Internet Engineering Task Force S. Yang
Internet-Draft X. Huang
Intended status: Informational CUHK(SZ)
Expires: October 26, 2019 R.W. Yeung
CUHK
J.K. Zao
NCTU
April 24, 2019

BATS Coding Scheme for Multi-hop Data Transport
draft-yang-nwcrg-bats-01

Abstract

This document describes a BATS coding scheme for communication through multi-hop networks. BATS code is a class of efficient linear network coding scheme with a matrix generalization of fountain codes as the outer code, and batch-based linear network coding as the inner code.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on October 26, 2019.

Copyright Notice

Copyright (c) 2019 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 (https://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.

Table of Contents

1. Introduction

This document specifies a BATS code [BATS] scheme for data delivery in multi-hop networks. The BATS code described here includes an outer code and an inner code. The outer code is a matrix generalization of fountain codes (see also the RapterQ code described in RFC 6330 [RFC6330]), which inherits the advantages of reliability and efficiency and possesses the extra desirable property of being network coding compatible. The inner code is formed by linear network coding for combating packet loss, improving the multicast efficiency, etc. A detailed design and analysis of BATS codes are provided in the BATS monograph [BATSMonograph].

The BATS coding scheme can be applied in multi-hop networks formed by wireless communication links, which are inherently unreliable due to interference. Existing network protocols like TCP/IP use end-to-end retransmission and store-and-forward at the relays, so that packet loss would accumulate along the way.

The BATS coding scheme can be used for various data delivery applications like file transmission, video streaming, etc.

1.1. Requirements Language

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].

2. Procedures

2.1. Introduction

A BATS coding scheme includes an outer code encoder (also called encoder), an inner code encoder (also called recoder) and a decoder. The BATS coding scheme described in this document can be used by a Data Delivery Protocol (DDP) with the following procedures.

Encoding at a source node which has the data for transmission:

- The DDP provides the data to be delivered and the related information to the BATS encoder.

- The BATS encoder generates a sequence of batches, each consisting of a set of coded packets and the information pertaining to the batch.

- The DDP forms and transmits the DDP packets using the batches and the corresponding batch information.

Recoding at an intermediate node that does not need the data:

- The DDP extracts the batches and the corresponding batch information from its received DDP packets.

- A BATS recoder generates recoded packets of a batch.

- The DDP forms and transmits DDP packets using the recoded packets and the corresponding batch information.

Decoding at a destination node that needs the data:

- The DDP extracts the batches and the corresponding batch information from its received DDP packets.

- A BATS decoder tries to recover the transmitted data using the received batches.

- The DDP sends the decoded data to the application that needs the data.

2.2. Data Partitioning and Padding

Suppose that the DDP has F octets of data for transmission. The construction of source packets from the data depends on two parameters K and T:

o K: the number of source packets.

o T: the size of a source packet, in octets.

If F is smaller than T*K, the data MUST be padded to have T*K octets, so that the data can be partitioned into K source packets, each of which has T octets.

2.3. Data Delivery Procedures

2.3.1. Source Node Procedures

A source node has the data for transmission. The DDP will first pad and partition the data into K source packets, each containing T octets. The DDP provides the BATS encoder with the following information:

o Batch size (M): the number of coded packets in a batch.

o Recoding field size (q): the number of elements in the finite field for recoding.

o The degree distribution (DD).

o A sequence of batch IDs (ID[j], j = 0, 1, ...).

o Number of source packets (K).

o Packet size (T): the number of octets in a source packet.

o The source packets (b[i], i = 0, 1, ..., K-1).

Using this information, the (outer code) encoder generates a batch for each batch ID. For the batch ID ID[j], the encoder returns the DDP that contains

- a sparse degree d[j], and

- M coded packets (x'[j,i], i =0, 1, ..., M-1), each containing T' = T + O octets, where O = ceil(M*log2(q)/8).

The DDP will use the batches to form DDP packets to be transmitted to other network nodes towards the destination nodes. The DDP MUST deliver with each coded packet its

- sparse degree d,

- batch ID and certain extra information so that any receiver of the coded packets of the batch can know whether the coded packets are in the same batch or not, and whether two different batches are generated from the same data source or not.

