

Article

A Design of Overlapped Chunked Code over Compute-and-Forward for Multi-Source Multi-Relay Networks[†]

Rithea Ngeth * , Brian M. Kurkoski, Yuto Lim and Yasuo Tan

School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Ishikawa 923-1292, Japan; kurkoski@jaist.ac.jp (B.M.K.); ylim@jaist.ac.jp (Y.L.); ytan@jaist.ac.jp (Y.T.)

* Correspondence: ngethritha@jaist.ac.jp

† This paper is an extended version of our paper: Ngeth, R.; Lim, Y.; Kurkoski, B.M.; Tan, Y. A Design of Overlapped Chunked Code over Compute-and-Forward in Multi-Source Multi-Relay Networks. In Proceedings of 2018 IEEE Global Communications Conference: Wireless Communications, Abu Dhabi, UAE, 9–13 December 2018.

Received: 2 August 2018; Accepted: 21 September 2018; Published: 25 September 2018



Abstract: This paper investigates the design of overlapped chunked codes (OCC) for multi-source multi-relay networks where a physical-layer network coding approach, compute-and-forward (CF) based on nested lattice codes (NLC), is applied for the simultaneous transmissions from the sources to the relays. This code is called OCC/CF. In this paper, OCC is applied before NLC before transmitting for each source. Random linear network coding is applied within each chunk. A decodability condition to design OCC/CF is provided. In addition, an OCC with a contiguously overlapping, but non-rounded-end fashion is employed for the design, which is done by using the probability distributions of the number of innovative codeword combinations and the probability distribution of the participation factor of each source to the codeword combinations received for a chunk transmission. An estimation is done to select an allocation, i.e., the number of innovative blocks per chunk and the number of blocks taken from the previous chunk for all sources, that is expected to provide the desired performance. From the numerical results, the design overhead of OCC/CF is low when the probability distribution of the participation factor of each source is dense at the chunk size for each source.

Keywords: overlapped chunked code; compute-and-forward; nested lattice code; multi-source multi-relay; empirical rank distribution; decodability

1. Introduction

This paper is an extended version of the work in [1].

In the current situation, wireless network nodes are ubiquitous and have increasing density. Since the wireless channel bandwidth is limited, the interference between nodes can affect the data transmission between nodes, e.g., causing message loss, longer latency, high energy consumption, etc. In order to solve this issue, the handshaking mechanism is applied to share the channel between nodes via a control message, e.g., request-to-send and clear-to-send. In addition, the transmit power control approach can allow multiple sources to transmit their messages simultaneously with appropriate interference between nodes. On the other hand, some proposed solutions exploited the interference instead of dealing with (or compensating) it. One of these solutions is physical-layer network coding (PNC) [2], which can also allow multiple sources to transmit their messages to the common receivers simultaneously.

In the packet switching network, a message file is divided into small packets or blocks. Network coding is a scheme where the outgoing block of the sender is a function of incoming blocks. By considering the form of data, network coding can be divided into straightforward network coding (SNC) and PNC. Straightforward network coding works with binary or with a symbol in the finite field, and PNC works with signal forms, i.e., with real values. Exclusive or (XOR) is a simple method of SNC. It was applied in a two-way relay channel with inter-flow network coding approach to improve the network throughput and also the energy efficiency [3]. Linear network coding is a general approach of XOR. It linearly combines the incoming blocks in a finite field of size q , \mathbb{F}_q ($q = 2$ for XOR). The combined coefficients are randomly drawn from \mathbb{F}_q for random linear network coding (RLNC). The application of RLNC was widely studied in wireless multi-hop networks for reliable communication [4,5].

PNC, sometimes called analog network coding, performs additive mixing of received electromagnetic waves (received signals) from different sources naturally at a receiver, then transforms and maps the superimposed received signals into a desired function of transmitted source signals. The time to complete the message transmission can be shorter with PNC if comparing with SNC for example in a two-way relay channel [2], cross atom topology [6], because PNC allows multiple sources to transmit their data simultaneously, i.e., via a non-orthogonal channel. Nazer and Gastpar [7] proposed a PNC method, compute-and-forward (CF) based on nested lattice codes (NLC), for multi-source multi-relay channels. All sources encode their blocks of message into NLC codewords and simultaneously transmit them to the relays. Each relay computes the superimposed codeword signals to obtain the linear combination of the codewords of all sources with a combination integer coefficient vector, then forwards the codeword combination to the destination. The destination can recover the original blocks of all sources from the codeword combinations forwarded from the relays if it receives enough linearly independent (i.e., innovative) codeword combinations. The cooperation between relays can provide the opportunity for the destination to collect enough linearly independent codeword combinations with desired purposes such as obtaining the highest sum rate [8,9] or the highest throughput [10] while the sources also take part.

However, in some cases, codeword combinations are not qualified to be forwarded to the destination, i.e., considered as unsuccessfully received, and the codeword combinations at the different relays might be linearly dependent on each other since all relays compute the superimposed signals independently. Retransmission and feedback sending back to the sources will be needed if the relays cannot provide enough linearly independent codeword combinations to the destination to recover the original blocks of all sources. However, feedback might be lost, has some delay in reaching the sources and causes some energy consumption. These shortcomings are called protocol overhead in this paper.

In the traditional communication network, when a block or a packet is correctly received by the destination, a feedback, i.e., an acknowledgment (ACK), is sent back to the source to manage the next transmission. Negative ACK (NACK) is used to inform about the unsuccessful reception of a sending block. Block ACK (BACK) can be used to reduce the needed ACKs as transmitting a certain number of blocks, and a BACK containing the reception states of the sending blocks is sent back to the source. However, if feedback is lost or its transmission time is significant, (e.g., the transmissions between ground stations and satellite), then the protocol overhead can affect the end-to-end network performance, especially in the lossy multi-hop communication multi-source networks because feedback needs to be forwarded via the intermediate nodes to the sources.

On the other hand, RLNC can be applied to reduce the protocol overhead because an ACK is needed when the receiver can decode the received coded blocks. However, if the number of input blocks is large, then the encoding and decoding computational complexity, which depends on the number of input blocks, especially decoding complexity, will be high and not practical. The large number of blocks is grouped into disjoint generations or chunks [11,12], then the computational complexity can be reduced. Nevertheless, when the number of blocks per chunk, i.e., size of the chunk, is too small, the protocol overhead is still significant. If the message success rate (MSR) of the transmission link is known, then the sources can transmit each chunk with an expected number

of coded blocks, and feedback can be avoided [4]. However, if MSR is not constant, then there will be some chunks that are not decodable, i.e., undecodable. To deal with this problem, the works in [13,14] proposed overlapped chunk code (OCC), where a block can belong to more than one chunk. A decoded chunk can be used to help to decode the other undecodable chunks by back-substitution, i.e., blocks from decoded chunks are substituted into the undecodable chunks that also have them as input blocks. The other designs of OCCs and those of the codes similar to OCC were proposed then such as in the works of [15–18]. These designed codes are mainly for single flow transmission or multicast transmission, i.e., the transmission of a source data. Up to the present, there is no design of OCC for the data transmission in multi-source multi-relay networks.

This paper considers the design and the application of OCC for the data transmissions in multi-source multi-relays networks where CF based on NLC is employed. The designed OCC is denoted as OCC/CF in this paper. The aim is to investigate the advantage of OCC/CF over a feedback-based transmission scheme. In addition, low computational complexity is considered such that the proposed work is applicable to low specification wireless nodes, e.g., wireless sensor nodes. This paper considers varying channel states where only receivers have knowledge of channel coefficients. The blocks of each source message are grouped into chunks. RLNC is done within each chunk before encoding with NLC. Only the transmissions from the sources to the relays are considered. The challenge to apply OCC in a multi-source multi-relay network is how to design OCC/CF such that the decodability of each chunk of all sources at the destination is ensured or the desired network performance is achieved. The contributions of this paper are as follows:

- analyzing the decodability for chunks received at the destination to design OCC for each source and providing a decodability condition to design OCC/CF;
- based on the condition of decodability, designing OCC/CF by employing an OCC with a contiguously overlapping, but non-rounded-end fashion at each source. The design is done by using the empirical rank distribution, i.e., the probability distribution of the number of linearly independent codeword combinations received at the destination per chunk transmission, as in the work of [17,18], and by using the probability distribution of the participation factor of each source to the received codeword combinations per chunk transmission. These two keys depend on the channel states from the sources to the relays, and they are applicable for any channel distribution;
- providing a decoding scheme based on the feature of the employed OCC. The decoding scheme considers the other opportunity of starting decoding besides back-substitution, the combination of chunks. The decoding complexity is bounded by the maximum number of combined chunks, and the storing overhead can be reduced;
- estimating the performance of the designed OCC/CF by following the decoding scheme and using table lookup for all allocations, i.e., the number of innovative blocks per chunk and the number of contiguously overlapped blocks for each source. The estimation is to determine which allocation can provide the desired performance such as high decodability, highest channel efficiency and acceptable decoding complexity;
- reducing the number of candidates for the linear combination coefficient vector computed at each relay. This is achieved by a trade-off between computational latency and the performance in the frame error rate.

The numerical results demonstrate that the design of OCC/CF not only depends on the empirical rank distribution, but also on the probability distribution of the participation factor of each source. The chance to improve the network performance by employing OCC/CF depends on the feedback latency and feedback reception success rate if comparing with a feedback-based CF transmission scheme.

The remainder of this paper is organized as follows. Related works are described in Section 2. A short review of NLC and CF is introduced in Section 3. Section 4 describes the system model of this paper work, which includes the scenario, channel model, encoding scheme at sources and computing at relays, acquiring the linear combination coefficient vector at each, the considered empirical rank

distributions and the analysis of decodability. Section 5 talks about the design of OCC/CF by using an OCC and the applied decoding scheme. The estimation of decodability for the OCC/CF designed is given in Section 6. The performance analysis and the reference schemes are described in Section 7. Section 8 shows the numerical results and discussion. At the end, Section 9 gives the conclusion.

2. Related Works

To complete the message transmission without the need for feedback, the code to be mentioned would be rateless code where the number of coded blocks is unlimited and the transmitter keeps sending the coded blocks until the receiver can recover all original blocks or packets. Fountain code [19] is an erasure code and a rateless code. The feature of fountain code is low computational complexity in encoding and decoding processes since they are done in the binary field, i.e., \mathbb{F}_2 . This includes LTcode [20], Raptor code [21] and online code [22]. The decodability for LT code depends on the degree distribution, which is determined based on the soliton distribution. Degree is the number of input blocks to generate a coded block. The input blocks for each coded block are randomly selected. Raptor code applied the precoding process before encoding such that while a fraction of coded blocks are received, then all original blocks are recoverable. Online codes applied a precoding process for the distributed networks. The decoding process, while employing fountain codes, starts when at least a one-degree coded block, i.e., plain block, exists and stops when there are no more one-degree coded blocks. The decoded blocks are back-substituted into the new received coded blocks, which also have them as input blocks. The application of the inactivation decoding method [23] was studied in the work [24] for the decoding process of LT code and Raptor code to reduce the decoding complexity because the transfer matrix of the received coded blocks, i.e., the coding coefficient matrix of the received coded blocks, is a sparse matrix.

For RLNC, each element of the coding coefficient matrix of the sending coded blocks are randomly drawn from a finite field \mathbb{F}_q (normally, q is enough large, e.g., $q = 2^8$). The linear independence between coded blocks with RLNC is higher than with sparse network coding (the generated coding coefficient matrix of the coded blocks is a sparse matrix) especially in lossy communication networks, but the computational complexity of RLNC is higher. RLNC was employed within each chunk for OCC proposed in the work of [13] where two overlapping fashion were given: rectangular grid code and diagonal grid code. The number of chunks is finite, but the decodability of received chunks was not clearly analyzed. The overlapping fashion of OCC in the work of [14] is contiguous and in a rounded-end fashion. The decodability is analyzed with chunk size, the number of contiguously overlapped blocks and the number of received coded blocks. However, achieving high decodability, i.e., the probability that a chunk is decodable, requires a large chunk size, which can make the computational complexity more significant. A small sized chunk was analyzed then in their later work [15]. However, the decoding process would start when the receiver has collected a sufficient number of coded blocks of all chunks in the worst case, i.e., when there are no more decodable chunks. Then, higher storing ability at the receiver would be required, and the decoding complexity is still significant. The design of OCC with the other overlapping fashion was proposed in the work of [16], where the overlapped blocks, i.e., the blocks taken from the other chunks, are randomly selected. Although the performance in decodability is better than OCC with the contiguously overlapping fashion [14], the decoding process still might start when a sufficient number coded blocks are received.

Batched sparse (BATs) codes proposed in the work of [17] inherit the feature of rateless code by employing fountain codes as the outer code (chunk size obeys a degree distribution) and random linear network code as the inner code (RLNC is employed within each chunk). The degree distribution is determined using the empirical rank distribution to obtain the optimal performance in achievable rate. The decoding process starts when there is at least a decodable chunk, and back-substitution is done then. The inactivation decoding method might be applied when there are no more decodable chunks. The other design, which also employs the empirical rank distribution, is in the work of [18], where

chunk size is fixed. Two degree distributions are defined, and a degree distribution is determined when another degree distribution is fixed to obtain the optimal achievable rate.

This paper provides the design of OCC/CF with a condition of decodability, which can be applied with the designs of codes for single flow transmission, which are described above. This paper employs an OCC in a contiguously overlapping fashion to design OCC/CF because it is simpler to determine which allocation for each source to obtain the desired performance since there are only two variables to be determined for each source. Although its performance in rate (channel efficiency, for this paper) is not higher than the other designs in single flow transmission, it has a potential to reduce the storage overhead and the computational complexity to suit its application with a low specification wireless node in multi-source multi-relay networks.

3. Preliminaries

3.1. Notation

Boldface letters are used for vectors, e.g., \mathbf{a} . The capital boldface letters are for matrices, e.g., \mathbf{G} . Superscripts T and $^{-1}$ refer to matrix transposition operation and inverse operation, respectively. \mathbb{R} and \mathbb{Z} denote the field of real values and the field of integer values, respectively. In addition, $\text{sign} \cdot$ refers to the multiplication operation, and $\text{sign} \times$ is used to express the size of the matrix.

3.2. Nested Lattice Codes

An n -dimensional lattice Λ is a linear additive subgroup of \mathbb{R}^n , i.e., if $\mathbf{x}_1, \mathbf{x}_2 \in \Lambda$, then $\mathbf{x}_1 + \mathbf{x}_2 \in \Lambda$ and $-\mathbf{x}_1 \in \Lambda$. A lattice point $\mathbf{x} \in \Lambda$ is generated by the generator matrix $\mathbf{G} \in \mathbb{R}^{n \times n}$ and an integer vector $\mathbf{b} \in \mathbb{Z}^n$ by:

$$\mathbf{x} = \mathbf{G} \cdot \mathbf{b}. \quad (1)$$

The fundamental Voronoi region of Λ , \mathcal{V} , is the space that is closer to the origin \mathbf{x}_o ($\mathbf{x}_o = \mathbf{0}$) than to the other lattice points. A scaled lattice $\Lambda_p = p \cdot \Lambda$ is obtained by scaling the generator matrix of Λ , i.e., $\mathbf{G}_p = p \cdot \mathbf{G}$. A lattice Λ_p is nested in Λ if $\Lambda_p \subseteq \Lambda$. If p is a non-zero positive integer, then $\Lambda_p = p \cdot \Lambda$ is nested in Λ .

