Wireless Telecommunications Symposium 2018 (WTS 2108)
April 18-20, 2018

Near-Instant Link Failure Recovery in 5G Wireless Fog-Based-Fronthaul Networks

Nabeel I. Sulieman, Eren Balevi, and Richard D. Gitlin
Department of Electrical Engineering
University of South Florida
Tampa, Florida 33620, USA

Email: nis@mail.usf.edu, erenbalevi@mail.usf.edu, richgitlin@usf.edu

***Abstract*—Rapid recovery from link failures was previously demonstrated via the synergistic combination of Diversity and Network Coding (DC-NC) for a wide variety of network architectures. In this paper, the DC-NC methodology is further enhanced to achieve near-instant recovery from multiple, simultaneous wireless link failures by modifying Triangular Network Coding (TNC) to create enhanced DC-NC (eDC-NC) that is applied to 5G wireless Fog computing based Radio Access Networks (F-RANs). In addition, an explicit algorithm for the eDC-NC decoding process is provided. Our results demonstrate that applying eDC-NC coding to a F-RAN fronthaul network will provide ultra-reliability, enable near-instantaneous fault recovery, and enhance the throughput by at least 20% (for three broadcasted data streams).**

Keywords—5G, Diversity Coding, F-RAN, Network Coding, reliability, throughput

I. Introduction

Several applications in 5G wireless communication systems are required to be ultra-reliable and very efficient with ultra-low latency communications [1]. This study describes a methodology for rapid recovery from link or node failures in the fronthaul networks of 5G Fog Radio Access Networks (F-RANs). F-RANs are an enhancement and an alternative to Cloud Radio Access Network (C-RAN) [2]-[4]. The key idea of F-RAN is to employ edge nodes with the ability to store data, control signals, and communicate to each other instead of centralizing processing in the baseband unit (BBU) at the C-RAN [2]-[4]. Diversity Coding (DC) [5]-[6], an open loop coding technique, can help address this challenge and is a forward error control technology over diverse routes. With DC once the failure is detected the lost message can be rapidly recovered without performing rerouting and/or retransmission.

In [7] and [8], DC is used to improve the reliability of a C-RAN network with the ability to tolerate multiple simultaneous link failures. Diversity Coding was described to improve the reliability of OFDM-based vehicular systems [9] and sensor networks [10]. Network Coding (NC) [11] has the ability to further improve 5G wireless F-RAN performance by increasing its throughput. Triangular Network Coding (TNC) [12] is another mode of NC that can be used for this purpose with less computational complexity.

DC-NC coding [13], a synergistic combination of Diversity Coding (DC) and Network Coding (NC), can simultaneously enhance wireless network reliability, provide high throughput and enable low latency 5G communications systems. DC-NC coding can be easily integrated into the-state-of-art F-RAN by deploying relay nodes that are configured to enable DC-NC coding.

The contributions of this paper are modifying TNC to enhance DC-NC coding and applying enhanced DC-NC coding (eDC-NC) to F-RAN wireless networks to improve their reliability with reduced computational complexity, provide extremely low recovery time for simultaneous multiple link failures, and to increase throughput for broadcasting or multicasting applications. In addition, an explicit algorithm for eDC-NC decoding process is presented.

The rest of this paper is organized as follows: Section II describes the network topology based on F-RANs. Section III presents a background about Triangular Network Coding. The modification to TNC and its utilization to enhance DC-NC coding and a decoding algorithm are presented in Section IV. Section V demonstrates the ability of eDC-NC coding to enable higher throughput and faster recovery from multiple simultaneous link failures in wireless fronthaul networks. The paper ends with concluding remarks in Section VI.

II. System Model

F-RANs were proposed in [2]-[4] to improve the performance of C-RANs by migrating a significant number of functions to the edge device and substantially upgrading the Remote Radio Heads (RRHs). These functions include controlling, communicating, measuring, managing, and storing data. In this way, an upgraded RRH is called a Fog Access Point (F-AP), and will be able to communicate and network with other F-APs and this architecture will reduce latency by performing functionality at the network edge rather than in the core [2]-[4]. The F-RAN architecture consists three layers as shown in Fig. 1 [2]. The BBU pool, network control, and centralized storage are the network layer functions. The access layer contains RRHs and F-APs. The terminal layer includes user equipment (UE) and Fog UE (F-UE) that access F-AP [2]-[3]. The F-APs can be formed into two topologies: a mesh topology and a tree-like topology. Both of topologies can significantly minimize the degrading effects of capacity-constrained fronthaul links [2].

