Delay-Efficient and Reliable Data Relaying in Ultra Dense Networks using Rateless Codes

Luyao Shang, Morteza Hashemi, Taejoon Kim, and Erik Perrins
Department of Electrical Engineering & Computer Science
University of Kansas, Lawrence, KS 66045
E-mails: {lshang, mhashemi, taejoonkim, esp}@ku.edu

***Abstract*—We investigate the problem of delay-efficient and reliable data delivery in ultra-dense networks (UDNs) that constitute macro base stations (MBSs), small base stations (SBSs), and mobile users. Considering a two-hop data delivery system, we propose a *partial decode-and-forward (PDF)* relaying strategy together with a simple and intuitive amicable encoding scheme for rateless codes to significantly improve user experience in terms of end-to-end delay. Simulation results verify that our amicable encoding scheme is efficient in improving the intermediate performance of rateless codes. It also verifies that our proposed PDF significantly improves the performance of the decode-and-forward (DF) strategy, and that PDF is much more robust against channel degradation. Overall, the proposed strategy and encoding scheme are efficient towards delay-sensitive data delivery in the UDN scenarios.**

I. Introduction

Within the last few years, we have witnessed a dramatic growth in the number of connected wireless devices that generate huge amounts of mobile data traffic. According to the Cisco Visual Networking Index (VNI), the global mobile data traffic was 0.24 exabytes (EB) per month in 2010. This number has grown to 40.77 EB per month so far in 2020, nearly a 170-fold increase over the past decade. To better serve this explosive demand in wireless data, ultra-dense network (UDN) has become one of the key enabling technologies in the fifth generation (5G) systems since it significantly improves the overall throughput by deploying small cells near the users [1, 2]. The small cells include relatively low-power base stations that cover geographical areas within a short distance from users.

As shown in Fig. 1, the UDN constitutes the macro base station (MBS), small base stations (SBS), and users. The communications between MBS and users are supported by SBSs, forming a two-hop relay system. Popular relaying strategies such as decode-and-forward (DF) and amplify-and-forward (AF) are commonly adopted in multi-hop relay systems. The AF outperforms the DF in terms of latency but the DF provides higher reliability. On the other hand, to enhance reliability under unknown channel conditions, rateless (fountain) codes [3–6], such as Luby transform (LT) codes [7] and Raptor codes [8, 9], can be deployed. Rateless codes are capacity achieving for large block lengths, and are widely considered for multi-hop relay systems [10, 11]. In contradistinction with fixed-rate codes, rateless codes do not assume a pre-defined coding rate. Instead, the encoder potentially generates an unlimited number of codewords until decoding is successful.

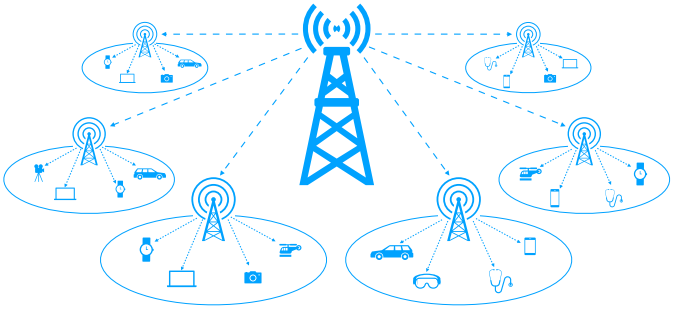

Fig. 1. Illustration of the downlink communications of an ultra-dense network. The network includes a MBS and multiple SBSs with a number of users. The SBSs are densely deployed within a short distance from the users. The MBS fetches data from the content provider through the core network router, and delivers it to the users that are served by the SBSs via a two-hop link that operates on orthogonal spectra.

Rateless codes are particularly beneficial for broadcasting/multicasting applications where a large number of receivers (e.g., massive machine-type and IoT communications) hinder channel estimation that can cause the feedback explosion problem. Yet, the problem with traditional fountain codes is that they show an *all-or-nothing* decoding property such that the decoder only recovers a small portion of source messages until the very end. This performance is not desirable for real-time and delay-sensitive data delivery such as the scalable video streaming where the video quality is improved progressively as data arrives.