NLC is formed by a coding lattice Λ_c and a shaping lattice Λ_s , where $\Lambda_s \subseteq \Lambda_c$. The codebook of NLC is the coset leaders of Λ_c/Λ_s , i.e., the lattice points (codewords) of Λ_c that are inside the fundamental Voronoi region of Λ_s , \mathcal{V}_s . If taking $\Lambda_s = q\Lambda_c$, where q is a prime number, and the generator matrix of Λ_c , \mathbf{G}_c , is full rank, then the coding rate of NLC is $R = \log_2 q$. The number of codewords is q^n . The feature of NLC is that the linear combination of two codewords is still a codeword. The encoding process of NLC can be done as below.

$$\mathbf{x} = [\mathbf{G}_c \cdot \mathbf{b}] \text{ mod } \mathcal{V}_s, \quad (2)$$

where $\mathbf{b} \in \mathbb{F}_q^n$ is the information, \mathbf{x} is the NLC codeword corresponding to \mathbf{b} and $[\] \text{ mod } \mathcal{V}_s$ is the operation mapping a lattice point of Λ_c into \mathcal{V}_s . This operation restricts the transmit power of a sending codeword by an assigned maximum transmit power P_{\max} .

The decoding process can be done as below:

$$\mathbf{b} = [\mathbf{G}_c^{-1} \cdot \mathbf{x}] \text{ mod } q, \quad (3)$$

where $[\] \text{ mod } q$ is the modulo operation by q or the operation mapping an integer value into the finite field \mathbb{F}_q .

3.3. Compute-and-Forward

If K sources transmit their codeword simultaneously, the accumulative codeword at relay l can be expressed by:

$$\mathbf{y}_l = \sum_{k=1}^K h_{kl} \cdot \mathbf{x}_k + \mathbf{z}_l, \tag{4}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is an n -dimensional NLC codeword, which is transmitted from source k for $k \in \{1, 2, \dots, K\}$. $h_{kl} \in \mathbb{R}$ is a real channel coefficient of the link from source k to relay l . For the case of a complex channel coefficient, the derivation can be done as in the work of [7]. On the other hand, $\mathbf{z}_l \in \mathbb{R}^n$ is additive white Gaussian noise (AWGN).

Relay l computes \mathbf{y}_l to obtain a linear combination of the codewords of K sources, \mathbf{v}_l , where:

$$\mathbf{v}_l = \sum_{k=1}^K a_{kl} \cdot \mathbf{x}_k, \tag{5}$$

where $\mathbf{a}_l = [a_{1l}, a_{2l}, \dots, a_{Kl}]^T \in \mathbb{Z}^K$ is a linear combination integer coefficient vector used at relay l and a_{kl} is called the k -th element of \mathbf{a}_l . \mathbf{v}_l is mapped into \mathcal{V}_s to obtain \mathbf{u}_l before forwarding to obey the transmit power constraint, i.e., $\frac{1}{n} \sum_{n'=1}^n u_{ln'}^2 \leq P_{\max}$, where $u_{ln'}$ is the n' -th element of \mathbf{u}_l , and $P_{\max} > 0$.

$$\mathbf{u}_l = [\mathbf{v}_l] \text{ mod } \mathcal{V}_s = \left[\sum_{k=1}^K \beta_{kl} \cdot \mathbf{x}_k \right] \text{ mod } \mathcal{V}_s, \tag{6}$$

where $\beta_l = [\beta_{1l}, \beta_{2l}, \dots, \beta_{Kl}]^T \in \mathbb{F}_q^K$ and $\beta_l = [\mathbf{a}_l] \text{ mod } q$. β_l is a combination coefficient vector for codeword combination \mathbf{u}_l .

4. System Model

4.1. Scenario

This paper takes a scenario of a K sources L relays single-destination network as shown in Figure 1, which is in the case $K = L = 2$. Each node is equipped with a single antenna. The direct links from the sources to the destination are not considered, and only the transmissions from the sources to the relays are considered. In this paper, all sources apply the NLC with the same coding rate $R = \log_2 q$. This paper assumes that the transmissions from the relays to the destination are lossless. The process of forwarding the codeword combinations to the destination can be done as in the works of [8,9] by exploiting the coordination from the destination via control messages between the relays and the destination to select which codeword combinations are to be forwarded and which relays to forward.

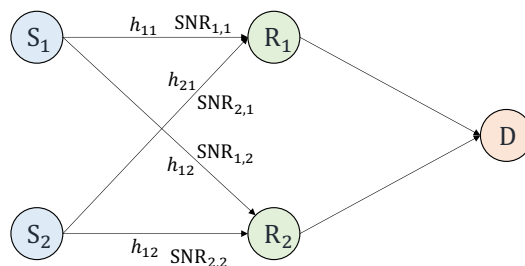


Figure 1. Scenario for the case of a two-source two-relay single-destination network. h_{kl} is the channel coefficient corresponding to the instantaneous received signal-to-noise ratio (SNR) of the link from source k to relay l , where $k \in \{1, 2, \dots, K\}$ and $l \in \{1, 2, \dots, L\}$. SNR_{kl} denotes the average received SNR of the link from source k to relay l .

This scenario is considered as data collection in wireless sensor networks or data backhauling in ultra-dense networks, and it is a part of the topologies of these networks. On the other hand, if its application in cognitive radio (CR) network is considered, the primary user (PU) is one of K sources, and the other sources are secondary users (SUs). Alternatively, all sources can be assigned as SUs.

There is a relay assigned for each source if the transmissions (used for reference schemes) via the orthogonal channel are considered.

4.2. Channel Model

This paper assumes that time is slotted and synchronized. Only real channel coefficients are considered, and the block channel fading is assumed, i.e., the channel coefficient for a whole block signal within a time slot along a channel link is constant. In addition, Rayleigh fading is considered, and the channel coefficient is independently and identically distributed for each channel link. Hence, the real channel coefficient is normally distributed. The average received signal-to-noise ratio for the link from source k to relay l is denoted by SNR_{kl} . On the other hand, AWGN has zero mean and unit variance in this paper.

4.3. Computing Combination Coefficient Vector

$\mathcal{R}_l(\mathbf{h}_l, \mathbf{a}_l)$ is defined as the computation rate region corresponding to the channel coefficient vector $\mathbf{h}_l = [h_{1l}, h_{2l}, \dots, h_{kl}]^T$ and correspondent \mathbf{a}_l . According to the work in [7], \mathcal{R}_l is achievable for any large enough n and for the existing encoders and decoders such that the receiver can recover the desired codeword combination with $\mathbf{a}_l \neq \mathbf{0}$ with the average probability of error $\epsilon > 0$ if the maximum coding rate of all sources, i.e., R for this paper, satisfies the condition:

$$\mathcal{R}_l(\mathbf{h}_l, \mathbf{a}_l) > R. \quad (7)$$

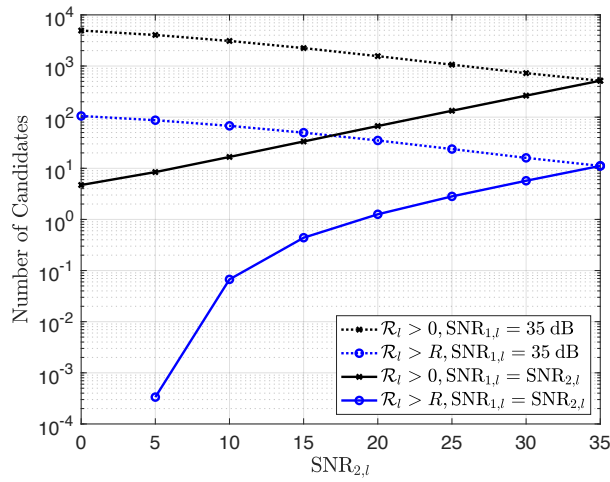
In this paper, \mathbf{a}_l is determined by applying the method proposed by U. Fincke and M. Pohst [25] as in the work of [26] to obtain the highest $\mathcal{R}_l(\mathbf{h}_l, \mathbf{a}_l)$. By considering the hardware specification of sensor nodes, this paper exploits Condition (7) to reduce the computational overhead by reducing the number of candidates of \mathbf{a}_l in searching, for which \mathbf{a}_l can provide the highest $\mathcal{R}_l(\mathbf{h}_l, \mathbf{a}_l)$. In addition, Condition (7) is also used to filter codeword combination for forwarding to the destination at each relay. Since $R = \log_2 q$, the higher value q results in a high message loss rate. In this paper, only a small value of q is considered. Reducing the number of candidates of \mathbf{a}_l , i.e., reducing the bounds of the value of the elements of \mathbf{a}_l , can be done as in the works of [26,27] by replacing the condition $\mathcal{R}_l(\mathbf{h}_l, \mathbf{a}_l) > 0$ with $\mathcal{R}_l(\mathbf{h}_l, \mathbf{a}_l) > R$.

However, this modification causes some decrease in performance in the block error rate or frame error rate (FER) because codeword combination might be correctly received without satisfying Condition (7). By comparing with the case that applies condition $\mathcal{R}_l(\mathbf{h}_l, \mathbf{a}_l) > 0$, the number of candidates, FER and computational latency are shown in Figure 2. The specification of the employed platform is shown in Table 1. The lattice code $E_8/7E_8$ is used for NLC in this comparison, where $q = 7$, and E_8 is a well-known $n = 8$ lattice.

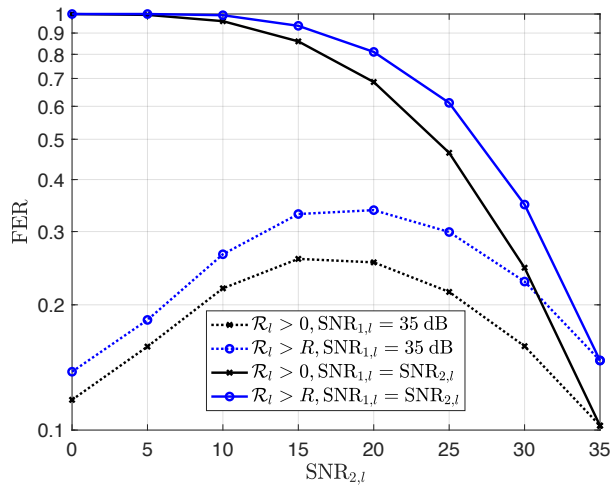
The result is obtained by considering the codeword combinations of two sources at relay l and taking $\text{SNR}_{2,1} = 0 \rightarrow 35$ dB and $\text{SNR}_{1,l}$ with two cases: $\text{SNR}_{1,l} = 35$ dB and $\text{SNR}_{1,l} = \text{SNR}_{2,l} = 0 \rightarrow 35$ dB. The FER for condition $\mathcal{R}_l(\mathbf{h}_l, \mathbf{a}_l) > 0$ was obtained by comparing the codeword combination with the combination of the original codewords. For the case with Condition (7), the codeword combination is filtered with Condition (7) first before comparing with the combination of the original codewords. From Figure 2, this setting performs the trade-off between the computational latency and the FER performance.

Table 1. Specification of the simulation platform.

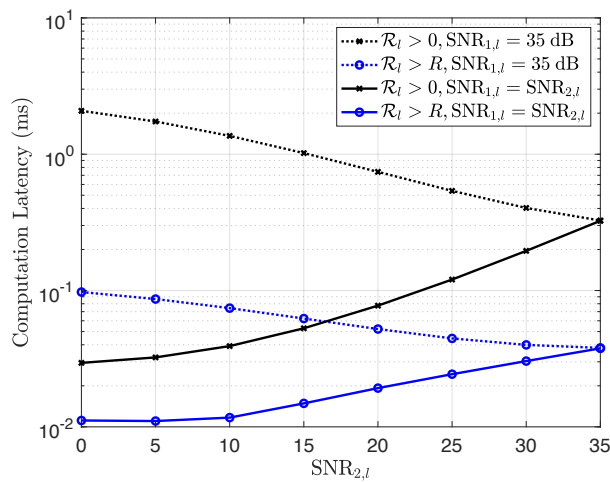
Term	Description
Processor	2.5-GHz Intel Core i7
Memory	16-GB 1600-MHz DDR3
Operating System	Mac OS
Software Tool	MATLAB



(a)



(b)



(c)

Figure 2. The performance applying condition $\mathcal{R}_l(\mathbf{h}_l, \mathbf{a}_l) > R$, compared with condition $\mathcal{R}_l(\mathbf{h}_l, \mathbf{a}_l) > 0$. (a) Number of candidates; (b) frame error rate (FER); (c) computational latency.

4.4. Encoding and Computing

A big file message is divided into small blocks, and blocks are selected to group into chunks or batches. For source k where $k \in \{1, 2, \dots, K\}$, the i -th chunk consists of $d_k^{(i)}$ blocks and is expressed by $\mathbf{B}_k^{(i)} = [\mathbf{b}_{k1}^{(i)}, \mathbf{b}_{k2}^{(i)}, \dots, \mathbf{b}_{kd_k^{(i)}}^{(i)}]$. $D^{(i)}$ and d_{\max} denote $\sum_{k=1}^K d_k^{(i)}$ and $\max\{d_k^{(i)}, 1 \leq k \leq K, \forall i\}$, respectively. RLNC is applied among chunks to generate M coded blocks, $\mathbf{W}_k^{(i)} = [\mathbf{w}_{k1}^{(i)}, \mathbf{w}_{k2}^{(i)}, \dots, \mathbf{w}_{kM}^{(i)}]$. The m -th coded block, $\mathbf{w}_{km}^{(i)}$, is obtained by:

$$\mathbf{w}_{km}^{(i)} = \left[\sum_{d=1}^{d_k^{(i)}} \chi_{kmd}^{(i)} \cdot \mathbf{b}_{kd}^{(i)} \right] \bmod q, \tag{8}$$

where $\chi_{km}^{(i)} = [\chi_{km1}^{(i)}, \chi_{km2}^{(i)}, \dots, \chi_{kmd_k^{(i)}}^{(i)}]^T$ is randomly drawn from $\mathbb{F}_q^{d_k^{(i)}}$. It is the coding vector of coded block $\mathbf{w}_{km}^{(i)}$. Superscript (i) is sometimes omitted here for convenience. The computational complexity of the encoding process depends on chunk size $d_k^{(i)}$.

The coded blocks of each chunk are then NLC encoded before transmitting to generate M NLC codewords, $\mathbf{X}_k^{(i)} = [\mathbf{x}_{k1}^{(i)}, \mathbf{x}_{k2}^{(i)}, \dots, \mathbf{x}_{kM}^{(i)}]$, as shown in Figure 3. All sources transmit these M codewords for each chunk simultaneously to the relays.

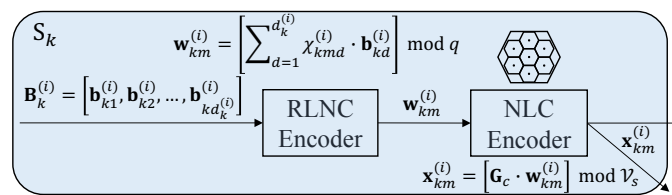


Figure 3. Encoding process at source k for chunk i . RLNC, random linear network coding.

The coding vector $\chi_{km}^{(i)}$ and the information of chunk i for source k , such as source ID (k), $d_k^{(i)}$, etc., can be attached to the transmitting data, e.g., at the header of the frame. However, the location of the attached information for a source should not overlap with those of the other sources, as shown in Figure 4. Hence, the small chunk size is preferred for the header with a limited length. This paper assumes that the length of the attached information is negligible compared with the length of the sending block. Alternatively, this information can be known by the receivers (relays or destination) by broadcasting from each source, for example. This paper assumes that the content of this information is correctly received.

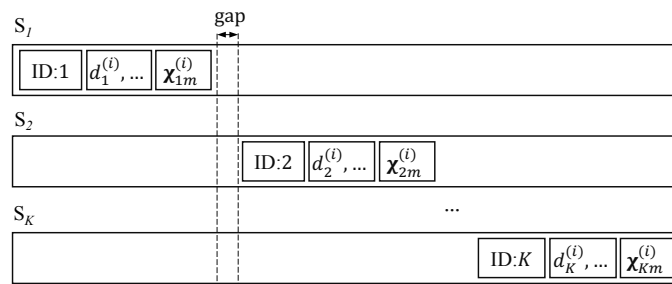


Figure 4. Locations of the attached information in the header of the sending frame of all sources for chunk i , where “ $d_k^{(i)}, \dots$ ” refers to the information about chunk i for source k such as chunk size $d_k^{(i)}$, information about overlapped blocks, etc.

