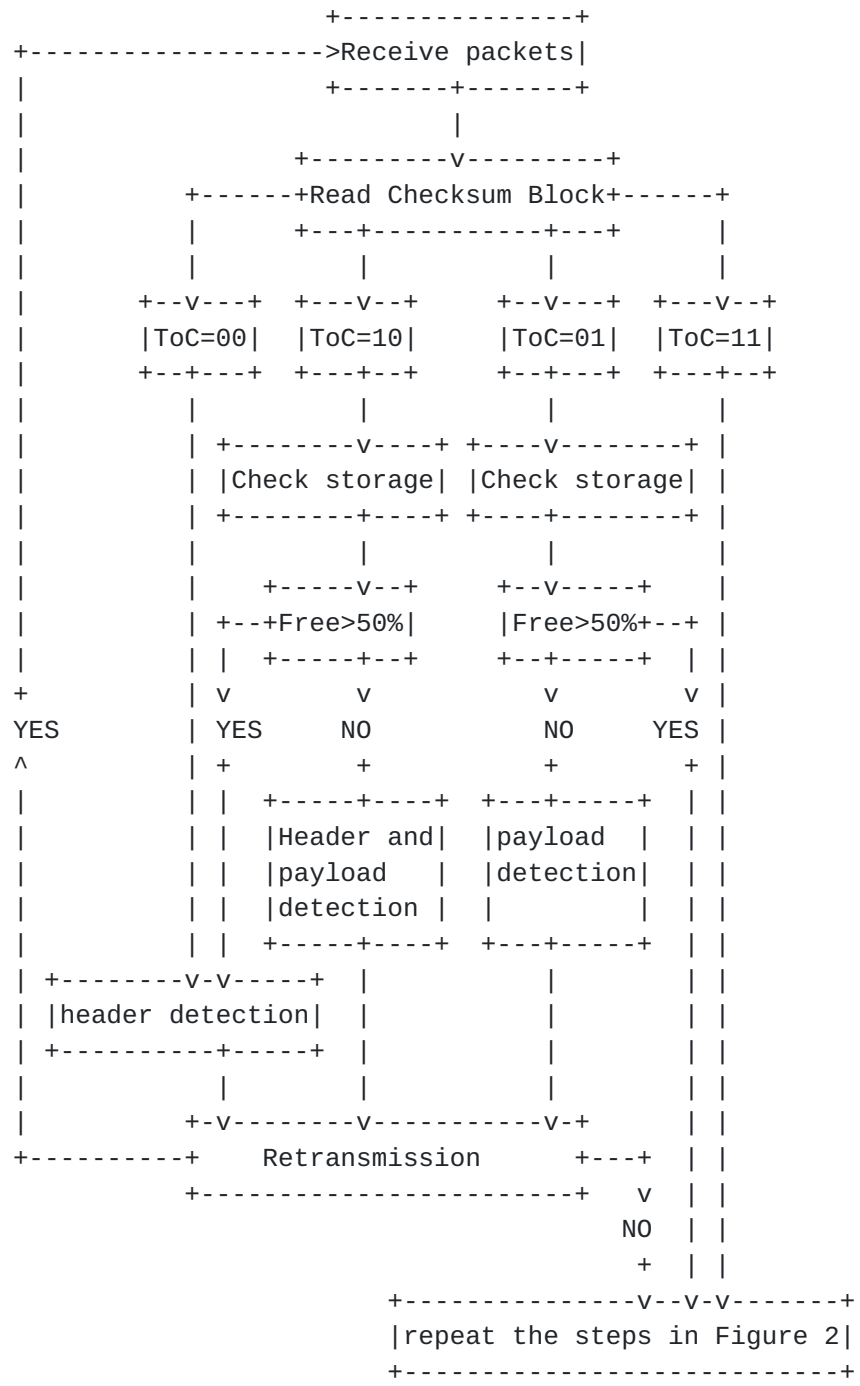

Figure 3 Processing Rules in intermediate nodes

The processing rules in intermediate nodes are shown in Figure 3. When intermediate nodes receive packets, they first read Checksum Block.