In order to improve the intermediate performance metric (i.e., number of recovered symbols as coding proceeds), there has been an extensive amount of research such as [12–19]. These works mostly rely on either *feedback channels* or encoding/decoding operations with *higher complexities*, none of which is favorable to UDNs since: (i) due to the massive number of devices, leveraging the feedback channel may not be feasible, and (ii) low-power SBS, IoT, and sensory devices are not well-suited for high complexity encoding and decoding operations. Recently, memory-based LT encoders (MBLTEs) [20–24] were proposed for improving the bit-error-rate (BER)/frame-error-rate (FER) performance of LT codes with relatively short block-lengths at the cost of adding memory into the encoder while maintaining the same low encoding/decoding complexity as LT codes.

Our work is motivated by the UDN use-cases that call for

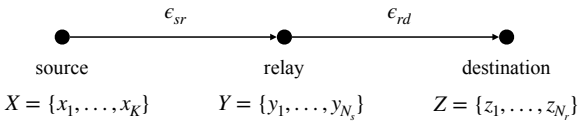

Fig. 2. The separated two-hop relay system. ϵ_{sr} and ϵ_{rd} denotes the erasure probability for the source-relay and the relay-destination link, respectively. The source node encodes K input symbols $x_1, \ldots, x_K$ to N_s output symbols $y_1, \ldots, y_{N_s}$; the relay node collects these output symbols then decodes, it then re-encodes the K input symbols to N_r output symbols $z_1, \ldots, z_{N_r}$ and forwards them to the destination; the destination node collects the output symbols then decodes.

delay-efficient data delivery with little-to-no coding feedback overhead and low-complexity encoding and decoding operations. To this end, we first demonstrate that memory-based LT codes improve the intermediate performance without utilizing feedback signals. Thus, they are especially suitable for UDN scenarios where feedback explosions and power/complexity are more of a concern than the amount of buffers/memory. Next, we propose a new relaying strategy based on DF for the SBSs in the UDN such that the real-time delay of data delivery to the users can be significantly improved. In addition, we propose an amicable rateless coding scheme that further improves the intermediate performance at the cost of adding temporary buffers. We present simulation results to verify that our proposed relaying strategy and the coding scheme are efficient in improving user experience in terms of real-time data delivery in UDNs.

The rest of the paper is organized as follows. Section II describes the system model including the two-hop relay system, LT codes, and second-order MBLTEs. Section III describes the new relaying strategy proposed for the small cells in the UDN. Section IV describes the amicable encoding scheme for second-order MBLTEs. Section V presents simulation results, and Section VI concludes the paper.

II. System Model

A. Two-Hop Relay System

In this paper, we consider the three-node two-hop wireless relay system for lossy packet networks as shown in Fig. 2. The system constitutes a source, a relay, and a destination. Two independent links, the source-relay and the relay-destination, are considered in the system. We do not consider the direct link where the source communicates directly with the destination. We term this setting *separated two-hop relay*, which is the simplest multi-hop communication problem and is frequently encountered in practice; for instance, in UDN scenarios, the source represents the MBS, the relay represents the SBS, and the destination represents the user.

We assume binary erasure channels (BECs) for the links. The packet/symbol that goes through a BEC is either received correctly or erased by the channel. Let X and Y denotes the channel input and output, respectively, then $X \in \{0, 1\}$ and $Y \in \{0, 1, e\}$, where e indicates an erasure, which occurs with probability ϵ.

For the traditional decode-and-forward (DF) relaying strategy, the communication is divided into two stages. In the first stage, the source encodes and transmits codewords to the relay. The relay starts decoding when it receives all of the codewords. It sends back an acknowledgement (ACK) to the source to terminate the transmission when decoding is successful. In the second stage, the relay re-encodes and transmits the codewords to the destination. The destination starts decoding when all codewords are received, and sends back an ACK to the relay in the event of a successful decoding. The DF achieves Shannon capacity of the separated two-hop relay system when the codeword length tends to infinity [25, 26]. However, DF is significantly suboptimal for finite block-length codes.

B. LT Codes

Encoding: For binary source data of length K, a K-by-N generator matrix encodes the K input symbols $x_1, x_2, \ldots, x_K$ to N output symbols $y_1, y_2, \ldots, y_N$. The operation is performed with bitwise, modulo-2 additions, i.e.:

$$y_n = \sum_{k=1}^{K} x_k G_{kn}, \tag{1}$$

where G_{kn} is an element of the generator matrix. The generator matrix $\mathbf{G}$ is obtained column by column. Each column represents a relationship between the input symbols and an output symbol. To obtain the output symbols, we sample the output symbol distribution Ω a total of N times. The process for each sample is summarized below [6]:

1) Randomly choose the degree d from a degree distribution Ω.
2) Uniformly choose d rows at random in the n-th column in which to place a value of "1".