Relay l for $l \in \{1, 2, \dots, L\}$ computes the superposition of K codewords $[\mathbf{x}_{1m}^{(i)}, \mathbf{x}_{2m}^{(i)}, \dots, \mathbf{x}_{Km}^{(i)}]$ to obtain their linear combination $\mathbf{u}_{lm}^{(i)} \in \mathbb{R}^n$ to forward to the destination, which is:

$$\mathbf{u}_{lm}^{(i)} = \left[\sum_{k=1}^K \beta_{klm}^{(i)} \cdot \mathbf{x}_{km}^{(i)} \right] \text{ mod } \mathcal{V}_s, \quad (9)$$

where $\boldsymbol{\beta}_{lm}^{(i)} = [\beta_{1lm}^{(i)}, \beta_{2lm}^{(i)}, \dots, \beta_{Klm}^{(i)}]^T \in \mathbb{F}_q^K$ is the combination coefficient vector computed at relay l for the m -th blocks of all sources for chunk i . Then, the combined coding coefficient vector of $\mathbf{u}_{lm}^{(i)}$, $\mathbf{c}_{lm}^{(i)} \in \mathbb{F}_q^D$, is:

$$\mathbf{c}_{lm}^{(i)} = \left(\left[\beta_{1lm}^{(i)} \cdot \boldsymbol{\chi}_{1m}^{(i)}, \beta_{2lm}^{(i)} \cdot \boldsymbol{\chi}_{2m}^{(i)}, \dots, \beta_{Klm}^{(i)} \cdot \boldsymbol{\chi}_{Km}^{(i)} \right]^T \right) \text{ mod } q. \quad (10)$$

4.5. Empirical Probability Distributions

Since lossless transmissions from the relays to the destination are assumed, the total number of linearly independent codeword combinations at the relays for each chunk is the same at the destination. $r^{(i)}$ (r for any chunk) denotes the number of linearly independent codeword combinations correctly received at the relays (destination) for chunk i . Hence, $r^{(i)}$ is the rank of matrix $\mathbf{C}^{(i)} \in \mathbb{F}_q^{D^{(i)} \times r^{(i)}}$, which is a set of $r^{(i)}$ linearly independent vectors taken from $L \cdot M$ vectors $[\mathbf{c}_{11}^{(i)}, \dots, \mathbf{c}_{L1}^{(i)}, \mathbf{c}_{12}^{(i)}, \dots, \mathbf{c}_{L2}^{(i)}, \dots, \mathbf{c}_{LM}^{(i)}]$.

The original blocks of all sources for chunk i are recoverable if there are $D^{(i)}$ linearly independent received codeword combinations for chunk i , i.e., $r^{(i)} = D^{(i)}$. If the channel state is stable, i.e., $r^{(i)}$ is constant for all i , all chunks can be decoded with a suitable value of M without the need for feedback from the destination. However, with the unstable channel state, $r^{(i)}$ varies with different chunks. Hence, without the aid of feedback, there are some chunks that are undecodable. As in the works of [17,28], in this paper, $\rho(r)$ denotes the empirical probability distribution of r , for $r \in \{0, 1, \dots, D_{\max}\}$, where $D_{\max} = \max\{D^{(i)}, \forall i\}$.

On the other hand, in this paper, $\theta_k^{(i)}$ (θ_k for any chunk) denotes the rank of the part of the matrix $\mathbf{C}^{(i)}$ from row $1 + \sum_{k'=1}^{k-1} d_{k'}^{(i)}$ to row $\sum_{k'=1}^k d_{k'}^{(i)}$. $\theta_k^{(i)}$ is defined as the participation factor of source k in $\mathbf{C}^{(i)}$, i.e., in the forwarded codeword combinations of chunk i . In addition, $\lambda_k(\theta_k)$ denotes the empirical probability distribution of θ_k , for $\theta_k \in \{0, 1, \dots, d_k\}$.

In practical applications, $\rho(r)$ and $\lambda_k(\theta_k)$ can be collected by employing a feedback-based transmission scheme only for the chunks without feedback loss, as in the work of [29], for example. The overhead caused by the linear dependence between coded blocks and between codeword combinations, i.e., due to the small value of finite field size q , is taken into account in the data collections of $\rho(r)$ and $\lambda_k(\theta_k)$. In addition, exploiting the probability distributions for the design of OCC/CF enables OCC/CF to be applicable to the other channel distributions, ensuring its robustness.

4.6. Decodability

In order to analyze the decodability, in this paper, p_d and p_k denote the probabilities that chunk i is decodable, i.e., $r^{(i)} = D^{(i)}$ and $\theta_k^{(i)} = d_k^{(i)}$, respectively, when employing an OCC, which is designed by using $\rho(r)$ and $\lambda_k(\theta_k)$, respectively, in single-transmission flow, i.e., transmission from a source to a relay via an orthogonal channel. The overlapping fashions of OCCs corresponding to $\rho(r)$ and $\lambda_k(\theta_k)$ are the same.

This paper considers the case that $K \geq L$, $M = d_{\max}$, and source k generates M coded blocks by the RLNC encoder with $d_k^{(i)}$ linearly independent coded blocks for chunk i and all k , i.e., $[\boldsymbol{\chi}_{k1}^{(i)}, \boldsymbol{\chi}_{k2}^{(i)}, \dots, \boldsymbol{\chi}_{kM}^{(i)}] \in \mathbb{F}_q^{d_k^{(i)} \times M}$ would be pseudorandom to ensure the linear independence between coded blocks. When employing OCC/CF, the codeword combinations of chunk i are recoverable at

the destination if there are $D^{(i)}$ received linearly independent codeword combinations, i.e., $r^{(i)} = D^{(i)}$. To determine the probability that a chunk is decodable, this paper studies two cases as below:

- Case I: $\beta_{lm}^{(i)} = [\beta_{1lm}^{(i)}, \beta_{2lm}^{(i)}, \dots, \beta_{Klm}^{(i)}]$ is a unit vector, i.e., only an element of $\beta_{lm}^{(i)}$ is equal to one, and the others are zero;
- Case II: $\beta_{lm}^{(i)}$ does not have zero elements; there are only M linearly independent codeword combinations, and they are only forwarded by a relay.

For Case I, $r^{(i)}$ can be written as $r^{(i)} = \sum_{k=1}^K \theta_k^{(i)}$. Hence, the decodability of each chunk only depends on the OCC design using $\lambda_k(\theta_k)$ for all k . In this case, the original blocks of each source can be recovered independently since every received codeword combination corresponds to the coded blocks from only one source. By assuming that chunk i for all sources is decodable, i.e., $r^{(i)} = D^{(i)}$, if the chunks of all sources are decodable, thus the probability that a chunk for all sources is decodable, p_{dec} , can be written as $p_{\text{dec}} = \prod_{k=1}^K p_k$, which is independent of p_d or $\rho(r)$.

For Case II, since $M \geq \max\{d_k^{(i)}, 1 \leq k \leq K, \forall i\}$ and there are $d_k^{(i)}$ linearly independent coded blocks from source k for chunk i , hence $\theta_k^{(i)} = d_k^{(i)}$ for chunk i . This case assumes that chunk i is not decodable and there are $\gamma_k^{(i)} \leq d_k^{(i)}$ blocks inside chunk i for source k with $k \in \{1, 2, \dots, K\}$, which also belong to the other chunks. If these $\gamma_k^{(i)}$ blocks have been already recovered with the decoded chunks, then there are still $\sum_{k=1}^K (d_k^{(i)} - \gamma_k^{(i)}) = D^{(i)} - \sum_{k=1}^K \gamma_k^{(i)}$ to recover for chunk i . From another point of view, it is equivalent to the case that matrix $\mathbf{C}^{(i)}$ has $\gamma_k^{(i)}$ eliminated rows, which are between row $1 + \sum_{k'=1}^{k-1} d_{k'}^{(i)}$ and row $\sum_{k'=1}^k d_{k'}^{(i)}$, and becomes a $(D^{(i)} - \sum_{k=1}^K \gamma_k^{(i)}) \times M$ matrix, $\mathbf{C}'^{(i)}$. Since $\chi_{km}^{(i)}$ is randomly drawn from $\mathbb{F}_q^{d_k^{(i)}}$ for $m \in \{1, 2, \dots, M\}$, hence $\mathbf{C}'^{(i)}$ can be approximately also drawn from $\mathbb{F}_q^{d_k^{(i)}}$. In addition, $\mathbf{C}'^{(i)}$ can be approximately obtained by eliminating $\sum_{k=1}^K \gamma_k^{(i)}$ rows from a $D^{(i)} \times M$ matrix, which is randomly drawn from $\mathbb{F}_q^{D^{(i)} \times M}$. It looks like $\sum_{k=1}^K \gamma_k^{(i)}$ blocks are back-substituted into a chunk i when employing OCC in single-flow transmission. Then, the decodability of each chunk when employing OCC/CF is the same as when employing OCC designed using $\rho(r)$ in single-flow transmission. Hence, in this case, $p_{\text{dec}} = p_d$. With Case II, the feature is that already recovered $\gamma_k^{(i)} \leq d_k^{(i)}$ blocks can be back-substituted into chunk i without waste.

In contrast, for the other case, by taking $\theta_k^{(i)} < d_k^{(i)}$ and $\gamma_k^{(i)} = d_k^{(i)} - \theta_k^{(i)}$ for example, these recovered blocks can successfully increase the number of linearly independent received coded blocks in chunk i if they are linearly independent of the existing received coded blocks in chunk i . In addition, the value of $\gamma_k^{(i)}$ should be appropriately selected by using $\theta_k^{(i)}$ or $\lambda_k(\theta_k)$ for all chunks. From the point of view of $\mathbf{C}^{(i)}$, an example taking $K = 2$, $d_1^{(i)} = 2$, $d_2^{(i)} = 2$, $M = 3$, $\gamma_1^{(i)} + \gamma_2^{(i)} = 2$ (according to the OCC design using $\rho(r)$) and $q = 7$ is shown in Figure 5. In Figure 5a,b, $r^{(i)} = 3$, $\theta_1^{(i)} = 2$ and $\theta_2^{(i)} = 2$ are given. By taking $\gamma_1^{(i)} = 1$ and $\gamma_2^{(i)} = 1$, then there are five linearly independent blocks in chunk i after back-substitution, i.e., chunk i is decodable, with four out of six chances. On the other hand, if taking $\gamma_1^{(i)} = 0$, $\gamma_2^{(i)} = 2$ as in Figure 5b, then chunk i is decodable with all three possibilities. Therefore, a suitable selection of $\gamma_1^{(i)}$ and $\gamma_2^{(i)}$ can provide better performance for OCC/CF. Hence, the OCC design using $\lambda_k(\theta_k)$ for all k is needed. In Figure 5c, $r^{(i)} = 3$, $\theta_1^{(i)} = 2$ and $\theta_2^{(i)} = 3$ are given. By taking any two different recovered blocks, chunk i is decodable with nine of ten chances. The undecodable outcome should be caused by the selection of $\gamma_1^{(i)} + \gamma_2^{(i)} = 2$, i.e., the OCC design using $\rho(r)$. Figure 5c represents Case II where $\theta_1^{(i)} = d_1^{(i)} = 2$, $\theta_2^{(i)} = d_2^{(i)} = 3$.

$$\begin{array}{c}
 \text{combined coding} \\
 \text{coefficient vectors} \\
 \text{of recovered blocks} \\
 \text{of Source 1}
 \end{array}
 \quad
 \begin{array}{c}
 \text{combined coding} \\
 \text{coefficient vectors} \\
 \text{of recovered blocks} \\
 \text{of Source 2}
 \end{array}$$

$$\mathbf{C}^{(i)} = \left[\begin{array}{ccc|cc|cc|c}
 \mathbf{1} & \mathbf{2} & \mathbf{3} & \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \text{or} & \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & \text{or} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} & \text{or} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \\
 \mathbf{2} & \mathbf{4} & \mathbf{4} & & & & & & & & & \\
 \mathbf{3} & \mathbf{3} & \mathbf{5} & & & & & & & & & \\
 \mathbf{4} & \mathbf{4} & \mathbf{4} & & & & & & & & & \\
 \mathbf{5} & \mathbf{5} & \mathbf{5} & & & & & & & & &
 \end{array} \right]$$

$$d_1^{(i)} = 2, d_2^{(i)} = 3, \gamma_1^{(i)} = 1, \gamma_2^{(i)} = 1$$

$$r^{(i)} = 3, \theta_1^{(i)} = 2, \theta_2^{(i)} = 2$$

4 of 6 possibilities are decodable

(a)

$$\begin{array}{c}
 \text{combined coding coefficient vectors} \\
 \text{of recovered blocks of Source 2}
 \end{array}$$

$$\mathbf{C}^{(i)} = \left[\begin{array}{ccc|cc|cc|c}
 \mathbf{1} & \mathbf{2} & \mathbf{3} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} & \text{or} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} & \text{or} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \\
 \mathbf{2} & \mathbf{4} & \mathbf{4} & & & & & & & & & \\
 \mathbf{3} & \mathbf{3} & \mathbf{5} & & & & & & & & & \\
 \mathbf{4} & \mathbf{4} & \mathbf{4} & & & & & & & & & \\
 \mathbf{5} & \mathbf{5} & \mathbf{5} & & & & & & & & &
 \end{array} \right]$$

$$d_1^{(i)} = 2, d_2^{(i)} = 3, \gamma_1^{(i)} = 0, \gamma_2^{(i)} = 2$$

$$r^{(i)} = 3, \theta_1^{(i)} = 2, \theta_2^{(i)} = 2$$

All 3 possibilities are decodable

(b)

$$\begin{array}{c}
 \text{combined coding coefficient vectors of} \\
 \text{recovered blocks of Source 1 and Source 2}
 \end{array}$$

$$\mathbf{C}^{(i)} = \left[\begin{array}{ccc|cc|cc|c}
 \mathbf{1} & \mathbf{2} & \mathbf{4} & \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \\
 \mathbf{2} & \mathbf{1} & \mathbf{6} & & & & & & & & & \\
 \mathbf{3} & \mathbf{4} & \mathbf{5} & & & & & & & & & \\
 \mathbf{4} & \mathbf{5} & \mathbf{3} & & & & & & & & & \\
 \mathbf{5} & \mathbf{3} & \mathbf{4} & & & & & & & & &
 \end{array} \right]$$

Selecting any two of five recovered blocks

$$d_1^{(i)} = 2, d_2^{(i)} = 3, \gamma_1^{(i)} + \gamma_2^{(i)} = 2$$

$$r^{(i)} = 3, \theta_1^{(i)} = 2, \theta_2^{(i)} = 3$$

9 of 10 possibilities are decodable

(c)

Figure 5. Example from point of view of the combined coding coefficient matrix. (a) $\gamma_1^{(i)} = 1, \gamma_2^{(i)} = 1, \theta_1^{(i)} = 2, \theta_2^{(i)} = 2$; (b) $\gamma_1^{(i)} = 0, \gamma_2^{(i)} = 2, \theta_1^{(i)} = 2, \theta_2^{(i)} = 2$; (c) $\gamma_1^{(i)} + \gamma_2^{(i)} = 2, \theta_1^{(i)} = 2, \theta_2^{(i)} = 3$. The decodability of chunk i is given with different possibilities of recovered blocks.