Different transmission modes can be used in a F-RAN such as the C-RAN and Local Distributed Coordination (LDC) modes as illustrated in Fig. 1 [2]. The core mode for the F-RAN is the

LDC mode and the C-RAN mode is similar to that in a C-RAN, where control signals, data storage, and computing processes are centralized in the BBU pool. In LDC, the F-APs communicate with other F-APs to serve the F-UEs. These transmission modes can work together to serve both UEs and F-UEs. For example, when a UE requests data that is stored in one of the F-APs, the RRH will send its request to the BBU then the BBU instead of sending the requested data, which increases the burden on the fronthaul network, will order the F-AP to send the requested data to the UE via the RRH [2]. In this way, the burden on the fronthaul network will be decreased. Hence, the interference can be quickly suppressed, and the required data will be sent to the F-UE and UE (via RRHs) not from the cloud server but from the F-APs [2].

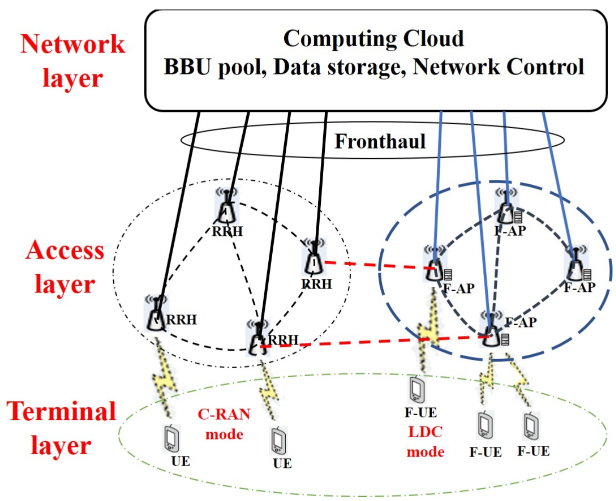

Fig. 1. F-RAN network architecture.

In this study, we apply eDC-NC coding to a mixture of the C-RAN and LDC transmission modes in a fronthaul network, where F-APs and RRHs are connected to each other in a mesh topology as shown in Fig. 2. Here, these connections are assumed to be wireless links. MIMO technology will likely be used by the F-APs and RRHs to decrease interference and be able to communicate with each other.

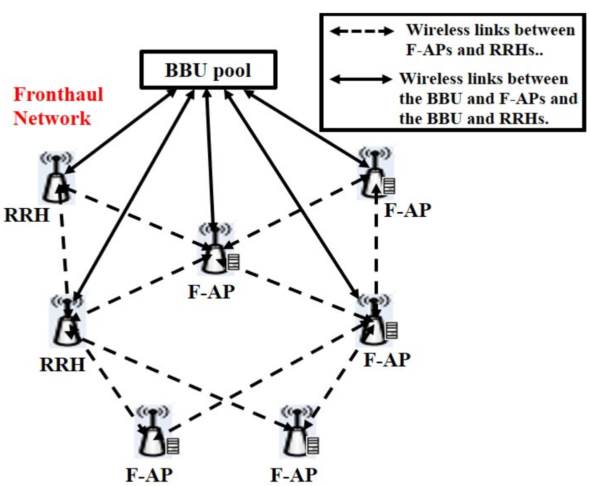

Fig. 2. Example F-RAN fronthaul network with wireless links.

III. Triangular Network Coding

Network Coding over a large finite field results in high encoding and decoding computational complexity. However, linear encoding and decoding over GF(2) can minimize computational complexity, but is not being able to generate more than one coded data stream. Triangular Network Coding (TNC) [12] is a mode of NC that has the ability to decrease the encoding and decoding computational complexity of LNC. The key idea of TNC is adding a string of "0" bit(s) on each data stream such that the XOR operation between the data streams will result a new coded data stream [12].