Decoding: The peeling decoder is used to decode LT codes over the BEC. Assume the decoder receives all N output symbols. The decoding process is summarized below [6]:

1) Find an output symbol y_n that is connected to only one input symbol x_k. If there is no such output symbol, the decoding process stops. If there is, go to Step 2.
2) Set the input symbol equal to the output symbol: x_k=y_n.
3) Find other output symbols y_i that are connected to x_k.
4) Add x_k to each y_i modulo 2.
5) Remove all the edges connected to x_k.
6) Return to Step 1.

Next, we describe the memory-based LT encoding (MBLTE) scheme.

C. Second-Order MBLTE

As outlined above, the LT encoder works by first sampling the degree distribution Ω to obtain a degree of d, and then sampling the input symbols a total of d times. For a memoryless LT encoder, the current value of d is all that is needed to proceed with sampling the input symbols. For a MBLTE, certain previous outcomes of d are remembered which results in the input symbols being sampled differently. The order of the memory refers to the number of outcome types it remembers.

Algorithm 1 The MBLTE encoding process.

```
Initialize an empty set S_1
for n = 1 : N do
    sample Ω to get d
    if d = 1 then
        select the input symbol with the highest instantaneous degree without replacement, then put this input symbol into set S_1
    else if d = 2 then
        first select an input symbol from the set S_1 uniformly at random with replacement; then select an input symbol with the highest instantaneous degree except for those in set S_1 without replacement
    else
        for kk = 1 : d do
            select an input symbol uniformly at random without replacement
        end for
    end if
end for
```

For instance, remembering i outcome types is called a i^{th}-order MBLTE. For the sake of exposition, we only consider the second-order MBLTE in this paper. In the remainder of this paper, the term *MBLTE* represents the second-order MBLTE unless otherwise stated.

The MBLTE remembers two outcomes with values $d = 1$ and $d = 2$. When $d = 1$, it samples the input symbol with the highest instantaneous degree, which is defined in [20, 21] as the highest degree at the time when the current output symbol is being constructed. When $d = 2$, it samples one input symbol from those with the highest instantaneous degrees, then another one from those symbols with the second highest instantaneous degree. The details are shown in Algorithm 1 [22].

III. Proposed Relaying Strategy

The DF relaying strategy provides reliable communication at the cost of non-negligible latency at the destination that grows linearly with the number of hops in the multi-hop system. The large latency at the destination results from the fact that the relay does not start re-encoding until the decoding is done. This problem would be greatly alleviated if decoding and re-encoding could operate simultaneously. Fig. 3 illustrates the motivation for proposing such a new relaying strategy. In the figure we assume zero propagation delay, i.e., once a codeword is generated by the encoder, it is received immediately by the decoder. In this way, the time it takes from source to destination is measured by the length of time it takes for the encodings and decodings in the system. Assume that the decoders at the relay and the destination are able to recover the source data after receiving t_2 and $t_6 - t_4$ time slots' coded data, respectively, as shown in (a) in Fig. 3. We see that the latency at the destination can be shortened by replacing fixed-rate codes with rateless codes, where decoding is complete right after sufficient data is collected, as shown in (b). The latency can be further improved

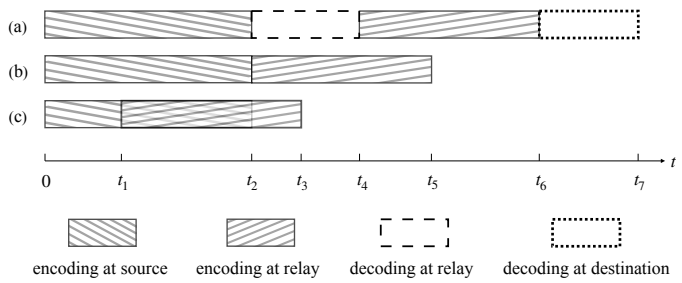

Fig. 3. Motivation of the proposed relaying strategy. The communications between source and destination via relay is measured by the length of time. The box filled with left- and right-tilted stripes represents the time it takes for encoding at the source and the relay, respectively; the blank box with dashed and dotted borders represents the time it takes for decoding at the relay and the destination, respectively.

if the relay starts re-encoding/transmitting before decoding is complete as shown in (c).