For the general case, by combining the two cases above, the effective probability that each chunk is decodable when OCC/CF is applied, denoted by p_{deff} , can be approximately obtained by:

$$p_{\text{deff}} = p_d \cdot \prod_{k=1}^K p_k. \tag{11}$$

On the other hand, for the case that $K > L > 1$, the values of M and $d_k^{(i)}$ for $k \in \{1, 2, \dots, K\}$ should be selected appropriately such that any chunk i can be decoded by itself, i.e., $r^{(i)} = D^{(i)}$.

For example, if taking $d_1^{(i)} = d_2^{(i)} = \dots = d_K^{(i)} = d_{\max}$ for all i , then d_{\max} should be chosen as a multiple of L , and $M \geq \frac{K \cdot d_{\max}}{L}$ to ensure that there is at most $L \cdot M$ codeword combinations to recover $K \cdot d_{\max}$ original blocks. For the case that $1 < K \leq L$, M can be taken by $M \geq d_{\max}$ for any value of $d_k^{(i)} > 0, \forall k$.

4.7. Channel Efficiency

In this paper, channel efficiency is defined as the ratio of the total number of decoded blocks from all sources to the total transmission time (the total number of time slots for OCC/CF or for the transmission schemes without the need for feedback from the relays) taken from the sources to the relays. η and η_{eff} denote the channel efficiencies corresponding to p_d and p_{def} , respectively.

For a K -source L -relay network, the ideal value of channel efficiency, which is obtained with lossless transmission and without linear dependence between codeword combinations, is $\min\{K, L\}$. Thus, for $L = 1$, the channel efficiency would be like in the case of single flow transmission via an orthogonal channel. Hence, applying an orthogonal channel might be a better option. This paper only considers the case that $L > 1$.

On the other hand, $\bar{\rho}$ denotes $\sum_{r=1}^{D_{\max}} r \cdot \rho(r)$, and $\bar{\eta}$ denotes $\frac{\bar{\rho}}{M}$. $\bar{\eta}$ is called channel capacity in this paper, i.e., the upper bound of η_{eff} . Many OCC designs in single-flow transmission try to obtain η_{eff} close to $\bar{\eta}$. In this paper, the (design) overhead is defined as the gap between η_{eff} and $\bar{\eta}$.

5. Design with an Overlapped Chunked Code

5.1. Encoding

This paper applies an OCC in a contiguously overlapping fashion, which is similar to the works of [14,15], but not in a rounded-end fashion for the design of OCC/CF in a multi-source multi-relay network where CF based on NLC is employed. The applied overlapping fashion is shown in Figure 6.

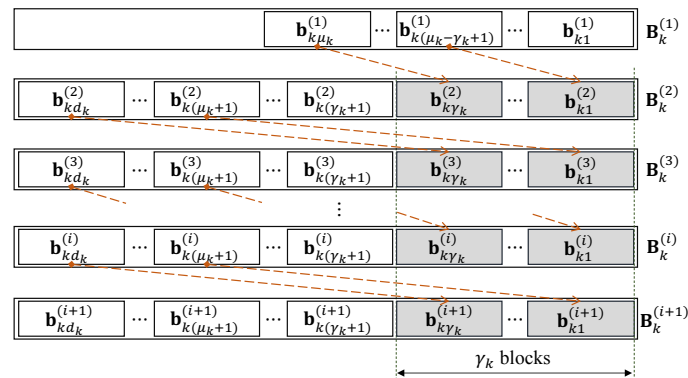


Figure 6. Applied overlapping fashion of overlapped chunked codes (OCC) for source k . The blocks in grey for chunk i are the blocks taken from the previous chunk, i.e., chunk $i - 1$.

In this fashion, for source k and each chunk, there are $\mu_k > 0$ innovative blocks, i.e., linearly independent blocks if comparing to the blocks of the other chunks, and there are γ_k overlapped blocks between two contiguous chunks. Hence, there are $d_k = \mu_k + \gamma_k$ blocks for all chunks except the first chunk where there are only μ_k blocks, i.e., $d_k^{(1)} = \mu_k$, since it is not the rounded-end fashion. μ and γ are defined as $\sum_{k=1}^K \mu_k$ and $\sum_{k=1}^K \gamma_k$, respectively. There are $\min\{M, d_k^{(i)}\}$ linearly independent coded blocks among M coded blocks for chunk i and source k , where $[\mathcal{X}_{k1}^{(i)}, \mathcal{X}_{k2}^{(i)}, \dots, \mathcal{X}_{kM}^{(i)}] \in \mathbb{F}_q^{d_k^{(i)} \times M}$ should be pseudorandomly generated to achieve this goal.

5.2. Decoding

The feature of OCC is that a decoded chunk can help the other undecodable chunks in decoding by using back-substitution (b.s). The recovered blocks of the decoded chunk are substituted into the undecodable chunks that consist of the same blocks, i.e., the overlapped blocks. Thus, the number of linearly independent codeword combinations of the back-substituted chunks might be increased, and it depends on the value of q and the pairs (d_k, γ_k) for all k [28]. With the OCC employed in this paper, for chunk i , left back substitution (l.b.s) and right back-substitution (r.b.s) denote b.s by the decoded neighboring chunk on the left, i.e., chunk $i - 1$, and on the right, i.e., chunk $i + 1$, respectively.

In addition to b.s, this paper considers the other decoding opportunity, called combination of chunks (co.cs) for the applied OCC. co.cs combines the contiguous undecodable chunks into the form of chain of chunks (ch.cs) with length $\phi \geq 1$, where ϕ is the number of the combined chunks. The decoding process can start without the need for at least an already decoded neighboring chunk as with b.s. The form of combined coding coefficient matrix of co.cs, \mathbf{C}_c , is shown in Figure 7.

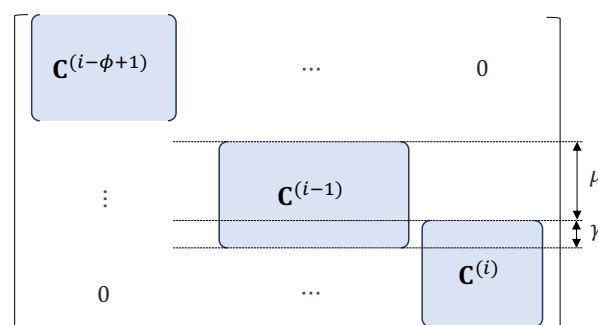


Figure 7. The form of combined coding coefficient matrix of the chain of chunks with length ϕ .

A ch.cs is decodable if the rank of \mathbf{C}_c , $\text{rank}(\mathbf{C}_c)$, is equal to the total number of original blocks inside that ch.cs, which is denoted by r_{ch} . A ch.cs is considered as a directly undecodable ch.cs without waiting to receive a new chunk if $\text{rank}(\mathbf{C}_c)$ is lower than a threshold value denoted by r_{th} . r_{co} and r_{th} are determined as described in Algorithm 1 by using the feature of the applied OCC. Chunk i can participate in the co.cs process if $r_p^{(i)} \leq r^{(i)} \leq D^{(i)} - 1$, where $r_p^{(i)}$ is determined as described in Algorithm 2. In Algorithms 1 and 2, l.b.s.s(i) and r.b.s.s(i) refer to the state of l.b.s and r.b.s, respectively, for considered chunk i . l.b.s.s(i) and r.b.s.s(i) declare whether undecodable chunk i has not been back-substituted by its left neighboring decoded chunk, i.e., chunk $i - 1$, and by its right neighboring decoded chunk, i.e., chunk $i + 1$, respectively. $r_p^{(t)} = 0$ means that chunk t cannot participate in co.cs, and its decodability depends on $r^{(t)}$. The process of co.cs is described in Algorithm 3, where d.s(ch.cs) refers to the decodability state of the currently obtained ch.cs.

The decoding process can be done as described in Algorithm 4, where d.s(t) and d.s($t - \phi + 1 : t$) declare whether chunk t and ch.cs, combining from chunk t back to $t - \phi + 1$, respectively, are decoded or not. The decoding process for a ch.cs with length $\phi \geq 2$ can be done by using the inactivation decoding method [23] in order to reduce the decoding complexity. However, Gaussian elimination is applied for the decoding process in this paper.

The chunks that are considered as directly undecodable chunks without waiting for the next received chunks can become decodable by the aid of feedback from the destination back to the sources. Otherwise, they can be discarded in order to reduce the storage overhead if they do not affect the recovery of all blocks, i.e., the original message. The latter option can be achievable by applying precoding before OCC at each source. With precoding, the original blocks can be recovered when a fraction of all coded blocks is decoded [12].

Algorithm 1 Determining r_{th} and r_{ch} of a co.cs starting from chunk t with length ϕ .

```

1: if  $t - \phi + 1 = 1$  then
2:    $r_{ch} = \phi \cdot \mu$ 
3:   if  $r.b.s.s(t) = \text{true}$  then
4:      $r_{th} = \phi \cdot \mu$ 
5:   else
6:      $r_{th} = \phi \cdot \mu - \gamma$ 
7:   end if
8: else
9:    $r_{ch} = \phi \cdot \mu + \gamma$ 
10:  if  $l.b.s.s(t - \phi + 1) = \text{true}$  and  $r.b.s.s(t) = \text{true}$  then
11:     $r_{th} = \phi \cdot \mu + \gamma$ 
12:  else if  $l.b.s.s(t - \phi + 1) = \text{true}$  and  $r.b.s.s(t) = \text{false}$  then
13:     $r_{th} = \phi \cdot \mu$ 
14:  else if  $l.b.s.s(t - \phi + 1) = \text{false}$  and  $r.b.s.s(t) = \text{true}$  then
15:     $r_{th} = \phi \cdot \mu$ 
16:  else if  $l.b.s.s(t - \phi + 1) = \text{false}$  and  $r.b.s.s(t) = \text{false}$  then
17:     $r_{th} = \phi \cdot \mu - \gamma$ 
18:  end if
19: end if

```

Algorithm 2 Determining $r_p^{(t)}$ for chunk t .

```

1: if  $t = 1$  then
2:   if  $r.b.s.s(t) = \text{true}$  then
3:      $r_p^{(t)} = 0$ 
4:   else
5:      $r_p^{(t)} = \mu - \gamma + 1$ 
6:   end if
7: else
8:   if  $l.b.s.s(t) = \text{true}$  and  $r.b.s.s(t) = \text{true}$  then
9:      $r_p^{(t)} = 0$ 
10:  else if  $l.b.s.s(t) = \text{true}$  and  $r.b.s.s(t) = \text{false}$  then
11:     $r_p^{(t)} = \mu + 1$ 
12:  else if  $l.b.s.s(t) = \text{false}$  and  $r.b.s.s(t) = \text{true}$  then
13:     $r_p^{(t)} = \mu + 1$ 
14:  else if  $l.b.s.s(t) = \text{false}$  and  $r.b.s.s(t) = \text{false}$  then
15:     $r_p^{(t)} = \mu - \gamma + 2$ 
16:  end if
17: end if

```

Algorithm 3 Combination of chunks.

```

1: Starting from chunk  $i$ 
   Taking  $t = i$ ,  $d.s(\text{ch.cs}) = \text{false}$ ,  $\phi = 1$ ,  $\mathbf{C}_c = \mathbf{C}^{(t)}$ 
2: if  $\text{rank}(\mathbf{C}_c) = r_{\text{ch}}$  then
3:   Update  $d.s(\text{ch.cs}) = \text{true}$ 
4:   return  $d.s(\text{ch.cs}), \phi$ 
5: else
6:   if  $t = 1$  or  $\text{rank}(\mathbf{C}_c) < r_p^{(t)}$  or  $\text{l.b.s.s}(t) = \text{true}$  then
7:     return  $d.s(\text{ch.cs}), \phi$ 
8:   end if
9: end if
10: while  $t - 1 > 0$  and  $r_p^{(t-1)} \leq r^{(t-1)} \leq D - 1$  and  $\text{r.b.s.s}(t - 1) = \text{false}$  do
11:   Update  $\phi = \phi + 1$ ,  $\mathbf{C}_c = [\mathbf{C}_c, \mathbf{C}^{(t-1)}]$ 
12:   if  $\text{rank}(\mathbf{C}_c) = r_{\text{ch}}$  then
13:     Update  $d.s(\text{ch.cs}) = \text{true}$ 
14:     return  $d.s(\text{ch.cs}), \phi$ 
15:   else
16:     Update  $t = t - 1$ 
17:   end if
18: end while

```

Algorithm 4 Decoding process.

```

1: Obtaining the codeword combinations of chunk  $i$ 
2: Take  $t = i$ 
3: if  $r^{(t)} = D$  and  $d.s(t) = \text{false}$  then
4:   Conduct decoding and update  $d.s(t) = \text{true}$ ,  $t = t - 1$ 
5:   Go to 3
6: else
7:   if  $t - 1 > 0$  and  $d.s(t - 1) = \text{true}$  and  $\text{l.b.s.s}(t) = \text{false}$  then
8:     Conduct l.b.s and update  $\text{l.b.s.s}(t) = \text{true}$ 
9:   end if
10:  if  $d.s(t + 1) = \text{true}$  and  $\text{r.b.s.s}(t) = \text{false}$  then
11:    Conduct r.b.s and update  $\text{r.b.s.s}(t) = \text{true}$ 
12:  end if
13:  if  $r^{(t)} = D$  and  $d.s(t) = \text{false}$  then
14:    Conduct decoding and
15:    update  $d.s(t) = \text{true}$ ,  $t = t - 1$ 
16:    Go to 3
17:  else
18:    if  $\text{l.b.s.s}(t) = \text{false}$  and  $t - 1 > 0$ 
19:      and  $d.s(t - 1) = \text{false}$  then
20:        Conduct co.cs to obtain ch.cs with length  $\phi$ 
21:        if  $d.s(\text{ch.cs}) = \text{true}$  then
22:          Conduct decoding and
23:          update  $d.s(t - \phi + 1 : t) = \text{true}$ ,  $t = t - \phi$ 
24:          Go to 3
25:        end if
26:      end if
27:    end if
28:  end if
29:  Wait to receive the codeword combinations of chunk  $i + 1$ 

```

5.3. Design with Applied Overlapped Chunked Code

The design of the applied OCC for multi-source multi-relay, i.e., the design of OCC/CF, is to determine an allocation $[(\mu_1, \gamma_1), (\mu_2, \gamma_2), \dots, (\mu_K, \gamma_K)]$ with the desired p_{deff} or with the desired effective channel efficiency η_{eff} . For convenience, this paper takes M as the maximum chunk size, i.e., $M \geq \max\{d_k, 1 \leq k \leq K\}$ and M can provide $\min\{\theta_k, 1 \leq k \leq K\} > 0$. If N_k is the total number of blocks for source k , the number of chunks that contain the blocks of all sources is equal to $\min \left\{ \left\lfloor \frac{N_k}{\mu_k} \right\rfloor, 1 \leq k \leq K \right\}$.

The finite number of chunks for the applied OCC might cause high overhead in single-flow transmission if compared with the other codes such as in the works of [16–18,30], which have a rateless feature. However, the design of the applied OCC in a multi-source multi-relay network might be simpler if compared with the other designs that determine the probability distribution of chunk size for all sources, for example. By taking the design of BATs codes [17] as an example, d_k is selected according to a determined degree distribution $\Psi_k = \{\psi_0, \psi_1, \dots, \psi_M\}$ for each chunk. Determining Ψ_k for all k must consider the outputs of p_k , p_d and p_{deff} , while Ψ_k for all k needs to satisfy $\Psi_1 * \Psi_2 * \dots * \Psi_K = \Psi$ [31], where Ψ is the degree distribution of $D = \sum_{k=1}^K d_k$ and sign $*$ refers to the discrete-time convolution operation. It becomes more complicated to determine Ψ_k for all k when K is large.