To illustrate the principle of TNC, let us assume that the number of data streams $N = 3$, and the data streams are d_1, d_2, and d_3. Each data stream has B bits so that the bit pattern of each data stream is $d_i = \{b_{i,1} \quad b_{i,2} \dots \dots b_{i,B}\}$, where i is the data stream number. To generate the first coded data stream, $N-1$ redundant bits "0", which is called r_{max} are required. No redundant bit "0" is added at the head of data stream d_1 and hence, it is denoted by $d_{1,0}$. A redundant bit "0" is added at the head of data stream d_2 and hence, it is denoted by $d_{2,1}$. In addition, two redundant bits "0" are added at the head of data stream d_3 and hence, it is denoted by $d_{3,2}$. To equalize the length of all data streams, Two "0" bits are added to the tail of data stream d_1 and a "0" bit is added to the tail of d_2. Therefore, in general, each data stream will be denoted by d_{i,r_i}, where r_i is the number of redundant bit(s) "0" that are added at the head of data stream i. A simple XOR operation between $d_{1,0}$, $d_{2,1}$, and $d_{3,2}$, will generate the first coded data stream, c_1. The unique ID of the encoded data stream is represented as $[r_1, r_2, r_3]$. Hence, the unique ID of c_1 is $[0, 1, 2]$, which in general is given by $[0, 1, \dots, N-1]$. To generate the second coded data stream, the position of "0" in the first ID will be fixed and all the other terms will be cyclically rotated. Hence, the second coded data stream's ID will be $[0, 2, 1]$. In this way, $N-1$ coded data streams can be generated. To generate another $N-1$ coded data streams, the position of "0" in the first ID will be changed to be in the second position such that the ID will be $[1, 0, \dots, N-1]$ and all other terms except "0" will be rotated. With N positions for "0" to be fixed, $N \times (N-1)$ coded data streams can be generated. So that in our example, $3 \times (3-1) = 6$ coded data streams can be generated. To generate another group of $N \times (N-1)$ coded data streams, the new ID will be $[0, 2, \dots, 2(N-1)]$, which is similar to the first ID with each r_i multiplied by a constant $\alpha = 2$. In similar way, another $N \times (N-1)$ coded data streams can be generated with $\alpha = 3, 4, 5, \dots$ with no limit except that higher α results in a larger redundant bits "0" [12]. It is shown in [12] that the decoding process can be easily done by bit XOR substitution. Below is a simple example to extract the required raw data streams from codes with IDs: $ID_{c_1} = [0, 1, 2]$, $ID_{c_3} = [1, 0, 2]$, $ID_{c_6} = [2, 1, 0]$. The bit representation of each code is shown in Fig. 3. Each encoded data stream is represented by a table where each row lists the bits of a data stream involved in the encoding. Starting from the left the first bit of c_1 is encoded by $b_{1,1} \oplus 0 \oplus 0$ which equals $b_{1,1}$. Similarly, $b_{2,1}$ and $b_{3,1}$ can be recovered from the first bit of c_3 and c_6 respectively. Now, the decoding process proceeds to the second bit position of the 3 coded data streams. By substituting $b_{1,1}$ into c_3 and $b_{2,1}$ into c_1

and c_6, $b_{2,2}$, $b_{1,2}$, and $b_{3,2}$ can be recovered directly. Going forward to the third bit position, bits $b_{1,3}$, $b_{2,3}$, and $b_{3,3}$ can be instantly obtained by substitution. All unknown bits can be obtained by continuing decoding process. In this way, the bits of all 3 data streams can be decoded by back substitution at the bit level [12]. However, TNC cannot work as described above if there is a raw (original) data stream in the receiver node as we will explain in detail in Section IV.

$b_{1,1}$	$b_{1,2}$	$b_{1,3}$		....	$b_{1,B}$	0	0
0	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$	....	$b_{2,B}$	0	
0	0	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$	....	$b_{3,B}$	

The ID of c_1 is $[0, 1, 2]$

0	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	....	$b_{1,B}$	0	
$b_{2,1}$	$b_{2,2}$	$b_{2,3}$		....	$b_{2,B}$	0	0
0	0	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$	....	$b_{3,B}$	

The ID of c_3 is $[1, 0, 2]$

0	0	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	....	$b_{1,B}$	
0	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$	....	$b_{2,B}$	0	
$b_{3,1}$	$b_{3,2}$	$b_{3,3}$		....	$b_{3,B}$	0	0

The ID of c_6 is $[2, 1, 0]$

Fig. 3. An example of the decoding process in TNC

IV. Enhanced DC-NC Coding

The DC-NC coding scheme depends on deterministically chosen coefficients from a finite (Galois) field [13] and the computational complexity will increase dramatically with an increased number of broadcasted data streams and/or the number of link failures that need to be protected. This will increase the link failure recovery time as it requires increasing the finite (Galois) field size. Consequently, the decoding process will consume more time as it includes matrix inversion. To address these issues, we will modify the DC-NC coding such that the encoding and decoding processes will be over GF(2), which means a simple XOR operation for coding and decoding.

In this section, the modification of TNC is explained and utilized to enhance DC-NC coding (dubbed eDC-NC). First, we will show that in the presence of a raw (i.e., uncoded in by DC or NC) data stream at the destination, which is generally the case in DC-NC networking, TNC cannot recover the other required data streams. Based on the example that we have given in section III, the first six IDs of the coded data streams will be as follows:

$$ID_{c_1} = [0,1,2], \quad ID_{c_2} = [0,2,1], \quad ID_{c_3} = [1,0,2],$$
$$ID_{c_4} = [2,0,1], \quad ID_{c_5} = [1,2,0], \quad ID_{c_6} = [2,1,0].$$

Now, let us assume that the receiver node has the raw data stream d_3 and it received c_1 and c_5. To recover d_1 and d_2, A separate XOR operation between d_3 and both received coded data streams will be performed as illustrated below:

$b_{1,1}$	$b_{1,2}$	$b_{1,3}$		....	$b_{1,B}$	0	0
0	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$	....	$b_{2,B}$	0	

Coded data stream c_1 after XOR operation with d_3

0	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	....	$b_{2,B}$	0	
0	0	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$	....	$b_{2,B}$	

Coded data stream c_5 after XOR operation with d_3.