To have the relay decode and re-encode simultaneously, two requirements must be met: (i) the relay node operates in the full-duplex mode; and (ii) the relay recovers source data as coded data arrives. Nowadays as the usage of full-duplex relays is widespread, and as techniques such as [27–29] that mitigate the self-interference of full-duplex relays are well under development, the first requirement is no longer a concern. To meet the second requirement, rateless codes with good intermediate performance are needed. Now that we have discussed in Section I that MBLTE is a good candidate in this regard, the second requirement is also met.

A. Partial Decode-and-Forward Relaying

In this section, we consider full-duplex relays with rateless codes in the separated two-hop system, and assume no self-interference. Discussions on weighing between half- or full-duplex relays or the self-interference of full-duplex relays are beyond the scope of this paper, and can be found in the literature, cf. e.g. [30–34]. We also make the following assumptions: (i) each output symbol takes one time slot to be generated by the encoder; (ii) the output symbol is immediately received and processed by the decoder once it is generated; and (iii) the decoder can recover as many input symbols as possible in one time slot. We let $\mathcal{R}(t)$ and $\mathcal{D}(t)$ denote the set of recovered input symbols by the decoder at the relay and the destination, respectively, at time t, thus $\mathcal{R}(t)$ and $\mathcal{D}(t)$ are empty initially. We propose a partial DF (PDF) relaying strategy, where the relay starts re-encoding immediately once $\mathcal{R}(t)$ becomes non-empty. The relay encodes over the input symbols in $\mathcal{R}(t)$ dynamically until it is notified by the destination to terminate. We assume that the relay starts encoding in the same time slot as $\mathcal{R}(t)$ becomes non-empty.

To better illustrate our proposed PDF relaying strategy, we resort to an example shown in Fig. 4. The circles and rectangles represent input and output symbols, respectively. There are five input symbols $x_1, \ldots, x_5$. The encoder at the source encodes these symbols into a potentially unlimited number of output symbols y_i using modulo-2 operations. The decoder at the

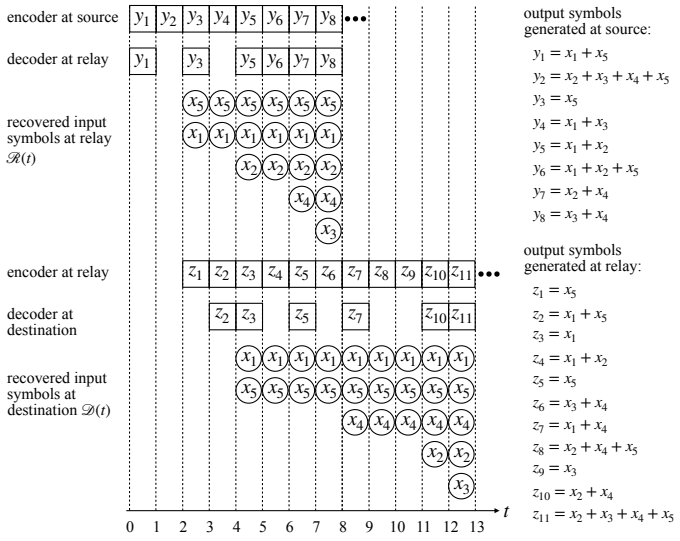

Fig. 4. Example of the proposed PDF relaying strategy.

relay process each successfully-received encoded symbol y_i. For instance, we assume that y_2 and y_4 are erased by the channel. The relay recovers x_5 and x_1 when $2 < t \leq 3$ and $\mathcal{R}$(t) becomes non-empty as shown in (2). The relay immediately starts re-encoding by encoding over x_5 and x_1 and generates its first output z_1, which is transmitted to the destination but is erased by the channel. The relay generates an output symbol by re-encoding over the input symbols in $\mathcal{R}$(t) during each time slot thereafter until $t = 13$ where the destination successfully completes decoding. The dynamic evolution of the sets $\mathcal{R}(t)$ and $\mathcal{D}(t)$ is shown in (2) and (3), respectively.