For the design of OCC/CF, d_k is fixed for all chunks except for the first chunk. Thus, there are $1 + 2 + \dots + M = \frac{M(M+1)}{2}$ candidates of (μ_k, γ_k) for source k and $\left[\frac{M(M+1)}{2} \right]^K$ candidates of $[(\mu_1, \gamma_1), (\mu_2, \gamma_2), \dots, (\mu_K, \gamma_K)]$ for all sources. It would be less if fixing $d_k = M$ and varying only γ_k for all k , as in the previous work of this paper [1], where there are only M candidates for each source and M^K candidates for all sources. However, the decoding complexity and the storage overhead at the destination might be high. For this work, the chunk size for each source is not large and bounded by M ; however, it can be large, as in the work of [14]. A larger chunk size with a large number of overlapped blocks can improve the decodability of all chunks, but it can cause high computational complexity, especially decoding complexity and storage overhead at the destination. An undecodable chunk needs to wait for several new received chunks to start decoding, i.e., ϕ is large. For example, if taking $\mu = 14$, $\gamma = 18$, $\bar{\rho} = 15$ and $M < D$, where $D = \mu + \gamma = 32$, then there are no chunks that can be decoded by themselves, i.e., $r = D$ without using b.s or co.cs. The decoding process can only start by co.cs, where ϕ at least satisfies:

$$\phi \cdot \bar{\rho} \geq \phi \cdot \mu + \gamma, \quad (12)$$

hence, $\phi \geq 18$.

With a large chunk size, the decoding process rarely starts with back-substitution, i.e., the ch.cs with length $\phi = 1$. The decoding process only start with co.cs with large ϕ . In this work, the length ϕ is bounded for the purpose of low decoding complexity by providing more opportunities to conduct b.s and to reduce latency, as an undecodable chunk needs to wait to become decodable.

6. Estimation of Decodability

6.1. Overview

In order to select an appropriate allocation $[(\mu_1, \gamma_1), (\mu_2, \gamma_2), \dots, (\mu_K, \gamma_K)]$ for the desired purpose, the estimation of p_{deff} (also η_{eff}) is done for each possible allocation. The estimations of p_k and p_d are conducted separately for each possible allocation. Then, p_{deff} is determined by (11), and η_{eff} is determined by:

$$\eta_{\text{eff}} = \mu \cdot p_{\text{deff}}. \quad (13)$$

In the previous work of this paper [1], the estimation was done by conducting a simulation to obtain the performance in p_d and p_k of all allocations. Alternatively, in this paper, the estimation is done by conducting table lookup and the accumulative sum of the probabilities that ch.cs with

the maximum length ϕ_{\max} are decodable for all possible combinations of $[r^{i-\phi+1}, r^{i-\phi+2}, \dots, r^i]$ and $[\theta_k^{i-\phi+1}, \theta_k^{i-\phi+2}, \dots, \theta_k^i]$ to determine p_d and p_k , respectively, using $\rho(r)$ and $\lambda(\theta_k)$, respectively, for $\phi \in \{1, 2, \dots, \phi_{\max}\}$ and $i > 1$. For convenience, only the estimation of p_d is described, and the estimation of p_k for all k can be done similarly.

At the start, it is assumed that an OCC with the fashion as in Figure 6 is applied from a sender to a receiver in single-flow transmission, and $\rho(r)$ with $r \in \{1, 2, \dots, D_{\max}\}$ is the obtained empirical rank distribution. The chunk size D is selected from the range value of r , then the maximum number of linearly independent codeword combinations (coded blocks) received per chunk becomes D . $\rho(r)$ is updated to $\rho'(r')$ where $r' \in \{1, 2, \dots, D\}$. This paper estimates $\rho'(r')$ from $\rho(r)$ by:

$$\rho'(r') = \begin{cases} \rho(r), & \text{if } r' \in \{1, 2, \dots, D-1\}. \\ \sum_{r=D}^{D_{\max}} \rho(r), & \text{if } r' = D. \end{cases} \quad (14)$$

For the estimation and for convenience, another four back-substitution states for a chunk are defined as below:

- (i) not back-substituted state (n.b.s.s): equivalent to the event that l.b.s.s and r.b.s.s for a chunk are all false;
- (ii) half back-substituted state (h.b.s.s): equivalent to the event that one of l.b.s.s and r.b.s.s is true;
- (iii) full back-substituted state (f.b.s.s): equivalent to the event that l.b.s.s and r.b.s.s are all true;
- (iv) quasi-half back-substituted state (q.b.s.s): equivalent to n.b.s.s in the b.s process and equivalent to h.b.s.s in the co.cs process.

n.b.s.s(r'), for example, refers to a chunk that has r' linearly independent coded blocks, and its state is n.b.s.s. $q_n(r')$, $q_h(r')$, $q_q(r')$ and $q_f(r')$ denote the probabilities that a chunk has r' linearly independent coded blocks and has n.b.s.s, h.b.s.s, f.b.s.s and q.b.s.s, respectively. They satisfy the condition below.

$$\sum_{r'=1}^D [q_n(r') + q_h(r') + q_q(r') + q_f(r')] = 1. \quad (15)$$

Initially, $q_n(r') = \rho'(r')$ and $q_h(r') = q_q(r') = q_f(r') = 0$ for $r' \in \{1, 2, \dots, D\}$ are given. The estimation here is to update $q_n(r')$, $q_h(r')$, $q_q(r')$ and $q_f(r')$ according to the decoding process for all values of r' . If ρ_d denotes the probability distribution of the number of linearly independent coded blocks in a chunk after conducting the updating process, then:

$$\rho_d(r') = q_n(r') + q_h(r') + q_q(r') + q_f(r'), \text{ for } r' \in \{1, 2, \dots, D\}. \quad (16)$$

In the updating process, the chunk with n.b.s.s, h.b.s.s and q.b.s.s is active, i.e., $q_n(r')$, $q_h(r')$ and $q_q(r')$ are used to conduct the updating process, and the chunk with f.b.s.s is inactive, i.e., $q_f(r')$ cannot be used to conduct the updating process and is only used in determining ρ_d . The updating process is to try transforming n.b.s.s(r') for all r' to the chunks with other states, i.e., to make $q_n(r')$ tend to zero for all r' . At the end of the updating process, p_d is obtained by taking $p_d = \rho_d(D)$.

The updating process is divided into two parts: b.s and co.cs, which are for $\phi = 1$ and for $2 \leq \phi \leq \phi_{\max}$, respectively. This paper assumes that the estimation of decodability is done at the destination. The destination informs about the desired allocation to the sources via feedback.

6.2. Combination of Chunks

A ch.cs with C_c as in Figure 7 with length ϕ , where $2 \leq \phi \leq \phi_{\max}$, is considered. The rank array of C_c is $[r^{(i-\phi+1)}, r^{(i-\phi+2)}, \dots, r^{(i)}]$, where $r^{(t)}$ is the rank of $C^{(t)}$ for $t \in \{i-\phi+1, i-\phi+2, \dots, i\}$. A combination of $[r^{(i-\phi+1)}, r^{(i-\phi+2)}, \dots, r^{(i)}]$ for a ch.cs (simply combination for convenience) with length ϕ is considered as a possible combination to be taken into account in the estimation if it satisfies:

$$\sum_{t=i-\phi+1}^i r'^{(t)} \geq \phi\mu + \gamma, \quad (17)$$

and it does not contain any possible combination with length ϕ' inside, where $\phi' \in \{1, 2, \dots, \phi - 1\}$. The ch.cs corresponding to a possible combination is decodable if $\text{rank}(\mathbf{C}_c) = \phi \cdot \mu + \gamma$. p_s denotes the probability that a combination can make the correspondent ch.cs decodable and $q_s = 1 - p_s$. All possible combinations and their p_s are obtained by conducting a computation in MATLAB in this work and known by the destination where table lookup is done while doing the estimation.

Based on three different locations of a chunk in a ch.cs, the other three probabilities are defined as below.

- (i) $q_b(r')$: the probability that a chunk that has r' linearly independent coded blocks can play the role of the beginning chunk, i.e., $t = i$. The beginning chunk can be with n.b.s.s or h.b.s.s or q.b.s.s.
- (ii) $q_i(r')$: the probability that a chunk that has r' linearly independent coded blocks can play the role of the intermediate chunk, i.e., $t \in \{i - \phi + 2, i - \phi + 3, \dots, i - 1\}$. The intermediate chunk only can be with n.b.s.s. Hence, $q_i(r') = q_n(r')$.
- (iii) $q_e(r')$: the probability that a chunk that has r' linearly independent coded blocks can play the role of the ending chunk, i.e., $t = i - \phi + 1$. The ending chunk can be with n.b.s.s, or h.b.s.s, or q.b.s.s. The beginning chunk and the ending chunk have a similar property because the combinations are reversible. Thus, only the beginning chunk is studied, and $q_e(r') = q_n(r') + q_h(r') + q_q(r')$ is taken.

Chunk t is decodable, i.e., ch.cs that contains chunk t and $t \in \{i - \phi + 1, \dots, i\}$ is decodable, by probability p_c , which is:

$$p_c = p_s q_b(r'^{(i)}) q_e(r'^{(i-\phi+1)}) \prod_{t=i-\phi+2}^{i-1} q_i(r'^{(t)}). \quad (18)$$

When the beginning chunk $t = i$ is focused on, it must have $r'^{(t)} \geq \mu + 1$, as in the case that one of l.b.s.s (t) and r.b.s.s (t) is true in Algorithm 2. For example, by taking $D = 18$ and $\gamma = 4$, a combination with rank array [16, 14, 14, 17] is a possible combination where $r'^{(t)} = 16$ and $p_s \approx 0.9663$. However, the combination with rank array [16, 15, 14, 17] is not a possible combination because it contains a possible combination with rank array [15, 14, 17].

When the intermediate chunk $t \in \{i - \phi + 2, \dots, i - 1\}$ is focused on, it must have $r'^{(t)} \geq \mu - \gamma + 2$, as in the case that l.b.s.s (t) and r.b.s.s (t) are all false in Algorithm 2. However, for $r'^{(t)} > \mu$, the possible combinations that have the same rank arrays as those of the possible combination when focusing on the beginning chunk with the same value of $r'^{(t)}$ are not included. In this estimation, the possible combinations that are both available while focusing on the beginning chunk and the intermediate chunk and have the same elements of the rank array, and the same focused chunk with n.b.s.s is applied once in the updating process with co.cs. This is because there is no constraint on the order of chunks in a ch.cs for Relation (18). For example, by taking $D = 18$ and $\gamma = 4$, two combinations with rank array [16, 13, 15, 16] focusing on the chunk with $r'^{(t)} = 13$ and with rank array [16, 15, 16] focusing on the chunk with $r'^{(t)} = 15$ are used because combinations with rank arrays [13, 16, 15, 16] focusing on the chunk with $r'^{(t)} = 13$ and with rank array [15, 16, 16] focusing on the chunk with $r'^{(t)} = 15$ are not possible combinations. However, the combination with rank array [16, 13, 15, 16] focusing on the chunk with $r'^{(t)} = 15$ is not a possible combination because the combination with rank array [15, 16, 13, 16] focusing on the chunk with $r'^{(t)} = 15$ is a possible combination.

The purpose of introducing q.b.s.s is described by the following example. By taking $D = 18$ and $\gamma = 3$, a combination with rank array [17, 16, 14, 17] has $p_s \approx 0.9473$. However, it is not a possible combination since the combination with rank array [17, 16] is a possible combination with $p_s \approx 0.8369$. Then, no matter how the rest of the decoding process (with b.s) is done, the combination with rank array [17, 16, 14, 17] is decodable with p_s less than 0.8369, which is lower than the real value. Hence,

in order to fix this problem, q.b.s.s is introduced by assuming that the beginning chunk with n.b.s.s of an undecodable possible combination (with a probability of $q_c = q_s q_b(r^{(i)}) q_e(r^{(i-\phi+1)}) \prod_{t=i-\phi+2}^{i-1} q_i(r^{(t)})$) becomes a chunk with $r^{(i)} = D - 1$, but no longer with n.b.s.s, and with a state like h.b.s.s during the updating process with co.cs and with a state like n.b.s.s during the updating process with b.s. After introducing q.b.s.s, the ch.cs with the rank array [17, 16, 14, 17] is decodable with a probability around 0.9498, which is close to the real value.

The relationship (transition) between four states of a chunk for the updating process with co.cs is shown in Table 2, where p_t refers to the transfer probability.

Table 2. Transition table between four chunk states for the updating process with the combination of chunks (co.cs). h.b.s.s, half back-substituted state; f.b.s.s, full back-substituted state; n.b.s.s, not back-substituted state; q.b.s.s, quasi-half back-substituted state.

Condition for Focused Chunk t with $r^{(t)}$	From	To	$q_b(r^{(i)})$	p_t
$t = i; r^{(t)} \geq \mu + 1$	h.b.s.s($r^{(t)}$)	f.b.s.s(D)	$q_h(r^{(t)})$	p_c
$t = i; r^{(i)} \geq \mu + 1$	h.b.s.s($r^{(t)}$)	h.b.s.s($r^{(t)}$)	$q_h(r^{(t)})$	q_c
$i - \phi + 2 \leq t \leq i - 1; r^{(t)} \geq \mu - \gamma + 2$	n.b.s.s($r^{(t)}$)	f.b.s.s(D)	$q_e(r^{(i)})$	p_c
$i - \phi + 2 \leq t \leq i - 1; r^{(t)} \geq \mu - \gamma + 2$	n.b.s.s($r^{(t)}$)	n.b.s.s($r^{(t)}$)	$q_e(r^{(i)})$	q_c
$t = i; r^{(t)} \geq \mu + 1$	n.b.s.s($r^{(t)}$)	h.b.s.s(D)	$q_n(r^{(t)})$	p_c
$t = i; r^{(t)} \geq \mu + 1$	n.b.s.s($r^{(t)}$)	q.b.s.s($D - 1$)	$q_n(r^{(t)})$	q_c
$t = i; r^{(t)} = D - 1$	q.b.s.s($r^{(t)}$)	h.b.s.s(D)	$q_q(r^{(t)})$	p_c
$t = i; r^{(t)} \geq \mu + 1$	q.b.s.s($r^{(t)}$)	q.b.s.s($r^{(t)}$)	$q_q(r^{(t)})$	q_c

The updating process with co.cs is done by applying the relationship in Table 2 and by following the concept of same increment same decrement, i.e., the decrement of the probability of a state (n.b.s.s or h.b.s.s or q.b.s.s) at a value of r' is the increment of probability of another state at the same value of r' or at the others. For example, by taking $D = 18, \gamma = 3, q = 7, q_n(r') = \{0.1, 0.2, 0.4, 0.3\}$ and $q_h(r') = q_q(r') = q_e(r') = \{0, 0, 0, 0\}$ for $r' \in \{15, 16, 17, 18\}$ and by considering a chunk with n.b.s.s and $r' = 15$ in a possible combination with rank array [16, 15, 17], then $p_s \approx 0.8166$ and $p_c = 0.0065$ are obtained. Then, $q_n(15)$ decreases by p_c , and $q_f(18)$ increases by p_c . The updating process with co.cs can be done repeatedly for a number of iterations. $q_n(r'), q_h(r'), q_q(r')$ and $q_f(r')$ for all values of r' are updated after an iteration of the updating process with co.cs and are used for the next iteration.

6.3. Back-Substitution

The work in [28] provides the probability that γ' recovered (overlapped) blocks successfully help undecodable chunk t that has already had $r^{(t)}$ linearly independent coded blocks turn into a decodable chunk, where $r^{(t)} + \gamma' \geq D$ and $\gamma' \in \{\gamma, 2 \cdot \gamma\}$. In this estimation, the increment of the number of linearly independent coded blocks in an undecodable chunk after conducting b.s by using γ' recovered blocks is also considered. In addition, the probability distribution of the increment is used.