It is clear that both data streams are similar and hence, the bit level back substitution scheme described above will not work, since only the bit $b_{1,1}$ can be obtained from both tables of coded data streams. Therefore, d_1 and d_2 cannot be recovered. Table I shows other cases that can lead to the same problematic result.

TABLE I. Other problematic cases for TNC

Available raw data stream	Coded data streams	
	First code & its ID	**Second code & its ID**
$\boldsymbol{d_1}$	c_1 [0, 1, 2]	c_4 [2, 0, 1]
$\boldsymbol{d_1}$	c_2 [0, 2, 1]	c_6 [2, 1, 0]
$\boldsymbol{d_2}$	c_2 [0, 2, 1]	c_3 [1, 0, 2]
$\boldsymbol{d_2}$	c_4 [2, 0, 1]	c_5 [1, 2, 0]
$\boldsymbol{d_3}$	c_3 [1, 0, 2]	c_6 [2, 1, 0]

TNC needs to be modified to be able to recover the required data streams when there is a raw data stream at the receiver nodes. The required data streams can be recovered when the coded data streams with a zero that is fixed in only one position in their IDs are used. However, only $(N-1)$ coded data streams can be generated with one position for "0" to be fixed. As discussed above, utilizing the same methodology used in TNC to produce another group of coded data streams, will not work with a raw data stream in the destination nodes. Therefore, to produce another group of $(N-1)$ coded data, the new ID should be [0, the smallest integer greater than r_{max} at the previous group (r_{2_α}), $r_{2_\alpha}+\alpha, \dots, r_{2_\alpha}+\alpha(N-2)$], where α represents the group number. In DC-NC coding, only $(N-1)$ coded data streams are required for throughput gain and N coded data streams are required for maximum reliability of the network. Hence, the maximum number of required coded data streams will be $(2N-1)$. For example, to broadcast 3 data streams i.e. $N=3$ and tolerate 2 link failures for each destination node, four coded data streams with the following IDs are required:

$$ID_{c_1} = [0,1,2], \quad ID_{c_2} = [0,2,1],$$
$$ID_{c_3} = [0,3,5], \quad ID_{c_4} = [0,5,3].$$

Hence, first group of coded data streams can be expressed as

$$c_1 = d_{1,0} \oplus d_{2,1} \oplus d_{3,2}, \qquad (1)$$
$$c_2 = d_{1,0} \oplus d_{2,2} \oplus d_{3,1}, \qquad (2)$$

And, second group of coded data streams can be expressed as

$$c_3 = d_{1,0} \oplus d_{2,3} \oplus d_{3,5}, \qquad (3)$$
$$c_4 = d_{1,0} \oplus d_{2,5} \oplus d_{3,3}, \qquad (4)$$

For the decoding process, although it is similar to that in TNC, we derive an algorithm and general notation for the decoding process as follows:

First: Selection of the coded data stream that will use to extract a specific raw data stream:

a. The IDs of $(N-1)$ available coded data streams at the destination node will be checked after neglecting $r_{\text{available raw data stream}}$ from there.
b. For each required raw data stream position in each coded data stream's ID, r_i will be compared. The coded data

stream with less r_i will be selected to extract the required raw data stream.

Example 1: For $N = 3$, d_1, the first raw data stream, c_1, and c_2 are available at the destination node. The IDs of c_1 is $[0, 1, 2]$ and c_2 is $[0, 2, 1]$. Now, r_1 from each ID will be neglected, then it is noted that r_2 in c_1 is less than that in c_2. Similarly, r_3 in c_2 is less than that in c_1. Hence, c_1 will use to extract d_2, the second raw data stream and c_2 will use to extract d_3, the third raw data stream.

c. In case of r_i, r_{i-1}, and so on in the same coded data stream are less than those in the second (others) coded data stream(s), the results of differences between r_i in coded data streams will decide which code will use to extract which raw data stream. The larger difference of r_i indicates that i^{th} data stream will be extracted from the coded data stream that has less r_i.

Example 2: For $N = 3$, d_1, c_1, and c_3 are available at the destination node. The IDs of c_1 is $[0, 1, 2]$ and c_3 is $[0, 3, 5]$. Now, r_1 from each ID will be neglected. It is noted that r_2 and r_3 in c_1 have less values than those in c_2. Hence, the difference of r_2 in c_1 and c_3 is calculated. Similarly, the difference of r_3 in c_1 and c_3 is calculated.

$$|r_2 \; in \; c_1 - r_2 \; in \; c_3| = |1 - 3| = 2,$$
$$|r_3 \; in \; c_1 - r_3 \; in \; c_3| = |2 - 5| = 3.$$

So that r_3 has larger difference and c_1 has the less r_3. Hence, c_1 will use to extract d_3 and c_2 will use to extract d_2.