$$\mathcal{R}(t) = \begin{cases} \{\}, & 0 \leq t \leq 2 \\ \{x_5, x_1\}, & 2 < t \leq 4 \\ \{x_5, x_1, x_2\}, & 4 < t \leq 6 \\ \{x_5, x_1, x_2, x_4\}, & 6 < t \leq 7 \\ \{x_5, x_1, x_2, x_4, x_3\}, & t > 7 \end{cases} \tag{2}$$

$$\mathcal{D}(t) = \begin{cases} \{\}, & t \leq 4 \\ \{x_1, x_5\}, & 4 < t \leq 8 \\ \{x_1, x_5, x_4\}, & 8 < t \leq 11 \\ \{x_1, x_5, x_4, x_2\}, & 11 < t \leq 12 \\ \{x_1, x_5, x_4, x_2, x_3\}, & t > 12. \end{cases} \tag{3}$$

Next, we propose *amicable* encoding scheme that will be used in conjunction with MBLTE.

IV. Amicable Encoding Scheme

Based on the proposed relaying strategy, the relay starts re-encoding and transmitting immediately once an input symbol is recovered. The earlier the relay starts re-encoding, the earlier the destination starts reception and decoding; thus, the real-time delay experienced by the users is reduced. As such, in this section we aim to propose an amicable MBLTE so that (i) it starts recovering input symbols earlier; and (ii) it recovers more input symbols during each time slot.

For the MBLTE, the decoder is not able to recover any input symbols until a degree-1 output symbol is received. Therefore, a good encoder should generate and transmit degree-1 output symbols early. In addition, according to the encoding process of the MBLTE, input symbols that are connected to the degree-1 output symbols are highly likely to be also connected to the degree-2 output symbols. Once the above-mentioned input symbols are recovered, their neighboring degree-2 output symbols reduce to degree-1, enabling the decoder to proceed in terms of recovering more input symbols. Therefore, a good MBLTE should also transmit degree-2 output symbols early, following right after the transmission of the degree-1 symbols. However, because the encoder does not know how many output symbols to generate in the rateless scenario, it is also unknown how many degree-1 or degree-2 output symbols will be generated, not to mention ordering the transmissions.

To solve this problem, we propose an amicable rateless encoding scheme that divides the encoding process into two stages where the first stage can be viewed as a fixed-rate transmission and the second stage is rateless. We now predetermine the number of output symbols N for the first stage. According to [24], the optimal memory order is the smallest integer i that satisfies

$$N \sum\nolimits_{d=1}^{i} \Omega_d \geq K, \tag{4}$$

where Ω_d is the probability that an output symbol is degree-d. In our case where the memory order is $i = 2$, we rearrange the inequality as

$$N \geq K/(\Omega_1 + \Omega_2). \tag{5}$$

Although (5) does not apply for the rateless scenario where N is unknown, it is a useful tool in determining the number of output symbols for the first stage of encoding. According to [24], ideally without channel erasures the decoder is able to recover all input symbols with the reception of N output symbols if N satisfies (5), while the performance degrades as N grows if the channel erasure probability is nonzero. Therefore, selecting N to be the smallest integer that satisfies (5) results in the second-order MBLTE being optimal. We thereby modify the MBLTE as detailed in Algorithm 2.

Because the amicable MBLTE does not change the degree distribution of the output symbols, and because it does not affect the decoding process, it maintains the same low encoding and decoding complexities as the traditional MBLTE. Compared to the traditional one, the amicable encoder requires N additional buffers at the initial stage of encoding. However, N is not large and only depends on the input symbol length K when the degree distribution is fixed regardless of the channel condition. Moreover, the number of buffers decreases linearly as encoder proceeds as detailed in Algorithm 2. For applications where buffer size is less of a concern than delay or complexity, the amicable MBLTE is beneficial.

Algorithm 2 The encoding process of the amicable MBLTE.