The DDP MUST deliver the following information to each recoder:

- batch size M, and

- recoding field size q.

The DDP MUST deliver the following information to each decoder:

- batch size M,

- recoding field size q,

- the data size F, and

- the number of source packet K.

The packet length information MUST be known by all the nodes.

2.3.2. Intermediate Node Procedures

An intermediate node does not need the data, but only helps to deliver the data to the destination nodes. At an intermediate node, the DDP only receives the DDP packets from the other network nodes, and should be able to extract coded packets and the corresponding batch information from these packets.

The DDP provides the recoder with the following information:

- the batch size M,
- the recoding field size q,
- a number of received coded packets of the same batch, each containing T' octets, and
- a number M' of recoded packets to be generated.

The recoder uses the information provided by the DDP to generate M' recoded packets, each containing T' octets. The DDP uses the M' recoded packets to form the DDP packets for transmitting.

2.3.3. Destination Node Procedures

A destination node needs the data transmitted by the source node. At the destination node, the DDP receives DDP packets from the other network nodes, and should be able to extract coded packets and the corresponding batch information from these packets.

The DDP provides the decoder with the following information:

- F: number of octets in the data,
- M: batch size,
- q: recoding field size,
- K: number of source packets
- A sequence of batches, each of which is formed by a number of coded packets belonging to the same batch, with their corresponding batch IDs and degrees.

The decoder uses this information to decode the K source packets. If successful, the decoder returns the recovered K source packets to the DDP, which will use the source packets to form the source data.

2.4. Recommendation for the Parameters

The recommendation for the parameters M, K, T, and T' is shown as follows:

- M is 1, 2, 4, 8, 16, 32, 64 and 128.
- q is 2 or 256. When q = 2, M >= 8.
- O = M*log2(q)/8

- T' is not larger than the maximum coded packet payload size.

- T = T' - O.

- K = ceil(F/T).

It is RECOMMENDED that K is at least 128. However, the encoder/decoder SHALL support an arbitrary positive integer value of K.

2.5. Example DDP Packet Format

A DDP can form a DDP packet includes a header and a payload.

2.5.1. Packet Header

The BATS packet header has 40 bits (5 octets) and includes F, M, K, q, ID, and d

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        F      |       K       |  M  |q|         ID            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        d      |
   +-+-+-+-+-+-+-+-+
```

Figure: BATS packet header format.

- F: 8-bit unsigned integer. Value i represents F=2^i.

- K: 8-bit unsigned integer.

- M: 3-bit unsigned integer. The value i represents M=2^i.

- q: 1-bit unsigned integer. The value 0 represents q=1, while the value 1 represents q=256.

- ID:12-bit unsigned integer.

- d: 8-bit unsigned integer.

2.5.2. Packet Payload

```
           +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           |   coefficient vector  |          coded data           |
           +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure: BATS packet payload format.

The playload has T' octets, where the first O octets are called the coeffcient vector and the remaining T octets are the coded data.

- coefficient vector: O octets.

- coded data: T octets.

3. BATS Code Specification

3.1. Background

The T octets of a source packets are treated as a column vector of T elements in GF(256). Linear algebra and matrix operations over finite fields are assumed in this section.

Suppose that a pseudo-random number generator Rand() is given.

3.2. Outer Code Encoder

Let b[0], b[1], ...,b[K-1] be the K source packets. A batch with batch ID ID[j] is generated using the following steps.

First, a degree d[j] = DD() is sampled using the given degree distribution DD() and pseudo-random generator Rand() (initialized with any seed).

Second, initialize Rand() with ID[j] as the seed. Using Rand() sample d packets among all the source packets. Suppose the indices of the packets sampled are i_1, i_2, ..., i_d. Let B[j] = (b[i_1], b[i_2], ..., b[i_d]).

Third, sample a dxM generator matrix G[j] with all the entries uniformly distribution among {0,1,...,255}.

Fourth, form the batch X[j] = B[j]*G[j].