Second: After selecting the coded data streams to decode the raw data streams. The required raw data stream is extracted using **the general decoding notation** as follows:

$$b_{i,k} = c_{s,(k+r_i \; in \; c_s)} \oplus b_{m,(k+(r_i-r_m) \; in \; c_s)} \oplus b_{l,(k+(r_i-r_l) \; in \; c_s)} \oplus \cdots \quad (5)$$

where $b_{i,k}$ is the bit k of raw data stream d_i, c_s is the selected coded data stream, $b_{m,(k+\cdots)}$, $b_{l,(k+\cdots)}$, and so on (based on the number of broadcasted data streams) are the known bits from other raw data streams. For $N = 3$ with one available raw data stream at destination node, the decoding processes are expressed in Table II while the decoding processes with *no* raw data stream at destination node, are expressed in Table III.

TABLE II. ENHANCED DC-NC DECODING SCHEME WITH ONE RAW DATA STREAM AT DESTINATION NODE

Available d_i	Coded data: 1st code & its ID	Coded data: 2nd code & its ID	Raw data streams after decoding (bit level)
d_1	c_1 [0, 1, 2]	c_2 [0, 2, 1]	$b_{2,k} = c_{1,(k+1)} \oplus b_{1,(k+1)} \oplus b_{3,(k-1)}$ $b_{3,k} = c_{2,(k+1)} \oplus b_{1,(k+1)} \oplus b_{2,(k-1)}$
d_1	c_1 [0, 1, 2]	c_3 [0, 3, 5]	$b_{2,k} = c_{3,(k+3)} \oplus b_{1,(k+3)} \oplus b_{3,(k-2)}$ $b_{3,k} = c_{1,(k+2)} \oplus b_{1,(k+2)} \oplus b_{2,(k+1)}$
d_1	c_1 [0, 1, 2]	c_4 [0, 5, 3]	$b_{2,k} = c_{1,(k+1)} \oplus b_{1,(k+1)} \oplus b_{3,(k-1)}$ $b_{3,k} = c_{4,(k+3)} \oplus b_{1,(k+3)} \oplus b_{2,(k-2)}$
d_1	c_2 [0, 2, 1]	c_3 [0, 3, 5]	$b_{2,k} = c_{3,(k+3)} \oplus b_{1,(k+3)} \oplus b_{3,(k-2)}$ $b_{3,k} = c_{2,(k+1)} \oplus b_{1,(k+1)} \oplus b_{2,(k-1)}$
d_1	c_2 [0, 2, 1]	c_4 [0, 5, 3]	$b_{2,k} = c_{2,(k+2)} \oplus b_{1,(k+2)} \oplus b_{3,(k+1)}$ $b_{3,k} = c_{4,(k+3)} \oplus b_{1,(k+3)} \oplus b_{2,(k-2)}$
d_1	c_3 [0, 3, 5]	c_4 [0, 5, 3]	$b_{2,k} = b_{3,(k+3)} \oplus b_{1,(k+3)} \oplus b_{3,(k-2)}$ $b_{3,k} = c_{4,(k+3)} \oplus b_{1,(k+3)} \oplus b_{2,(k-2)}$
d_2	c_1 [0, 1, 2]	c_2 [0, 2, 1]	$b_{1,k} = c_{1,k} \oplus b_{2,(k-1)} \oplus b_{3,(k-2)}$ $b_{3,k} = c_{1,(k+2)} \oplus b_{2,(k-1)} \oplus b_{1,(k+1)}$
d_2	c_1 [0, 1, 2]	c_3 [0, 3, 5]	$b_{1,k} = c_{3,k} \oplus b_{2,(k-3)} \oplus b_{3,(k-5)}$ $b_{3,k} = c_{1,(k+2)} \oplus b_{2,(k+1)} \oplus b_{1,(k+2)}$
d_2	c_1 [0, 1, 2]	c_4 [0, 5, 3]	$b_{1,k} = c_{4,k} \oplus b_{2,(k-5)} \oplus b_{3,(k-3)}$ $b_{3,k} = c_{1,(k+2)} \oplus b_{2,(k+1)} \oplus b_{1,(k+2)}$
d_2	c_2 [0, 2, 1]	c_3 [0, 3, 5]	$b_{1,k} = c_{3,k} \oplus b_{2,(k-3)} \oplus b_{3,(k-5)}$ $b_{3,k} = c_{2,(k+1)} \oplus b_{2,(k-1)} \oplus b_{1,(k+1)}$
d_2	c_2 [0, 2, 1]	c_4 [0, 5, 3]	$b_{1,k} = c_{4,k} \oplus b_{2,(k-5)} \oplus b_{3,(k-3)}$ $b_{3,k} = c_{2,(k+1)} \oplus b_{2,(k-1)} \oplus b_{1,(k+1)}$
d_2	c_3 [0, 3, 5]	c_4 [0, 5, 3]	$b_{1,k} = c_{3,k} \oplus b_{2,(k-3)} \oplus b_{3,(k-5)}$ $b_{3,k} = c_{4,(k+3)} \oplus b_{2,(k-2)} \oplus b_{1,(k+3)}$
d_3	c_1 [0, 1, 2]	c_2 [0, 2, 1]	$b_{1,k} = c_{2,k} \oplus b_{2,(k-2)} \oplus b_{3,(k-1)}$ $b_{2,k} = c_{1,(k+1)} \oplus b_{1,(k+1)} \oplus b_{3,(k-1)}$
d_3	c_1 [0, 1, 2]	c_3 [0, 3, 5]	$b_{1,k} = c_{3,k} \oplus b_{2,(k-3)} \oplus b_{3,(k-5)}$ $b_{2,k} = c_{1,(k+1)} \oplus b_{1,(k+1)} \oplus b_{3,(k-1)}$
d_3	c_1 [0, 1, 2]	c_4 [0, 5, 3]	$b_{1,k} = c_{4,k} \oplus b_{2,(k-5)} \oplus b_{3,(k-3)}$ $b_{2,k} = c_{1,(k+1)} \oplus b_{1,(k+1)} \oplus b_{3,(k-1)}$
d_3	c_2 [0, 2, 1]	c_3 [0, 3, 5]	$b_{1,k} = c_{3,k} \oplus b_{2,(k-3)} \oplus b_{3,(k-5)}$ $b_{2,k} = c_{2,(k+2)} \oplus b_{1,(k+2)} \oplus b_{3,(k+1)}$
d_3	c_2 [0, 2, 1]	c_4 [0, 5, 3]	$b_{1,k} = c_{4,k} \oplus b_{2,(k-5)} \oplus b_{3,(k-3)}$ $b_{2,k} = c_{2,(k+2)} \oplus b_{1,(k+2)} \oplus b_{3,(k+1)}$
d_3	c_3 [0, 3, 5]	c_4 [0, 5, 3]	$b_{1,k} = c_{4,k} \oplus b_{2,(k-5)} \oplus b_{3,(k-3)}$ $b_{2,k} = c_{3,(k+3)} \oplus b_{1,(k+3)} \oplus b_{3,(k-2)}$