Initialize two empty sets: S_1 and S_d
Define N as the smallest integer that satisfies (5)
Sample Ω a total of N times, store values in S_d. Rearrange the elements in S_d such that they are in ascending order
for $n = 1 : \infty$ **do**
  if $n \leq N$ **then**
    $d =$ the first element in S_d
    remove the first element from S_d
  else
    sample Ω to get d
  end if
  Select d input symbols as detailed in Alg. 1, Lines 4–18
end for

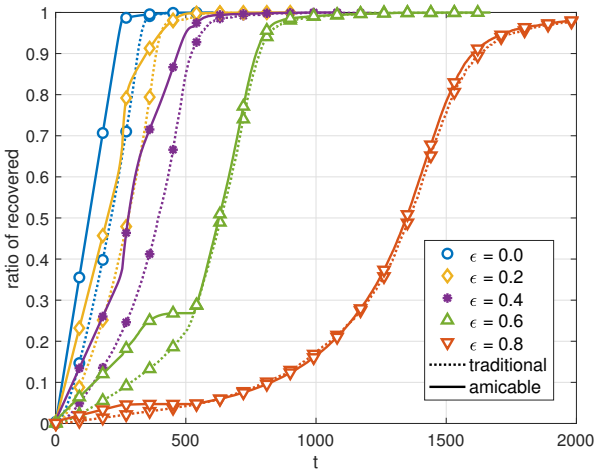

Fig. 5. Intermediate performance comparison of traditional MBLTE and amicable MBLTE as coding proceeds. The dotted lines represent the traditional encoder and the solid lines represent the amicable encoder. The channel erasure probability is denoted by ϵ. The input symbol length is fixed at $K = 256$.

V. Simulation Results

In this section, we present simulation results to evaluate the proposed partial decode-and-forward (PDF) strategy. Because the PDF would benefit from rateless codes that have good intermediate performance, we first compare the traditional MBLTE with the amicable MBLTE encoder and show that the latter has a better intermediate performance than the former. We then employ the amicable codes as the rateless codes in the two-hop system to evaluate the performance of our proposed PDF method against the decode-and-forward (DF) scheme. In the simulations, we fix the length of the input symbols at $K = 256$, and adopt the robust soliton distribution (RSD) with $c = 0.03$ and $\delta = 0.5$ [7].

A. Performance of Amicable MBLTE for Single-hop

Fig. 5 shows the intermediate performance of the traditional MBLTE and amicable MBLTE under various erasure probabilities ranging from $\epsilon = 0$ to $\epsilon = 0.8$. From the results, we note that although the performance of each encoder

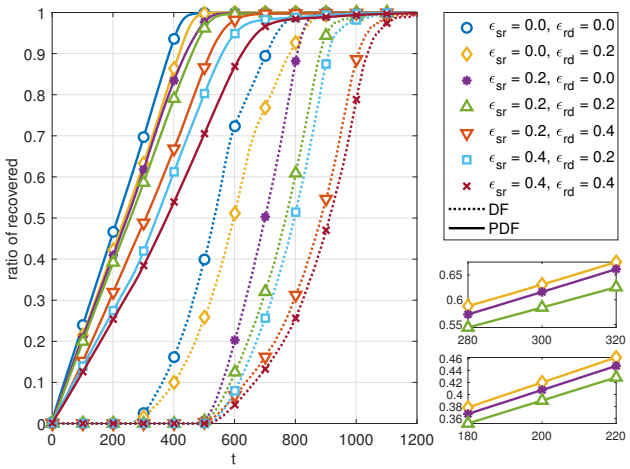

Fig. 6. Comparison of DF and PDF using the amicable MBLTE as coding proceeds. The dotted lines represent DF and the solid lines represent PDF. The vertical axis represents the ratio of recovered input symbols at the destination. The input symbol length is fixed at $K = 256$. Various respective channel erasure probabilities ϵ_{sr} and ϵ_{rd} for the source–relay and relay–destination links are evaluated. The two sub-figures give a detailed view of PDF for some channel conditions around $t = 200$ and $t = 300$.

degrades as channel conditions worsen, the amicable encoder outperforms the traditional encoder for all channel conditions shown in the figure. Specifically, the gap between the encoders is about the same for the cases where $\epsilon = 0$, $\epsilon = 0.2$, and $\epsilon = 0.4$; the gap becomes negligible during most of the decoding period when the channel erasure probability increases to $\epsilon = 0.6$ and $\epsilon = 0.8$. One of the interesting phenomena in the figure is that for $\epsilon = 0.6$ there is an obvious performance gain of the amicable encoder relative to the traditional one at the early stage of decoding where $t \leq 500$. It should be noted that we predetermine the number of output symbols N for the first stage of encoding in Algorithm 2 such that the amicable encoder differs from the traditional one only in the first N output symbols. For the parameters used in this simulation, the predetermined N is around 500. Therefore, this phenomenon appearing in the case of $\epsilon = 0.6$ is consistent with the amicable encoding process. Moreover, it verifies that the proposed amicable encoder is effective in improving the intermediate performance of the traditional MBLTE. A similar phenomenon is also observed for the case of $\epsilon = 0.8$ although it is less obvious because of the poor channel condition. This is not observed for cases where $\epsilon \leq 0.4$ because both codes have recovered most of the input symbols and are close to completion of decoding at $t = 500$.