Last, append coefficient vectors to the packets of the batches. Let x[j,i], i=0,1,...,M-1, be the (i+1)th column of X[j]. The coefficient vector of x[j,i] is the (i+1)th column of the MxM identity matrix with entries in GF(q), which can be represented by ceil(M*log2(q)/8) octets. The coded packet x'[j,i] is formed by appending the coefficient vector before x[j,i].

3.3. Inner Code Encoder (Recoder) Recommendations

The inner code comprises (random) linear network coding applied on the coded packets belonging to the same batch. At a particular network node, recoded packets are generated by (random) linear

combinations of the received coded packets of a batch. The recoded packets have the same batch ID, sparse degree and coded packet length.

Suppose that coded packets x'[j,i], i = 0, 1, ..., r-1, which belonging to the same batch j, have been received at an intermediate node. Using the recommended packet format, it can be verified that the corresponding packet header of these coded packets are the same. Then a recoded packet can be generated by one of the following two approaches:

- systematic recoding: use x'[j,i] as a recoded packet.

- linear combination recoding: choose a sequence of coefficients c[i], i = 0, 1, ..., r-1 from GF(q). Generate c[0]x'[j,0]+c[1]x'[j,1]+...+c[r-1]x'[j,r-1] as a recoded packet.

A recoder can combine these two approaches to generate recoded packets. For example, when M' > r, the recoder can output x'[j,i], i = 0, 1, ..., r-1 as r systematic recoded packets and generate M'-r recoded packets using linear combinations of randomly chosen coeffcients.

3.4. Decoder Recommendations

A belief propagation (BP) decoder is described.

The decoder receives a sequence of batches Y'[j], j = 0, 1, ..., n-1, each of which is a T'-row matrix over GF(256). Let Y[j] be the submatrix of the last T rows of Y'[j]. When q = 256, let H[j] be the first M rows of Y'[j]; when q = 2, let H[j] be the matrix over GF(256) formed by embedding each bit in the first M/8 rows of Y'[j] into GF(256).

The decoder is initialized with the following steps similar to those of encoding. For each batch j,

- Initialize Rand() with ID[j] as the seed. Using Rand() sample d[j] packets among all the source packets. Obtain the indices of the packets sampled.

- Sample a dxM generator matrix G[i] with all the entries uniformly distributed among {0,1,...,255}.

A batch j is said to be decodable if the system of linear equations Y[j] = B[j]G[j]H[j] with B[j] as the variable matrix has a unique solution, i.e., rank(G[j]H[j]) = d[j]. The BP decoding algorithm has mutiple iterations. Each iteration is formed by the following steps:

- decoding step: Find all batches j that are decodable. Solve the corresponding system of linear equations Y[j] = B[j]G[j]H[j] and decode B[j].

- substituting step: Substitute the decoded source packets into undecodable batches. Suppose that a decoded source packet b[k] is used in generating a undecodable Y[j]. The subsutiting involves 1) removing the column in B[j] corresponding to b[k], 2) removing the row in G[j] corresponding to b[k], and 3) reducing d[j] by 1.

The BP decoder repeats the above steps until no batches are decodable during the decoding step.

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

Subsuming both Random Linear Network Codes (RLNC) and fountain codes, BATS codes naturally inherit both their desirable capability of offering confidentiality protection as well as their vulnerability towards pollution attacks.

5.1. Provision of Confidentiality Protection

Since the transported messages are linearly combined with random coefficients at each recoding node, it is statistically impossible to recover individual messages by capturing the coded messages at any one or small number of nodes. As long as the coding matrices of the transported messages cannot be fully recovered, any attempt of decoding is equivalent to randomly guessing the transported messages. Thus, with the use of BATS codes, information confidentiality throughout the data transport process is assured.

The only thread towards confidentiality exists in the form of eavesdropping onto the initial encoding process, which takes place at the encoding nodes. In these nodes, the transported data are presented in plain text and can be read along their transfer paths. Hence, information isolation between the encoding process and all other user processes running on the node must be assured.

In addition, the authenticity and trustworthiness of the encoding, recoding and decoding program running on all the nodes must be attested by a trusted authority. Such a measure is also necessary in countering pollution attacks.