In order to obtain these data, a computation in MATLAB is conducted to obtain the rank probability distribution of matrix $\mathbf{C}^{(t)} \in \mathbb{F}_q^{D \times r^{(t)}}$ with γ' rows eliminated, i.e., a $(D - \gamma') \times r^{(t)}$ matrix; hence, the obtained rank $r_m^{(t)}$ has range of values $\{0, 1, \dots, r_{\max}^{(t)}\}$, where $r_{\max}^{(t)} = \min\{D - \gamma', r^{(t)}\}$. The probability distribution of $r_m^{(t)}$ is denoted by $\rho_m(r_m^{(t)})$. Since the initial rank of $\mathbf{C}^{(t)}$ is $r^{(t)}$, then the probability distribution of the rank of the $(D - \gamma') \times r^{(t)}$ matrix, denoted by $\rho'_m(r_m^{(t)})$ where $r_m^{(t)} \in \{r_{\min}^{(t)}, \dots, r_{\max}^{(t)}\}$ and $r_{\min}^{(t)} = \max\{0, r^{(t)} - \gamma'\}$, is approximately obtained by:

$$\rho'_m(r_m^{(t)}) = \left\{ \rho_m(r_{\min}^{(t)}), \dots, \rho_m(r_{\max}^{(t)}) \right\} \cdot \frac{\sum_{r_m^{(t)}=0}^{r_{\max}^{(t)}} \rho_m(r_m^{(t)})}{\sum_{r_m^{(t)}=r_{\min}^{(t)}}^{r_{\max}^{(t)}} \rho_m(r_m^{(t)})} \tag{19}$$

The rank increment from the aid of γ' overlapped blocks to undecodable chunk t is denoted by $r_i^{(t)}$; hence, $r_i^{(t)} = r_m^{(t)} + \gamma' - r'^{(t)}$, where $r_i^{(t)} \in \{0, 1, \dots, \gamma'\}$. If $\rho_i(r_i^{(t)})$ is the probability distribution of $r_i^{(t)}$, then $\rho_i(r_i^{(t)}) = \rho_m'(r_m^{(t)})$. $p_{bs} = \varrho_n(D) + \varrho_h(D)$ is taken as the fraction (probability) of the decoded chunks that can be used for b.s, and $q_{bs} = 1 - p_{bs}$. For the case that only half b.s (l.b.s or r.b.s) is executable, then $\gamma' = \gamma$. If full b.s is executable, then $\gamma' = 2 \cdot \gamma$. The transition table between four chunk states for the updating process with b.s is shown in Table 3. The focused chunk t has $C^{(t)}$ with rank $r'^{(t)}$.

Table 3. Transition table between four chunk states for the updating process with b.s.

Half or Full b.s or N/A	From	To	p_t
Half	n.b.s.s($r'^{(t)}$)	h.b.s.s($r'^{(t)} + r_i^{(t)}$)	$2 \cdot p_{bs} \cdot q_{bs} \cdot \rho_i(r_i^{(t)})$
Full	n.b.s.s($r'^{(t)}$)	f.b.s.s($r'^{(t)} + r_i^{(t)}$)	$p_{bs}^2 \cdot \rho_i(r_i^{(t)})$
N/A	n.b.s.s($r'^{(t)}$)	n.b.s.s($r'^{(t)}$)	q_{bs}^2
Half	h.b.s.s($r'^{(t)}$)	f.b.s.s($r'^{(t)} + r_i^{(t)}$)	$p_{bs} \cdot \rho_i(r_i^{(t)})$
N/A	h.b.s.s($r'^{(t)}$)	h.b.s.s($r'^{(t)}$)	q_{bs}
Half	q.b.s.s($r'^{(t)}$)	h.b.s.s($r'^{(t)} + r_i^{(t)}$)	$2 \cdot p_{bs} \cdot q_{bs} \cdot \rho_i(r_i^{(t)})$
Full	q.b.s.s($r'^{(t)}$)	f.b.s.s($r'^{(t)} + r_i^{(t)}$)	$p_{bs}^2 \cdot \rho_i(r_i^{(t)})$
N/A	q.b.s.s($r'^{(t)}$)	q.b.s.s($r'^{(t)}$)	q_{bs}^2

The updating process with b.s is done by applying the relationship of four chunk states shown in Table 3 and by following the concept of the same increment same decrement as in the updating process with co.cs. The updating process with b.s can be done repeatedly as with co.cs. After the end of each iteration of the updating process with b.s, p_{bs} is added to $\varrho_f(D)$, and it is updated according to the updated value of $\varrho_h(D)$ because $\varrho_n(D)$ becomes zero after the first iteration of the updating process with b.s.

6.4. Decoding Complexity

From the work in [17], to encode a block by applying RLNC from d_k input blocks required $\mathcal{O}(\mathcal{T} \cdot d_k)$ finite field operation, where \mathcal{T} is the number of symbols per block. By applying Gaussian elimination for the decoding process, decoding a chunk with size d_k requires $\mathcal{O}(d_k^2 + \mathcal{T} \cdot d_k)$ finite field operations per block on average. Since decoding complexity is more significant than encoding complexity, thus only decoding complexity is discussed in this paper.

With the applied decoding scheme, the main key to look at the decoding complexity is the mean length of ch.cs, $\bar{\phi}$. With OCC/CF, it is hard to estimate $\bar{\phi}$ since a successful b.s or co.cs does not depend on the selection of $[D, \gamma]$ alone, but actually on the selection of $[(\mu_1, \gamma_1), (\mu_2, \gamma_2), \dots, (\mu_K, \gamma_K)]$. However, this paper employs the OCC designed using $\rho(r)$, i.e., the selection of $[D, \gamma]$, to estimate $\bar{\phi}$ since $p_d \geq p_{deff}$ and the obtained value of $\bar{\phi}$ might rely on the way of conducting the estimation.

$\varrho_\phi(\phi)$ denotes the probability (fraction) that a ch.cs with length ϕ is successfully decoded. $\varrho_\phi(\phi)$ for $2 \leq \phi \leq \phi_{max}$ can be estimated or collected along with the process of updating probabilities with co.cs, i.e., the increase in $\varrho_h(D)$ and in $\varrho_f(D)$ for a considered ϕ is the increase in $\varrho_\phi(\phi)$. In addition, $\varrho_\phi(1)$ can be obtained by $p_d - \sum_{\phi=2}^{\phi_{max}} \varrho_\phi(\phi)$. Hence, $\bar{\phi}$ can be obtained by:

$$\bar{\phi} = \frac{\sum_{\phi=1}^{\phi_{max}} \phi \cdot \varrho_\phi(\phi)}{p_d}. \quad (20)$$

Therefore, while employing OCC/CF, for each successfully decoding, it would need $\mathcal{O}(\bar{\phi}^2 \cdot D^2 + \mathcal{T} \cdot \bar{\phi} \cdot D)$ finite field operations per block. It would be lower since C_c is an approximately sparse matrix, as shown in Figure 7.

6.5. Obtaining the Estimated Performance

The process of updating probabilities with co.cs and b.s above can be done repeatedly and also alternately. Since there are μ innovative blocks per chunk, then chunk t with $r^{(t)} > \mu$ can help the undecodable chunks to decode. This paper considers that applying the updating process with b.s first might make the value of $q'_n(r^{(t)})$ where $r^{(t)}$ is close to D tend to zero early and then might make the other $q'_n(r^{(t)})$ hardly tend to zero via the updating process. Therefore, this paper applies the updating process with co.cs first.

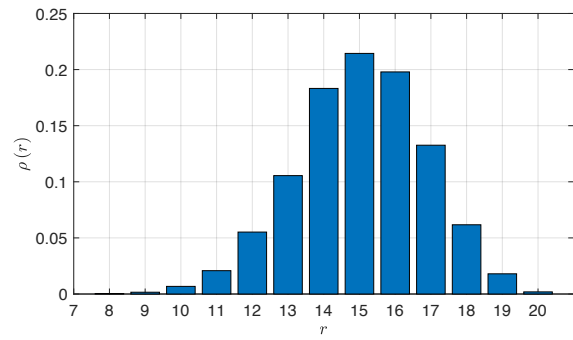
By taking the overlapping fashion of the applied OCC as the pseudo-reference, the updating process with co.cs is applied for one iteration, then the updating process with b.s is applied for ϕ_{\max} iterations. If N is the total number of original blocks, then there are approximately $N_{\text{ch}} = \left\lceil \frac{N}{\phi_{\max}} \right\rceil$ chains of chunks with length ϕ_{\max} . It is assumed that one time of the updating process is the updating process with co.cs for one iteration and then with b.s for ϕ_{\max} iterations. Then, N_{ch} times of the updating process are needed such that the effect of the first ch.cs reaches the last ch.cs, and thus, $2 \cdot N_{\text{ch}} - 1$ times of the updating process are needed such that this effect returns back to the first chunk. In this work, one round of the updating process is $2 \cdot N_{\text{ch}} - 1$ times the updating process. In order to obtain the ultimate $\rho_d(r')$, the updating process is done for a number of rounds until the increment of obtained p_d is lower than an assigned ϵ_t , then the updating process terminates, and the ultimate p_d is obtained. The channel efficient η is obtained by $\eta = \mu \cdot p_d$. The computational complexity for the estimation depends on ϵ_t . It is higher when ϵ_t is smaller, but the accuracy might be higher.

For example, by taking $K = L = 2$, $M = 10$, $\phi_{\max} = 5$, $\text{SNR}_{1,1} = \text{SNR}_{2,2} = \text{SNR}_{1,2} = 35$ dB, $\text{SNR}_{2,1} = 15$ dB, $E_8/7E_8$ as NLC and by taking only the case that $\mu - \gamma > 0$, the empirical rank distribution $\rho(r)$, the estimated p_d and the correspondent η are shown in Figure 8. The simulation result obtained with the same condition and by using the empirical rank distribution in Figure 8a is also shown in Figure 8b,c for comparison.

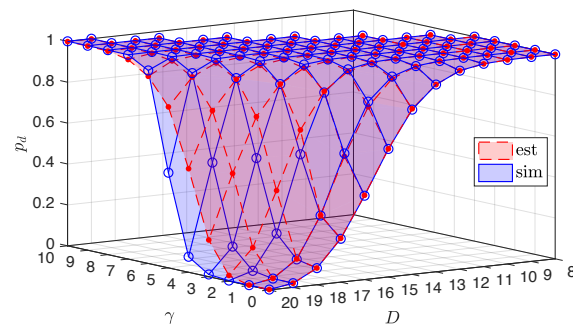
From Figure 8b, the estimation causes high deviation from the simulation result for the allocations (μ, γ) that provide low p_d , e.g., lower than 0.9, and it causes low deviation for the allocations (μ, γ) that provide high p_d . The error might be caused by the inaccuracy of the updated rank distribution ρ' or by the imperfectness of the process of updating probability with limited length of ch.cs. However, because the performance with high p_d is preferred for this paper, thus the estimation applied in this paper is acceptable.

On the other hand, from Figure 8c, the maximum η from the estimation, $\eta_{\text{max-est}}$ is obtained by taking allocation (14, 5). However, the maximum η from the simulation, $\eta_{\text{max-sim}}$ is obtained by taking allocation (14, 6). Both allocations have the same μ , but different output p_d . From the work of [14,15], without limiting the length of ch.cs, larger γ with the same μ can provide higher p_d . Due to the estimation deviation, the allocation to provide η_{max} is not correctly given. However, during the application, the allocation providing $\eta_{\text{max-est}}$ can be switched to the other allocation with larger γ , but with the same μ to find out which allocation is more appropriate.

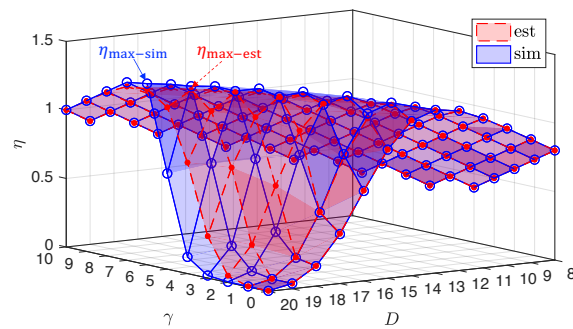
From Figure 8d, the estimated $\bar{\phi}$ is much larger than $\bar{\phi}$ from the simulation, since the process of updating probabilities with co.cs is executed first, which is different from the real fact that b.s should be conducted as soon as possible according to the decoding scheme described in Algorithm 4. Thus, $q_\phi(\phi)$ is abnormally high for $\phi \geq 2$, especially, when γ is large. However, the obtained results of $\bar{\phi}$ with the same $\mu = D - \gamma$, but with different set values (D, γ) from both estimation and simulation show that larger γ results in higher $\bar{\phi}$, hence higher decoding complexity. In addition, $\bar{\phi}$ from the estimation somehow can serve as $\bar{\phi}$ obtained in the worst case.



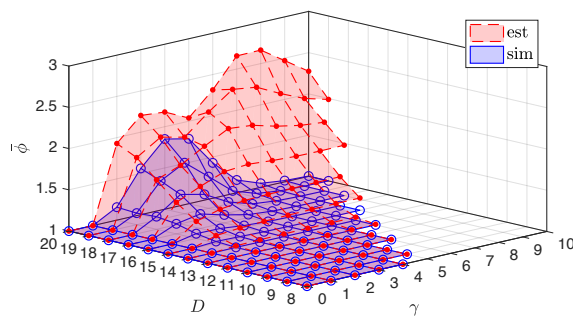
(a)



(b)



(c)



(d)

Figure 8. Comparison between the estimated p_d and η (est) and the result from the simulation (sim) that were obtained from an empirical rank distribution $\rho(r)$. (a) Empirical rank distribution $\rho(r)$; (b) decodability; (c) channel efficiency; (d) mean length of ch.cs.

7. Performance Analysis

7.1. Examples of Allocations

This paper assumes that the fairness between sources is achieved if $\eta_1 = \eta_2 = \dots = \eta_K$ where η_k is the channel efficiency for source k . η_k is defined as the ratio of the number of decoded blocks from source k to the number of time slots taken from the sources to the relays. If individually decoding is not considered, η_k for all k only depends on $p_{d\text{eff}}$. In this case, the fairness is achieved by taking $\mu_1 = \mu_2 = \dots = \mu_K$.

By taking the data in Figure 8, Table 4 lists some allocations and their performances in $p_{d\text{eff}}$ and η_{eff} from estimation and simulation. In addition, $\bar{\phi}$ is the average value of ϕ , and it is counted when a ch.cs with length $\phi \geq 1$ is decoded. $\bar{\phi}$ represents the computational complexity of the decoding process.

Table 4. Some examples of allocations.

Allocations					Estimation					Simulation			
No.	μ_1	μ_2	γ_1	γ_2	p_1	p_2	p_d	$p_{d\text{eff}}$	η_{eff}	$\bar{\phi}$	$p_{d\text{eff}}$	η_{eff}	$\bar{\phi}$
1	7	7	2	3	0.9982	0.9722	0.9206	0.8935	1.2509	2.1671	0.9107	1.2750	1.4702
2	8	6	2	3	0.9868	0.9935	0.9206	0.9026	1.2636	2.1671	0.9295	1.3013	1.4264
3	8	6	2	4	0.9868	0.9951	0.9152	0.8986	1.2580	2.3136	0.9341	1.3077	1.4820
4	6	8	3	2	0.9988	0.8309	0.9206	0.7641	1.0687	2.1671	0.6674	0.9344	1.6135
5	7	6	3	4	0.9975	0.9951	0.9756	0.9684	1.2589	2.6592	0.9991	1.2988	1.1812
6	7	6	2	4	0.9982	0.9951	0.9779	0.9714	1.2628	2.3758	0.9986	1.2982	1.1767
7	7	6	1	4	0.9989	0.9951	0.9710	0.9710	1.2623	2.0224	0.9938	1.2919	1.1679
8	7	6	0	4	0.9981	0.9951	0.9674	0.9710	1.2577	1.6220	0.9807	1.2749	1.1549
9	7	6	1	2	0.9989	0.9947	0.9697	0.9635	1.2525	1.2880	0.9833	1.2783	1.0815

From Table 4, Allocation 1 can provide the fairness between sources, but it cannot provide the highest channel efficiency, while Allocations 2 and 3 can provide the highest channel efficiency by estimation and simulation, respectively. The interchanged Allocation 3, i.e., Allocation 4, shows the affect of an unsuitable selection of (μ_2, γ_2) , which has low $p_{d\text{eff}}$. Allocations 1–4 do not provide high $p_{d\text{eff}}$. On the other hand, Allocation 5 can provide high $p_{d\text{eff}}$, but not the highest channel efficiency. However, the decoding complexity is lower since $\bar{\phi}$ is smaller. If the difference between the provided channel efficiency and the highest channel efficiency is small, then this allocation can be applied instead if lower decoding complexity is required. Allocations 5–9 have the same μ , but different γ , which varies from 7–3. They show the outcome of different values of γ to $p_{d\text{eff}}$ and $\bar{\phi}$. From the result in Table 4, larger γ provides higher $p_{d\text{eff}}$, but higher decoding complexity.