B. Performance of Partial Decode-and-Forward for Two-hop

Fig. 6 compares the performance of PDF with DF by employing the amicable MBLTE as the rateless codes at the source and the traditional MBLTE at the relay. Various channel erasure probabilities are evaluated with ϵ_{sr} (source to relay) and ϵ_{rd} (relay to destination) ranging from 0 to 0.4, respectively. From the results, we see that PDF significantly improves the performance of DF for all channel conditions. Specifically,

for each case with the PDF strategy, the destination starts to recover input symbols right from the beginning; while with DF, the destination is not able to recover any input symbols until $t = 256$. In addition, in each erasure scenario shown in the figure, DF only recovers less than 20% of input symbols at the time when 97% are recovered by PDF.

Moreover, from the figure we also notice that the gap between the best and worst scenarios is much smaller for PDF than it is for DF, i.e. PDF recovers 50% of the input symbols when $t = 216$ in the case of $\epsilon_{sr} = 0$ and $\epsilon_{rd} = 0$, and when $t = 377$ in the case of $\epsilon_{sr} = 0.4$ and $\epsilon_{rd} = 0.4$, resulting in a gap of $\Delta t = 161$; while PDF does so at $t = 533$ and $t = 912$, resulting in a gap of $\Delta t = 379$, which is more than double that of PDF. Therefore, PDF is much more robust against performance degradation due to erasures.

VI. Conclusions

In this paper, we have proposed a partial decode-and-forward (PDF) relaying strategy and have realized it with rateless codes. We have argued that the proposed PDF benefits from rateless codes that have good intermediate performance. We have showed that the recently developed MBLTE has good intermediate performance and we have proposed an amicable encoding scheme for a further performance gain at the cost of additional temporary buffers. We have presented simulation results to show that our amicable encoding algorithm is efficient in further improving the intermediate performance especially when the channel erasure probability is not too high, and that our proposed PDF not only significantly improves the performance of DF but is also more robust against performance degradation due to erasures. Overall, the proposed encoding scheme and the relaying strategy are efficient towards near real-time data delivery in the UDN and also the Internet of Things (IoT) scenario where many devices/sensors are power-limited thus a direct link from the source to destination is impossible. In our future work, we will investigate the tradeoffs between the delay and the overhead of the PDF strategy and extend the work to scenarios with more than two hops.

References

[1] J. Hoadley and P. Maveddat, "Enabling small cell deployment with HetNet," *IEEE Wireless Communications*, vol. 19, pp. 4–5, April 2012.

[2] H. S. Dhillon, R. K. Ganti, F. Baccelli, and J. G. Andrews, "Modeling and analysis of K-tier downlink heterogeneous cellular networks," *IEEE Journal on Selected Areas in Commun.*, vol. 30, no. 3, pp. 550–560, 2012.

[3] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 56–67, 1998.

[4] J. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to asynchronous reliable multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 1528–1540, Oct 2002.

[5] P. Maymounkov, "Online codes," tech. rep., New York University, 2002.

[6] D. MacKay, "Fountain codes," *IEE Proceedings - Communications*, vol. 152, pp. 1062–1068, Dec 2005.

[7] M. Luby, "LT codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–280, 2002.

[8] A. Shokrollahi, "Raptor codes," in *IEEE ISIT'04*, June 2004.

[9] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, pp. 2551–2567, June 2006.

[10] J. Castura and Y. Mao, "Rateless coding for wireless relay channels," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1638–1642, 2007.

[11] R. Gummadi and R. S. Sreenivas, "Relaying a fountain code across multiple nodes," in *IEEE Info. Theory Workshop*, pp. 149–153, 2008.

[12] A. Talari and N. Rahnavard, "Enhanced intermediate packet delivery in delay tolerant networks," in *MILCOM 2010*, pp. 575–580, 2010.

[13] A. Talari and N. Rahnavard, "On the intermediate symbol recovery rate of rateless codes," *IEEE T. Commun.*, vol. 60, no. 5, pp. 1237–1242, 2012.