If ToC = 00 (checksum of primary block), the primary block (header) will be checked. If ToC = 10(checksum of primary block and payload block), storage space will be detected and if free storage is more than 50%, the primary block will be checked. If free storage is less than 50%, both the primary block and the payload block will be checked. If ToC = 01 (checksum of payload block), storage space will be detected and if free storage is less than 50%, the payload block will be checked. If errors are detected, retransmission will be called. If no errors are detected, or ToC = 11 (no checksum of either primary block or payload block), or free storage is more than 50% when ToC = 01, the following processing steps will be the same as the source nodes in Figure 2. However, if the CoD of received packets is 10, there is no need to collect the network status and calculate new CoD. In other words, the decision of mandatory detection at the source node by the end users should not be modified by the intermediate nodes.

4.3. Processing Rules in Destination Nodes

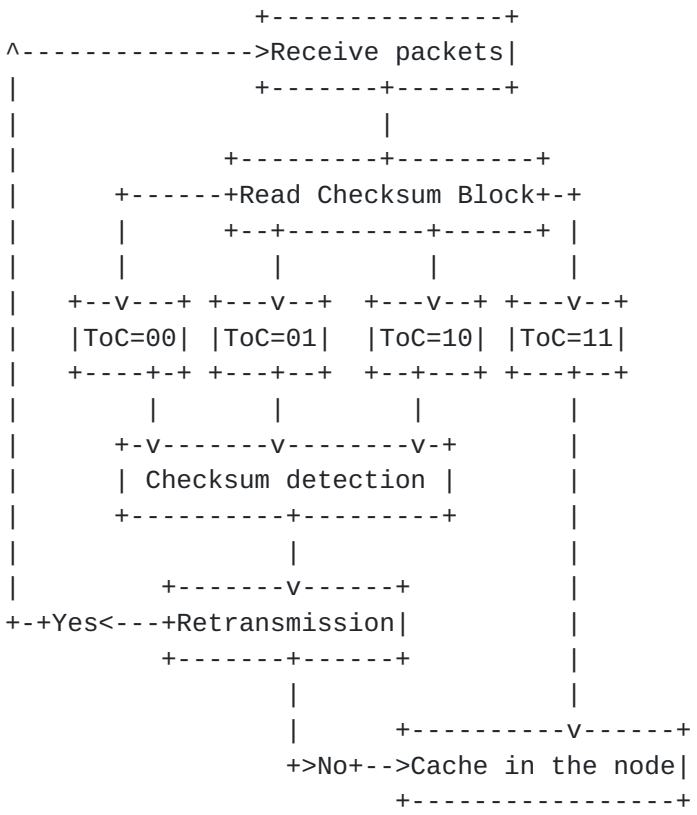

Figure 4 Processing Rules in Destination Nodes

The processing rules in destination nodes are shown in Figure 4. When destination nodes receive packets, they will read the checksum block. If ToC is 00 (checksum of primary block), or 01 (checksum of payload block), or 10 (checksum of primary block and payload block), the related blocks will be checked by a designated algorithm. If errors are detected, retransmission will be called. If no errors are detected, or ToC is 11, the received packets will be regarded as acceptable and be cached and stored in local.

5. Error correcting code-based detecting mechanism

In this section, we introduce a detecting mechanism that based on the idea of error correcting code. This mechanism is designed to reduce the checksum computing times.