TABLE III. ENHANCED DC-NC DECODING SCHEME WITH NO RAW DATA STREAM AT DESTINATION NODES

Coded data: 1st code & its ID	Coded data: 2nd code & its ID	Coded data: 3rd code & its ID	Raw data streams after decoding
c_1 [0, 1, 2]	c_2 [0, 2, 1]	c_3 [0, 3, 5]	$b_{1,k} = c_{3,k} \oplus b_{2,(k-3)} \oplus b_{3,(k-5)}$ $b_{2,k} = c_{1,(k+1)} \oplus b_{1,(k+1)} \oplus b_{3,(k-1)}$ $b_{3,k} = c_{2,(k+1)} \oplus b_{1,(k+1)} \oplus b_{2,(k-1)}$
c_1 [0, 1, 2]	c_2 [0, 2, 1]	c_4 [0, 5, 3]	$b_{1,k} = c_{4,k} \oplus b_{2,(k-5)} \oplus b_{3,(k-3)}$ $b_{2,k} = c_{1,(k+1)} \oplus b_{1,(k+1)} \oplus b_{3,(k-1)}$ $b_{3,k} = c_{2,(k+1)} \oplus b_{1,(k+1)} \oplus b_{2,(k-1)}$
c_1 [0, 1, 2]	c_3 [0, 3, 5]	c_4 [0, 5, 3]	$b_{1,k} = c_{3,k} \oplus b_{2,(k-3)} \oplus b_{3,(k-5)}$ $b_{2,k} = c_{1,(k+1)} \oplus b_{1,(k+1)} \oplus b_{3,(k-1)}$ $b_{3,k} = c_{4,(k+3)} \oplus b_{1,(k+3)} \oplus b_{2,(k-2)}$
c_2 [0, 2, 1]	c_3 [0, 3, 5]	c_4 [0, 5, 3]	$b_{1,k} = c_{4,k} \oplus b_{2,(k-5)} \oplus b_{3,(k-3)}$ $b_{2,k} = c_{3,(k+3)} \oplus b_{1,(k+3)} \oplus b_{3,(k-2)}$ $b_{3,k} = c_{2,(k+1)} \oplus b_{1,(k+1)} \oplus b_{2,(k-1)}$

In Table II and Table III, for $b_{i,(k-a)}$, where a is any number between 0 and r_{max},

$$b_{i,(k-a)} = 0 \qquad 0 > (k-a) > B \qquad (6)$$

In this way, eDC-NC coding can provide maximum reliability, ultra-low recovery time with minimal computational complexity, and high throughput.