[14] N. Thomos, R. Pulikkoonattu, and P. Frossard, "Growth codes: Intermediate performance analysis and application to video," *IEEE Transactions on Communications*, vol. 61, no. 11, pp. 4710–4721, 2013.

[15] Y. Cassuto and A. Shokrollahi, "Online fountain codes with low overhead," *IEEE Trans. Info. Theory*, vol. 61, no. 6, pp. 3137–3149, 2015.

[16] M. Hashemi and A. Trachtenberg, "Near real-time rateless coding with a constrained feedback budget," in *52nd Annual Allerton Conf. on Commun., Control, and Computing (Allerton)*, pp. 529–536, 2014.

[17] M. Hashemi, Y. Cassuto, and A. Trachtenberg, "Fountain codes with nonuniform selection distributions through feedback," *IEEE Trans. Info. Theory*, vol. 62, no. 7, pp. 4054–4070, 2016.

[18] B. Jun, P. Yang, J. No, and H. Park, "New rateless codes for receiver with limited memory," in *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 482–484, 2015.

[19] B. Jun, P. Yang, J. No, and H. Park, "New fountain codes with improved intermediate recovery based on batched zigzag coding," *IEEE Transactions on Communications*, vol. 65, no. 1, pp. 23–36, 2017.

[20] K. F. Hayajneh, S. Yousefi, and M. Valipour, "Left degree distribution shaping for LT codes over the binary erasure channel," in *27th Biennial Symposium on Communications (QBSC)*, pp. 198–202, June 2014.

[21] K. Hayajneh, S. Yousefi, and M. Valipour, "Improved finite-length Luby-Transform codes in the binary erasure channel," *IET Communications*, vol. 9, no. 8, pp. 1122–1130, 2015.

[22] L. Shang and E. S. Perrins, "Second-order memory based LT encoder design," in *2016 IEEE Globecom Workshops*, pp. 1–6, Dec 2016.

[23] L. Shang and E. Perrins, "Memory based LT encoders over BIAWGN channels," in *IEEE GLOBECOM 2018*, pp. 1–6, Dec 2018.

[24] L. Shang and E. Perrins, "Optimal memory order of memory based LT encoders for finite block-length codes over binary erasure channels," *IEEE Transactions on Communications*, vol. 67, pp. 875–889, Feb 2019.

[25] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Info. Theory*, vol. 25, p. 572?584, Sep. 1979.

[26] C. Wang, D. J. Love, and D. Ogbe, "Transcoding: A new strategy for relay channels," in *2017 55th Allerton Conf.*, pp. 450–454, 2017.

[27] T. Riihonen, S. Werner, and R. Wichman, "Mitigation of loopback self-interference in full-duplex MIMO relays," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 5983–5993, 2011.

[28] I. Atzeni, M. Maso, and M. Kountouris, "Optimal low-complexity self-interference cancellation for full-duplex MIMO small cells," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2016.

[29] M. T. Kabir, M. R. A. Khandaker, and C. Masouros, "Reducing self-interference in full duplex transmission by interference exploitation," in *IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2017.

[30] V. Aggarwal, M. Duarte, A. Sabharwal, and N. K. Shankaranarayanan, "Full- or half-duplex? a capacity analysis with bounded radio resources," in *2012 IEEE Information Theory Workshop*, pp. 207–211, 2012.

[31] T. Riihonen, S. Werner, and R. Wichman, "Hybrid full-duplex/half-duplex relaying with transmit power adaptation," *IEEE Transactions on Wireless Communications*, vol. 10, no. 9, pp. 3074–3085, 2011.

[32] M. Duarte, C. Dick, and A. Sabharwal, "Experiment-driven characterization of full-duplex wireless systems," *IEEE Transactions on Wireless Communications*, vol. 11, no. 12, pp. 4296–4307, 2012.

[33] E. Aryafar, M. A. Khojastepour, K. Sundaresan, S. Rangarajan, and M. Chiang, "MIDU: enabling MIMO full duplex," in *18th Annual Intl. Conf. on Mobile Computing and Networking*, pp. 257–268, Aug. 2012.

[34] M. M. Razlighi and N. Zlatanov, "On buffer-aided relaying for the two-hop full duplex relay channel with self-interference," in *IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, 2017.