On the other hand, the highest channel efficiency can be obtained by the precoding process at each source before employing OCC. However, this might cause additional decoding complexity and latency caused from re-ordering blocks after the decoding process. This paper assumes that there is no precoding overhead, i.e., the number of required received coded blocks is equal to the number of original blocks when the maximum channel efficiency is considered.

7.2. Impact of the Participation Factor of Each Source

From now on, this paper uses the term OCC as the applied OCC, which uses the allocation providing the highest channel efficiency. In addition, OCC' refers to the applied OCC that uses the allocation providing the highest channel efficiency with condition $p_d \geq p_{\text{thr}}$ or $p_k \geq p_{\text{thr}}$. OCC/CF and OCC'/CF refer to the transmission schemes employing OCC and OCC', respectively, before NLC in a multi-source multi-relay network. The decodability condition for OCC'/CF is $p_{d\text{eff}} \geq p_{\text{thr}}$.

η_{max} denotes the maximum channel efficiency that can be provided by the applied OCC designed using $\rho(r)$. If individually decoding is not considered, from (11), the upper bound of η_{eff} is η_{max} . As mentioned above, $\bar{\eta} = \frac{\bar{\rho}}{M}$ is the channel capacity or the upper bound of the channel efficiency for

the transmission scheme employing OCC/CF from the sources to the destination. By taking $K = L = 2$, $M = 10$, $\phi_{\max} = 5$, $E_8/7E_8$ as NLC, $p_{\text{thr}} = 0.97$, $\text{SNR}_{1,1} = \text{SNR}_{2,2} = 35$ dB, $\text{SNR}_{1,2} \in \{5, 20, 35\}$ dB, $\text{SNR}_{2,1} \in \{0, 5, 10, 15, 20, 25, 30, 35\}$ dB. The performance of OCC/CF and OCC'/CF in decodability and channel efficiency from estimation (with postfix “-est”) and simulation (with postfix “-sim”) is shown in Figure 9.

From Figure 9, the estimated channel efficiencies of OCC/CF and OCC'/CF are around 97.95% and 98.34%, respectively, of those of the channel efficiency obtained from simulation on average. In addition, the gap between the channel efficiency of OCC and $\bar{\eta}$ represents the design overhead of applied OCC. From the simulation result, η_{\max} is around 87.71% of $\bar{\eta}$ on average. The design overhead of OCC/CF should be the aggregate of the design overhead using $\rho(r)$ and $\lambda_k(\theta_k)$ for all k . The channel efficiency of OCC/CF and OCC'/CF is close to η_{\max} when $\text{SNR}_{1,2}$ or $\text{SNR}_{2,1}$ is close to (as high as) $\text{SNR}_{1,1}$ or $\text{SNR}_{2,2}$. This is because, in this case, the participation factor of each source is dense around $\theta_k = d_k$. When $\lambda_k(d_k)$ is very dense, most of the allocations (μ_k, γ_k) can provide p_k close to one. Hence, the design of OCC/CF or OCC'/CF can only depend on $\rho(r)$.

In the case that $\text{SNR}_{1,2}$ and $\text{SNR}_{2,1}$ are low, the received combined coded blocks are almost plain coded blocks, i.e., $\beta_{lm}^{(i)}$ are almost in the form of unit vectors for all l and m . The design overheads of OCC/CF and OCC'/CF in this case should be the aggregate of the design overhead of OCCs using $\lambda_k(\theta_k)$ for all k . Since the participation factor of each source is not dense around $\theta_k = d_k$, the design overheads of OCC and OCC' might be high, and higher than those of the prior case.

In order to make $\lambda_k(d_k)$ denser, in addition to forcing to obtain $\beta_{lm}^{(i)}$ without zero elements at each relay, improving the diversity of the received codeword combinations by increasing the number of participating relays or equipping more antennas at relays as in work of [8,9] might be a solution.

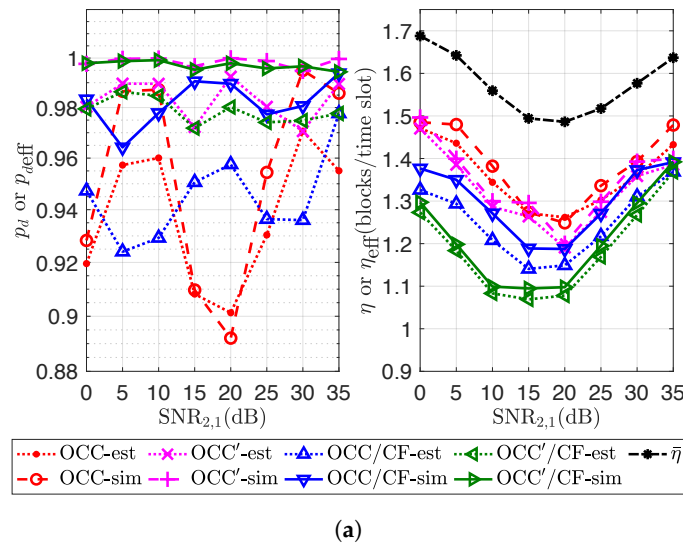
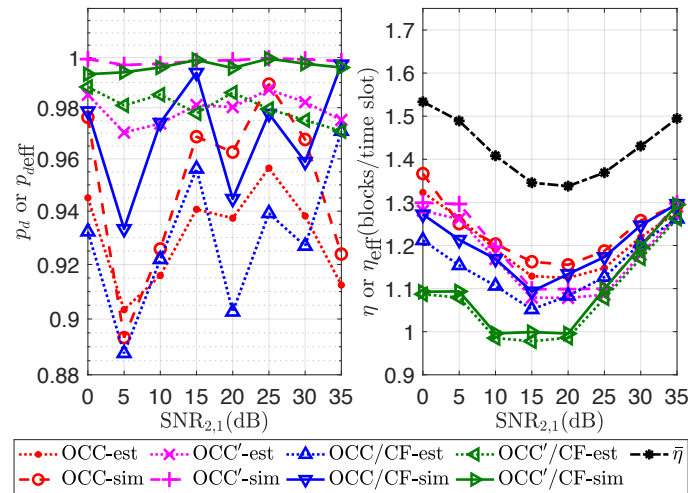
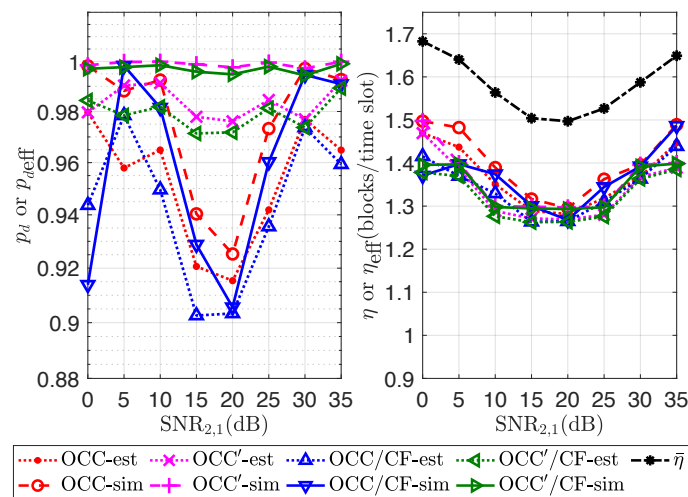


Figure 9. Cont.



(b)



(c)

Figure 9. Decodability and channel efficiency of OCC/compute-and-forward (CF) and OCC'/CF from estimation and simulation. (a) $\text{SNR}_{1,2} = 5$ dB; (b) $\text{SNR}_{1,2} = 20$ dB; (c) $\text{SNR}_{1,2} = 35$ dB.

7.3. Reference Schemes

Because the original work in [7] did not consider the retransmission, a feedback-based transmission scheme is used as the reference scheme instead to evaluate the performance of OCC/CF and OCC'/CF, and it is called CF with protocol overhead (CF/PO) in this paper. For each round of CF/PO, each source applies NLC without OCC and needs feedback from the destination after sending a block to know which blocks have been decoded and which blocks need to be retransmitted. Feedback is forwarded by relays via an orthogonal channel. There is no decoding delay constraint, i.e., a source can transmit a new original block although the previous blocks of the other sources have not been decoded [6].

The protocol overhead (the transmission time of feedback and the loss of feedback) are taken into account. The feedback reception success rate is denoted by p_f , and the ratio of the transmission time of feedback to a slot time is denoted by τ_f . p_f is obtained by conducting a simulation where NLC is applied on a link from a source to a relay with the highest SNR via an orthogonal channel. Because it is hard to track the performance of CF/PO with varying p_f , only the performance with different values of τ_f is considered.

In addition to the channel efficiency, the transmission efficiency ε_t is also considered to evaluate the performances of OCC/CF and OCC'/CF with a scheme called RLNC with orthogonal channel (RLNC/OC) where RLNC is applied before NLC at each source for the transmissions from the sources to the relays via an orthogonal channel. For each source, original blocks are grouped into disjoint chunks with M blocks per chunk. RLNC is applied within each chunk, and a feedback (ACK) is needed when a transmitted chunk is decodable. The protocol overhead is also considered, and it is assumed that feedback cannot be received instantaneously by the source to stop transmitting [32]. In this paper, ε_t is defined as the ratio of the total number of decoded blocks to the total number of transmissions taken between the sources and the relays, while the transmission of feedback is also taken into account. In this paper, τ_f is assumed as the ratio of the length of feedback data to the length of payload data per block. The performance in transmission efficiency reflects the energy consumption of each scheme.

On the other hand, a transmission scheme employing LT code [20,30] at each source before NLC, called fountain code with CF (FC/CF), is also used as a reference scheme. The considered parameters of LT code are c_{fc} and δ_{fc} . In the case of single flow transmission with N_k original blocks, the receiver can recover all blocks with probability $1 - \delta_{fc}$ if receiving $N_k + 2 \cdot \log_e(S_{fc}/\delta_{fc}) \cdot S_{fc}$ coded blocks, where $S_{fc} \equiv c_{fc} \cdot \log_e(N_k/\delta_{fc}) \cdot \sqrt{N_k}$.

In the CF/PO and FC/CF schemes, the decoding process is done for each source block transmission. The destination tries to decode if there are K linearly independent codeword combinations. If undecodable, the destination stores the undecodable blocks (after decoding using (2)) and waits for the next received codeword combinations.

Since feedback is not needed in OCC/CF, OCC'/CF and FC/FC, their transmission efficiency is $1/K$ times their channel efficiency. If N_{dec} , N_{fb} and N_{ts} denote the total number of decoded blocks, the total number of feedback and the total number of time slots taken excluding the transmission time of feedback, respectively, then the channel efficiencies and the transmission efficiencies of CF/PO and RLNC/OC can be written as below.

$$\eta_{\text{eff_CF/PO}} = \frac{N_{\text{dec}}}{N_{\text{ts}} + \tau_f \cdot N_{\text{fb}}} \quad (21)$$

$$\varepsilon_{t_CF/PO} = \frac{N_{\text{dec}}}{K \cdot N_{\text{ts}} + \tau_f \cdot N_{\text{fb}}} \quad (22)$$

$$\eta_{\text{eff_RLNC/OC}} = \frac{N_{\text{dec}}}{N_{\text{ts}}} \quad (23)$$

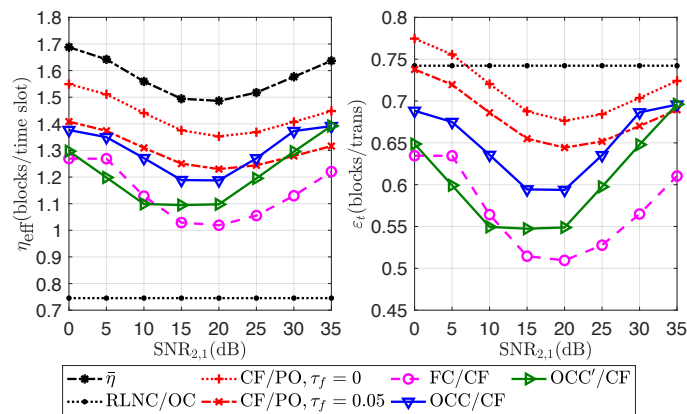
$$\varepsilon_{t_RLNC/OC} = \frac{N_{\text{dec}}}{N_{\text{ts}} + \tau_f \cdot N_{\text{fb}}} \quad (24)$$

8. Numerical Results and Discussion

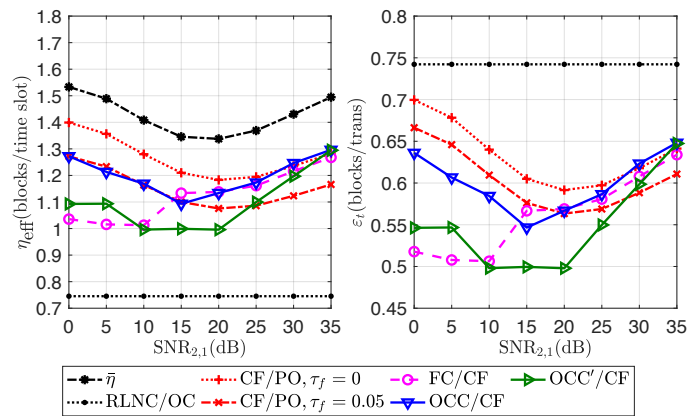
This papers considers two scenarios to observe the performance of the transmission schemes employing OCC/CF and OCC'/CF by comparing with the reference schemes. The first scenario investigates the performance in a two-source two-relay network with an asymmetric channel state at relays, i.e., average SNRs of all links from all sources to a relay might be different, as used in Section 7.2. The second scenario considers a varying number of relays with a symmetric channel state, i.e., average SNRs of all links from all sources to a relay are the same, and the number of sources is fixed to two. The numerical results are obtained by conducting simulations taking $E_8/7E_8$ as NLC, $M = 10$, $N_k = 1000$ for all k , $c_{fc} = 0.01$, $\delta_{fc} = 0.01$, $\tau_f = 0.05$ for RLNC/OC and $\tau_f \in \{0, 0.05\}$ for CF/OF. The simulation frequency is 100 times. Each simulation terminates when there is at least source k having the rest of innovative blocks less than μ_k for OCC/CF, OCC'/CF and RLNC/OC and when all original blocks of at least one source are recovered for CF/PO and FC/CF.