Table IV shows the performance differences between eDC-NC and DC-NC.

TABLE IV. THE COMPARISON BETWEEN ENHANCED DC-NC AND REGULAR DC-NC

Criteria	eDC-NC	DC-NC
Encoding and decoding complexity	Less and same for any # of coded data	High and increases with increasing the # of coded data
Decoding scheme	bit by bit XOR substitution	Matrix inversion

Failed data recovery	Near-instant	Fast but decreases with increased number of coded data streams

V. Applying Enhanced DC-NC Coding to F-RANs

Wireless link capacity limitations and failures can cause degradation in throughput and reliability for 5G wireless fronthaul F-RANs. Enhanced DC-NC is a promising technology to maximize the reliability of F-RAN fronthaul networks with enabling ultra-low recovery time from link/node failures and increase throughput (in broadcast and other applications).

The application of eDC-NC coding to a mixing of the C-RAN and Local Distributed Coordination (LDC) transmission modes in F-RAN network is illustrated in Fig. 4. Here, wireless links are connecting F-APs and RRH to each other in a mesh topology, where each fronthaul link is bi-directional. To model the application of broadcasting three data streams from the BBU pool and two F-APs to one RRH and one F-AP, A multipoint-to-multipoint network topology are considered. With eDC-NC, five disjoint paths are needed to broadcast three data streams from the BBU pool and two F-APs to two destination nodes. Utilizing direct links, data streams d_2 and d_3 are sent from F-AP2 and BBU pool to F-AP6 and RRH1 respectively. In addition, F-AP3 receives data streams d_1, d_2, and d_3 and form coded data c_1, c_2, c_3 and c_4 as shown in (1)-(4) then sent them to F-AP4, F-AP5, F-AP7 and F-AP8 respectively. F-AP4 and F-AP5 send c_1 and c_2 respectively to RRH1 and F-AP6. Coded data streams $c_1$and c_2 in addition to data stream $d_1$are decoded in RRH1 to obtain d_2 and d_3 as described in Section IV and shown in Table II. Similarly, c_1, c_2 and d_3 are decoded in F-AP6 to get d_1 and d_2. The throughput gains in this example network improve by at least 20% [13].

Wireless fronthaul network reliability can be improved by transmitting c_3 and c_4 from F-AP7 and F-AP8 respectively to RRH1 and F-AP6. The coded data c_3 and c_4 will be ignored when there is no link failure. In the presence of a link failure, for example if the link from the F-AP2 to F-AP6 fails, F-AP6 detects the failure then recovers d_1, d_2 and d_3 by utilizing c_1, c_2 and c_3 using the decoding algorithm that described in Section IV and shown in Table III. In addition, if c_1 is lost, F-AP6 has d_2, c_2 and c_3 then can quickly and easily recover d_1 and d_3. Furthermore, if two link failures at F-AP6 are happened, for example d_1 and c_2, F-AP6 detects the failures then recovers d_1, d_2 and d_3 by utlizing c_1, c_3 and c_4 as illustrated in Table III. Similarly, any two link failures can be recovered in the same way. Note that the BBU can transmit d_3 directly to F-AP6 using the direct link between them instead of sending it to F-AP3, but in this case, d_3 will not be recoverable since it will be not included in coded data streams. In addition, the BBU pool can send d_3 to both destination nodes RRH1 and F-AP6 and to F-AP3. However, this will increase the burden on the fronthaul network, while the F-RAN was introduced to decrease fronthaul complexity. As shown in Fig. 4, only two links from the BBU pool to the RRH1 and F-AP3 are enough to transmit the required data stream and make it recoverable.

Moreover, not only multiple link failures can be recovered by eDC-NC coding in this example fronthaul network. Two intermediate node failures such as F-AP4 and F-AP5 can be tolerated since this corresponds to four simultaneous link failures that each pair is associated with different destination node. Also, when failing occurs on F-AP7 and F-AP8, c_3 and c_4 will be lost i.e. protection of the network but, if this is the only failures, successful data communication can still be achieved. In general, eDC-NC networks can tolerate n link failures for each destination node at k destination nodes, however, $kn + n$ redundant links are required, where $n \leq N$.

There is no need to simulate the results in this paper because the link failure is taken into account regardless of the failure reason and it is illustrated mathematically how the F-RAN network can be enhanced and protected by eDC-NC.

Although in this study, we solely focused on applying the eDC-NC coding scheme in a F-RAN with a wireless fronthaul network, our future work will investigate the application of this approach to more general and complex network topologies.