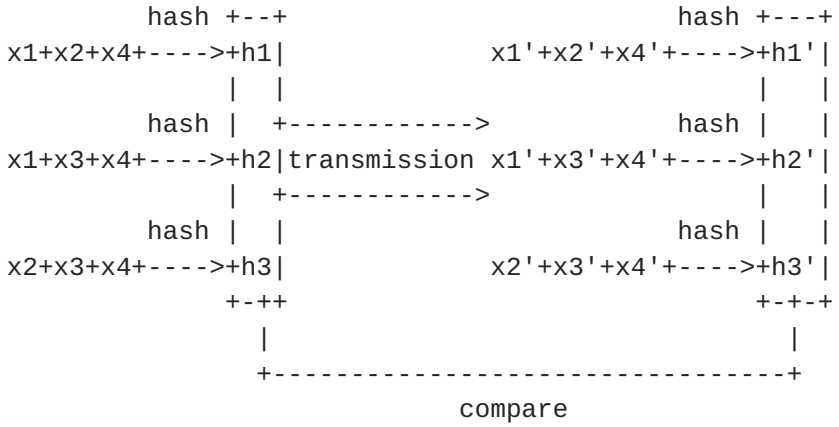

Figure 5 Error Correcting Code-based Detecting Mechanism

We introduce the mechanism taking hamming code as an example. As is shown in Figure 5, we suppose that a file is transmitted in the form of four packets, x1, x2, x3, and x4. The source node and the destination node will compute the checksum for each of the four packet in the traditional way. But in our design, the source node conducts the hash operation for data (x1+x2+x4) and get h1. Then the source node conducts the same operation for data (x1+x3+x4) and (x2+x3+x4), and get h2 and h3. h1, h2 and h3 are carried along with the packets. The destination node just need to conducts hash operation three times, for (x1+x2+x4), (x1+x3+x4) and (x2+x3+x4), and get h1', h2', and h3'. Then the destination node should compare h1, h2 and h3 with h1', h2', and h3'. If h1, h2 and h3 are the same as h1', h2', and h3', then, there are no wrong data in the four packets. If one or several of h1, h2 and h3 is different from h1', h2', and

h3', then the packet that carries wrong data can be located and retransmission will be called to retransmit the wrong packet.

In this way, the checksum computing times can be reduced.

6. Security Considerations

The Multi-strategy Based Payload Integrity Assurance method provides data integrity service for the Bundle Protocol, which is a necessary aspect of security problems.

The proposed method can suit with the Payload Integrity Block (PIB) and Bundle Authentication Block (BAB) in Bundle Security Protocol [RFC6257].

7. IANA Considerations

This specification allocates a codepoint from the "Bundle Block Types" registry defined in [RFC6255].

Additional Entry for the Bundle Block Type Codes Registry:

Value	Description	Reference
20	Checksum Block	This document

Figure 6 Additional Entry for the Bundle Block Type Codes Registry

8. Conclusions

The hybrid integrity assurance strategy proposed in this document describes how to ensure the different levels of integrity of bundles based on different environments.

9. References

9.1. Normative References

[RFC4838]Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and Weiss, H., "Delay-Tolerant Networking Architecture", RFC 4838, April 2007.

[RFC5050] Scott, K., and Burleigh, S., "Bundle Protocol Specification", RFC 5050, RFC5050, November 2007.

[RFC5325] Burleigh, S., Ramadas, M., and Farrell, S., "Licklider Transmission Protocol - Motivation", RFC 5325, September 2008.

[RFC5326] Ramadas, M., Burleigh, S., and Farrell, S., "Licklider Transmission Protocol - Specification", RFC 5326, September 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6258] Symington, S., "Delay-Tolerant Networking Metadata Extension Block", RFC 6258, May 2011.

[RFC6255] Blanchet, M., "Delay-Tolerant Networking (DTN) Bundle Protocol IANA Registries", RFC 6255, May 2011.

[RFC6257] Symington, S., Farrell, S., Weiss, H., Lovell, P., "Bundle Security Protocol Specification ", RFC 6257, May 2011.

9.2. Informative References

[WOOD09] Wood, L., Eddy, W., and Holliday, P., "A Bundle of Problems", Proc. Aerospace conference 2009 pp. 1-17.

[I-D.templin-dtnhiaps-00] Templin, F., "Delay Tolerant Networking Header Integrity Assurance-Problem Statement", draft-templin-dtnhiaps-00 (Expires), March 2014.

[I-D.ietf-dtn-bpsec-06] Birrane, E., "Bundle Protocol Security Specification", draft-ietf-dtn-bpsec-06, October 2017.

10. Acknowledgments

The work in this document was supported by National High Technology of China ("863 program") under Grant No.2015AA015702.

Authors' Addresses

Taixin Li
Beijing Jiaotong University
Beijing, 100044, P.R. China

Email: 14111040@bjtu.edu.cn

Guanwen Li
Beijing Jiaotong University
Beijing 100044, P.R. China

Email: 14120079@bjtu.edu.cn

Huachun Zhou
Beijing Jiaotong University
Beijing 100044, P.R. China

Email: hchzhou@bjtu.edu.cn