The performances of OCC/CF and OCC'/CF in the first scenario are the same as in Figure 9. The second scenario takes $\text{SNR}_{1,1} = \text{SNR}_{2,2} = \text{SNR}_{1,2} = \text{SNR}_{2,1} \in \{30, 35, 40\}$ dB with correspondent

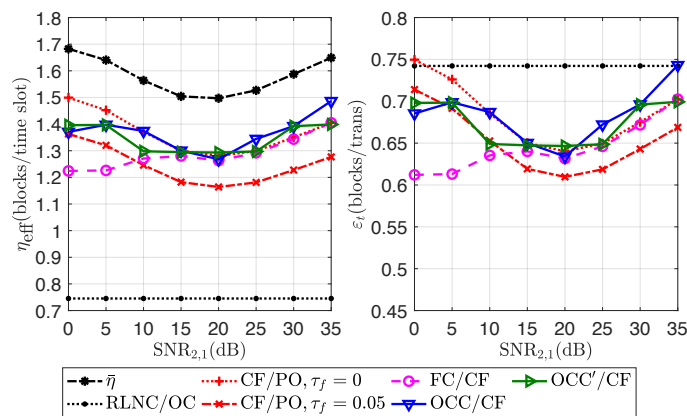
$p_f \in \{0.8266, 0.9021, 0.9449\}$. The performances in channel efficiency and transmission efficiency of all schemes in Scenario 1 and 2 are shown in Figures 10 and 11, respectively.



(a)

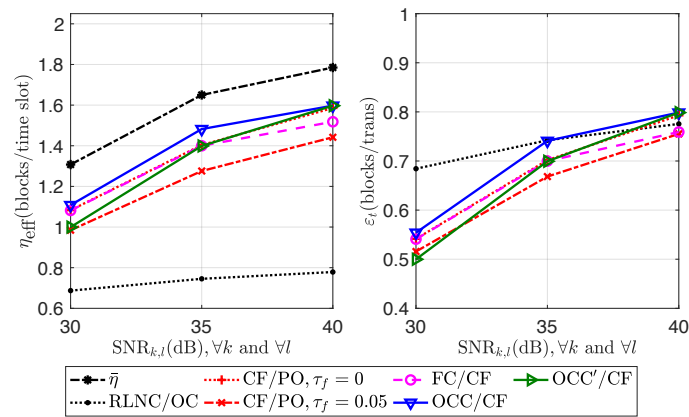


(b)

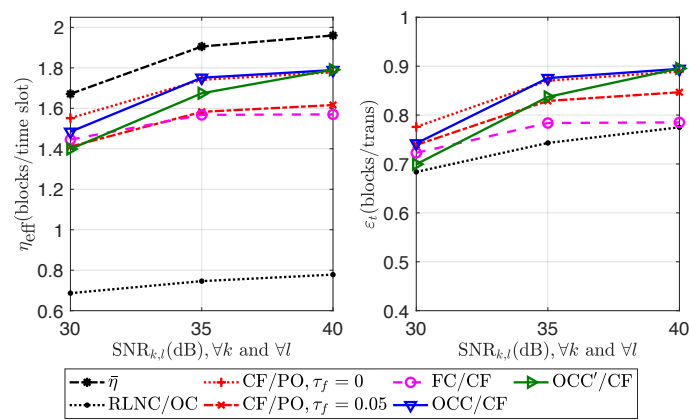


(c)

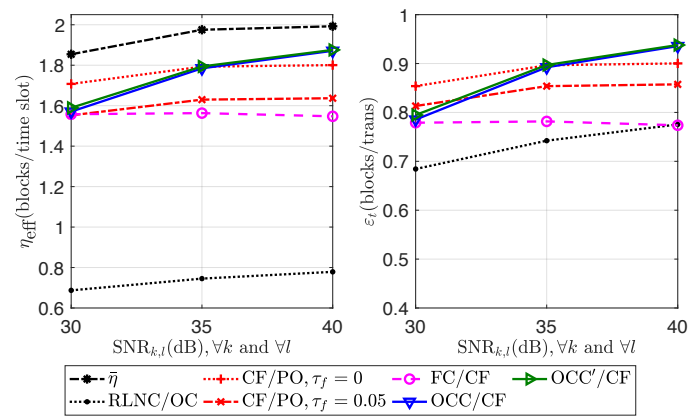
Figure 10. Channel efficiency and transmission efficiency for Scenario 1. (a) $SNR_{1,2} = 5$ dB; (b) $SNR_{1,2} = 20$ dB; (c) $SNR_{1,2} = 35$ dB.



(a)



(b)



(c)

Figure 11. Channel efficiency and transmission efficiency for Scenario 2. (a) Two-source two-relay network; (b) two-source three-relay network; (c) two-source four-relay network.

From Figure 10, OCC/CF and OCC'/CF increased channel efficiency by 72.98% and 63.28%, respectively, on average if comparing with RLNC/OC. For $\tau_f = 0.05$, OCC/CF had a 4.71% increment of channel efficiency on average if comparing with CF/PO. However, OCC/CF and OCC'/CF provided higher channel efficiency than CF/PO only when SNR_{1,2} or SNR_{2,1} was similarly high as SNR_{1,1} or SNR_{2,2}, i.e., when the participation factor of source k was dense around $\theta_k = d_k$ for all k . OCC/CF and OCC'/CF provided the increment of channel efficiency up to 16.41% and 13.41% if comparing with

CF/PO for $\tau_f = 0.05$. On the other hand, by comparing with FC/CF, OCC/CF had higher channel efficiency than FC/CF in almost all cases. The increment was around 9.36% on average. For OCC'/CF, because of the higher design overhead, it sometimes could not provide higher channel efficiency than FC/CF. There was only a 2.91% increment of channel efficiency on average if comparing with FC/CF. The issue of employing fountain code in this scenario was that the number of coded blocks to ensure the desired decodability was larger than the expected number, because some coded blocks from a source could not be extracted from the codeword combinations forwarded from the relays during each source block transmission.

On the other hand, the transmission efficiency of RLNC/OC was higher than the other schemes in all cases except CF/PO for $\tau_f = 0$. Thus, OCC/CF and OCC'/CF had less of a chance to perform better than CF/PO when τ_f was small. The transmission efficiency of OCC/CF and OCC'/CF was 86.85% and 81.98% on average, respectively, of the transmission efficiency of RLNC/OC. It seems the performance of the proposed schemes was a trade-off between the increment of channel efficiency and the decrement of transmission efficiency if comparing with an orthogonal channel transmission scheme, e.g., RLNC/OC for this scenario. In order to improve the transmission efficiency of the proposed schemes, increasing the diversity of codeword combinations at relays, i.e., increasing the number of participating relays, was considered and discussed in Scenario 2.

From Figure 11, the channel efficiencies of OCC/CF and OCC'/CF increased when $\text{SNR}_{k,l}$ increased, since the channel capacity $\bar{\rho}$ also increased with $\text{SNR}_{k,l}$. OCC/CF and OCC'/CF performed similarly at high $\text{SNR}_{k,l}$, and they had a chance to perform better than CF/PO for $\tau_f = 0$ in the case of four relays at $\text{SNR}_{k,l} = 40$ dB since the loss of feedback also impacted the performance of CF/PO. If comparing with CF/PO, OCC/CF and OCC'/CF increased channel efficiency up to 3.93% and 4.17%, respectively, for $\tau_f = 0$, and up to 16.18% and 14.59%, respectively, for $\tau_f = 0.05$. If comparing with RLNC/OC, OCC/CF and OCC'/CF increased channel efficiency up to 140.32% and 140.88%, respectively. OCC'/CF sometimes performed slightly better than OCC/CF, because of the estimation deviation. In addition, the channel efficiency of OCC/CF and OCC'/CF increased faster than that of FC/CF when $\text{SNR}_{k,l}$ increased or the number of relays increased. This is because the overhead of fountain code was the same if the parameters c_{fc} and δ_{fc} were fixed. In addition, it might have been because of the condition of stopping the simulation, i.e., all original blocks of a source were recovered, and those of the other source had not been all recovered, the channel efficiency of FC/CF did not increase when $\text{SNR}_{k,l}$ increases or the number of relays increased, as shown in Figure 11c.

For the performance in transmission efficiency in Figure 11, OCC/CF and OCC'/CF had higher transmission efficiency than RLNC/OC when the number of relays was higher than the number of sources. However, employing more relays might have increased the complexity of the network such as how to select which relays to join, how to achieve time synchronization at all relays, etc.

On the other hand, the performance of OCC/CF and OCC'/CF could be improved, especially at low $\text{SNR}_{1,2}$ and low $\text{SNR}_{2,1}$ in the first scenario by applying decoding individually, but this might have increased the complexity of the decoding process if K were large.

9. Conclusions

This paper proposed a design of OCC that is applied before NLC in multi-source multi-relay networks, called OCC/CF. A decodability condition was provided for the design. This paper took an OCC with a contiguously overlapping fashion, but not a rounded-end fashion, to design OCC/CF. The decoding scheme and the estimation of designed OCC/CF are provided. The estimation is done for each allocation, i.e., the number of innovative blocks per chunk and the number of blocks taken from the previous chunk, to search for which allocation can provide the desired performance such as the highest channel efficiency, or the preferred decodability, or the acceptable decoding complexity. The estimation deviation is low when the decodability is sufficiently high. Since there are a limited number of chunks for the designed OCC/CF, the design overhead is high if comparing with channel capacity. From the numerical results, the advantage of OCC/CF over a feedback-based transmission scheme depends

on the level of protocol overhead, i.e., the transmission time and the size of feedback, the feedback loss rate. The performance of OCC/CF, especially transmission efficiency when comparing with an orthogonal channel transmission, can be improved by increasing the number of relays. Future work is to consider decoding individually and the cooperation between feedback and OCC/CF for higher performance.

Author Contributions: R.N. proposed and worked on the research theme and wrote the paper. B.M.K. provided the source code of nested lattice codes and insightful guidance. Y.L., and Y.T. provided insightful guidance.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ngeth, R.; Lim, Y.; Kurkoski, B.M.; Tan, Y. A Design of Overlapped Chunked Code over Compute-and-Forward in Multi-Source Multi-Relay Networks. In Proceedings of the 2018 IEEE Global Communications Conference: Wireless Communications, Abu Dhabi, UAE, 9–13 December 2018.
2. Liew, S.C.; Zhang, S.; Lu, L. Physical-layer network coding: Tutorial, survey, and beyond. *Phys. Commun.* **2013**, *6*, 4–42. [[CrossRef](#)]
3. Katti, S.; Rahul, H.; Hu, W.; Katabi, D.; Médard, M.; Crowcroft, J. XORs in the air: Practical wireless network coding. *IEEE/ACM Trans. Netw.* **2008**, *16*, 497–510. [[CrossRef](#)]
4. Chachulski, S.; Jennings, M.; Katti, S.; Katabi, D. *Trading Structure for Randomness in Wireless Opportunistic Routing*; ACM: New York, NY, USA, 2007; Volume 37.
5. Lun, D.S.; Médard, M.; Koetter, R.; Effros, M. On coding for reliable communication over packet networks. *Phys. Commun.* **2008**, *1*, 3–20. [[CrossRef](#)]
6. He, J.; Liew, S.C. ARQ for physical-layer network coding. *IEEE Trans. Mob. Comput.* **2016**, *15*, 1614–1631. [[CrossRef](#)]
7. Nazer, B.; Gastpar, M. Compute-and-forward: Harnessing interference through structured codes. *IEEE Trans. Inf. Theory* **2011**, *57*, 6463–6486. [[CrossRef](#)]
8. Wei, L.; Chen, W. Compute-and-forward network coding design over multi-source multi-relay channels. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 3348–3357. [[CrossRef](#)]
9. Hejazi, M.; Azimi-Abarghouyi, S.M.; Makki, B.; Nasiri-Kenari, M.; Svensson, T. Robust Successive Compute-and-Forward Over Multiuser Multirelay Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 8112–8129. [[CrossRef](#)]
10. Chen, Z.; Fan, P.; Letaief, K.B. Compute-and-forward: Optimization over multisource–multirelay networks. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1806–1818. [[CrossRef](#)]
11. Chou, P.A.; Wu, Y.; Jain, K. Practical network coding. In Proceedings of the Annual Allerton Conference on Communication Control and Computing, Monticello, IL, USA, 1–3 October 2003; University of Illinois: Champaign, IL, USA, 2003; Volume 41, pp. 40–49.
12. Maymounkov, P.; Harvey, N.J.; Lun, D.S. Methods for efficient network coding. In Proceedings of the 44th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 27–29 September 2006; pp. 482–491.
13. Silva, D.; Zeng, W.; Kschischang, F.R. Sparse network coding with overlapping classes. In Proceedings of the 2009 Workshop on Network Coding, Theory, and Applications, Lausanne, Switzerland, 15–16 June 2009; pp. 74–79.
14. Heidarzadeh, A.; Banihashemi, A.H. Overlapped chunked network coding. In Proceedings of the 2010 IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo), Cairo, Egypt, 6–8 January 2010; pp. 1–5.
15. Heidarzadeh, A.; Banihashemi, A.H. Analysis of overlapped chunked codes with small chunks over line networks. In Proceedings of the 2011 IEEE International Symposium on Information Theory Proceedings, St. Petersburg, Russia, 31 July–5 August 2011; pp. 801–805.
16. Li, Y.; Soljanin, E.; Spasojevic, P. Effects of the generation size and overlap on throughput and complexity in randomized linear network coding. *IEEE Trans. Inf. Theory* **2011**, *57*, 1111–1123. [[CrossRef](#)]
17. Yang, S.; Yeung, R.W. Batched sparse codes. *IEEE Trans. Inf. Theory* **2014**, *60*, 5322–5346. [[CrossRef](#)]

18. Tang, B.; Yang, S. An LDPC Approach for Chunked Network Codes. *IEEE/ACM Trans. Netw.* **2018**, *26*, 605–617. [[CrossRef](#)]
19. Byers, J.W.; Luby, M.; Mitzenmacher, M.; Rege, A. A digital fountain approach to reliable distribution of bulk data. *ACM SIGCOMM Comput. Commun. Rev.* **1998**, *28*, 56–67. [[CrossRef](#)]
20. Luby, M. LT codes. In Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, Vancouver, BC, Canada, 16–19 November 2002; pp. 271–280.
21. Shokrollahi, A. Raptor codes. *IEEE Trans. Inf. Theory* **2006**, *52*, 2551–2567. [[CrossRef](#)]
22. Maymounkov, P. *Online Codes*; Technical Report; New York University: New York, NY, USA, 2002.
23. Shokrollahi, M.A.; Lassen, S.; Karp, R. Systems and Processes for Decoding Chain Reaction Codes through Inactivation. U.S. Patent 6,856,263, 2005.
24. Lázaro, F.; Liva, G.; Bauch, G. Inactivation Decoding of LT and Raptor Codes: Analysis and Code Design. *IEEE Trans. Commun.* **2017**, *65*, 4114–4127. [[CrossRef](#)]
25. Fincke, U.; Pohst, M. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Math. Comput.* **1985**, *44*, 463–471. [[CrossRef](#)]
26. Hasan, M.N.; Kurkoski, B.M. Practical compute-and-forward approaches for the multiple access relay channel. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.
27. Hejazi, M.; Nasiri-Kenari, M. Simplified compute-and-forward and its performance analysis. *IET Commun.* **2013**, *7*, 2054–2063. [[CrossRef](#)]
28. Tang, B.; Yang, S. An improved design of overlapped chunked codes. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6.
29. Ngeth, R.; Kurkoski, B.M.; Lim, Y.; Tan, Y. Random linear network coding over compute-and-forward in multi-source multi-relay networks. In Proceedings of 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017; pp. 805–810.
30. MacKay, D.J. Fountain codes. *IEE Proc. Commun.* **2005**, *152*, 1062–1068. [[CrossRef](#)]
31. Hogg, R.V.; McKean, J.; Craig, A.T. *Introduction to Mathematical Statistics*; Pearson Education: London, UK, 2005.
32. Apavatjirut, A.; Goursaud, C.; Jaffrès-Runser, K.; Gorce, J.M. Fountain codes and network coding for WSNs. In *Network Coding*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2012; pp. 27–72.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).