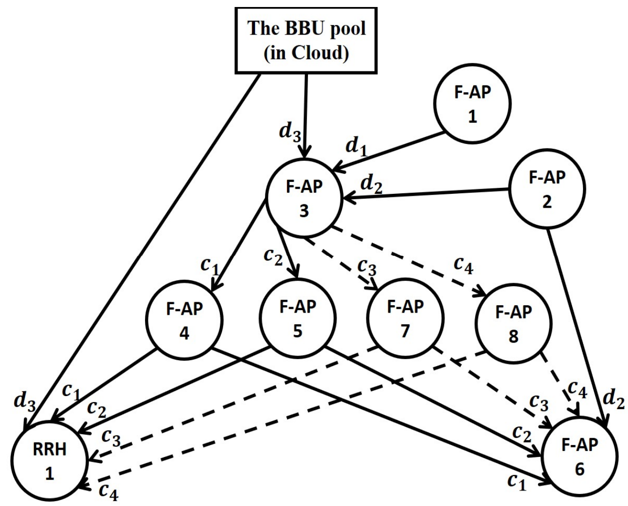

Fig. 4. Example of eDC-NC coding applied to a 5G wireless fronthaul F-RAN, where solid lines represent the links that carry coded data streams and are used to improve network throughput whereas dashed lines represent the links that carry coded data streams and are used to maximize network reliability.

VI. Conclusions

Enhanced DC-NC, which synergistically combines Diversity and modified Triangular Network Coding, is introduced in this paper to improve the performance of 5G wireless fronthaul F-RANs. It is illustrated that eDC-NC can simultaneously recover from multiple link/node failures nearly-instantaneously. In addition, an algorithm and a general notation for the eDC-NC decoding process are presented. Furthermore, eDC-NC networks can tolerate n link failures, where $n \leq$ number of broadcasted data streams for each receiver node at k receiver nodes, with $kn + n$ redundant links. Applying eDC-NC coding minimizes the impact on latency of multiple link/node failures in wireless fronthaul network links, while simultaneously improving the throughput in the network by at least 20% for three broadcasted data streams.

Acknowledgment

Nabeel I. Sulieman is supported by the Higher Committee for Education Development in Iraq (HCED-IRAQ).

References

[1] H. Tullberg *et al.*, "The METIS 5G system concept: meeting the 5G requirements," *IEEE Commun. Mag.*, vol. 54, no. 12, Dec. 2016.

[2] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46-53, Jul./Aug. 2016.

[3] M. Peng and K. Zhang, "Recent advances in fog radio access networks: performance analysis and radio resource allocation," *IEEE Access*, vol. 4, pp. 5003-5009, Aug. 2016.

[4] T. Chiu, W. Chung, A. Pang, Y. Yu, and P. Yen, "Ultra-low latency services provision in 5G fog-radio access networks," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Valencia, Spain, Sep. 2016.

[5] E. Ayanoglu, C.-L. I, R. D. Gitlin, and J. E. Mazo, "Diversity Coding: Using error control for self-healing in communication networks," in Proc. *IEEE INFOCOM'90*, San Francisco, CA, June 1990, pp. 95–104, vol. 1.

[6] E. Ayanoglu, C.-L. I, R. D. Gitlin, and J. E. Mazo, "Diversity Coding for transparent self-healing and fault-tolerant communication networks," IEEE Trans. Commun., vol. 41, pp. 1677–1686, Nov. 1993.

[7] N. Sulieman, K. Davaslioglu, and R. D. Gitlin, "Link failure recovery via Diversity Coding in 5G fronthaul wireless networks," *IEEE Wireless and Microwave Technology Conference,* Cocoa Beach, FL, April 2017.

[8] N. Sulieman, K. Davaslioglu, and R. D. Gitlin, "Diversity coded 5G fronthaul wireless networks," *IEEE Wireless Telecommunication Symposium,* Chicago, IL, April 2017.

[9] G. E. Arrobo and R. D. Gitlin, "Improving the performance of OFDM-based vehicular systems through Diversity Coding," *Journal of Commu. And Netw.,* vol. 15, no. 2, pp. 132-141, April 2013.

[10] G. E. Arrobo and R. D. Gitlin, "Minimizing energy consumption for cooperative Network and Diversity coded sensor networks," *IEEE Wireless Telecommunication Symposium,* Washington, D.C., April 2014.

[11] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," IEEE Trans. Inform. Theory, vol. 46, no. 4, July 2000.

[12] J. Qureshi, C. H. Foh, and J. Cai, "Optimal solution for the Index Coding problem using Network Coding over GF(2)," *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Seoul, South Korea, June 2012.

[13] N. I. Sulieman, E. Balevi, K. Davaslioglu, and R. D. Gitlin, "Diversity and Network coded 5G fronthaul wireless networks for ultra reliable and low latency communications," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Oct. 2017.