5.2. Countermeasures against Pollution Attacks

Like all network codes, BATS codes are vulnerable under pollution attacks. In these attacks, one or more compromised coding node(s) can pollute the coded messages or inject forged messages into the coding network. These attacks can prevent the receivers from recovering the transported data correctly. Although error detection mechanisms can be put in place to prevent the receivers from getting the wrong messages, detection and discard of the polluted messages still constitute a form of denial-of-service (DoS) attack.

The research community has long been investigating the use of various signature schemes (including homomorphic signatures) to identify the forged messages and stall the attacks (see Zhao07 [Zhao07], Yu08 [Yu08], Agrawal09 [Agrawal09]). Nevertheless, these countermeasures are regarded as being computationally too expensive to be employed in broadband communications. A practical approach to protect against pollution attacks consists of the following system-level countermeasures:

1. Attestation and Validation of all encoding, recoding and decoding nodes in the network: Remote attestation and repetitive validation of a node based on valid public key certificates with proper authorization MUST be a pre-requisite of admitting that node into a network and permitting it to remain in that network.

2. Attestation of all encoding, recoding and decoding programs used in the coding nodes: All programs used to perform the BATS encoding, recoding and decoding processes MUST be remotely attested before they are permitted to run on any of the coding nodes. Reloading or alteration of programs MUST NOT be permitted during an encoding session. Programs MUST be attested or validated again when they are executed in new execution environments instantiated even in the same node.

3. Packet Integrity and Original Authentication protection SHOULD be provided to all coded messages communicating among the encoding, re-coding and decoding nodes using network or transport level security protocols. If the coded communication occurs at the network layer, the IPsec ESP protocol in Transport Mode RFC 4303 [RFC4303] employing the HMAC-SHA2-256-128 function RFC 4868 [RFC4868] SHOULD be used. Preferably, the HMAC-SHA2-512-256 function MAY be used to provide the necessary protection RFC 8221 [RFC8221]. If the communication occurs at the transport layer then the DTLS protocol employing the same integrity and origin authentication functions SHOULD be used to provide the protection RFC 6347 [RFC6347].

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

6.2. Informative References

[Agrawal09] Agrawal, S. and D. Boneh, "Homomorphic MACs: MAC-based integrity for network coding", International Conference on Applied Cryptography and Network Security , 2009.

[BATS] Yang, S. and R. W. Yeung, "Batched Sparse Codes", IEEE Transactions on Information Theory 60(9), 5322-5346, 2014.

[BATSMonograph] Yang, S. and R. W. Yeung, "BATS Codes: Theory and Practice", Morgan & Claypool Publishers , 2017.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <https://www.rfc-editor.org/info/rfc4868>.

[RFC6330] Luby, M., Shokrollahi, A., Watson, M., Stockhammer, T., and L. Minder, "RaptorQ Forward Error Correction Scheme for Object Delivery", RFC 6330, DOI 10.17487/RFC6330, August 2011, <https://www.rfc-editor.org/info/rfc6330>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC8221] Wouters, P., Migault, D., Mattsson, J., Nir, Y., and T. Kivinen, "Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 8221, DOI 10.17487/RFC8221, October 2017, <https://www.rfc-editor.org/info/rfc8221>.

[Yu08] Yu, Z., Wei, Y., Ramkumar, B., and Y. Guan, "An Efficient Signature-Based Scheme for Securing Network Coding Against Pollution Attacks", NFOCOM , 2008.

[Zhao07] Zhao, F., Kalker, T., M.Meard, and K. Han, "Signatures for content distribution with network coding", ISIT , 2007.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

Shenghao Yang
CUHK(SZ)
Shenzhen, Guangdong
China

Phone: +86 755 8427 3827
Email: shyang@cuhk.edu.cn

Xuan Huang
CUHK(SZ)
Shenzhen, Guangdong
China

Phone: +86 755 8427 3814
Email: 115010159@link.cuhk.edu.cn

Raymond W. Yeung
CUHK
Hong Kong, Hong Kong SAR
China

Phone: +852 3943 8375
Email: whyeung@ie.cuhk.edu.hk

John K. Zao
NCTU
Hsinchu, Taiwan
China

Email: jkzao@ieee.org