Low Overhead Growth Degree Coding Scheme for Online Fountain Codes with Limited Feedback

Pengcheng Shi[1], Zhenyong Wang[1,2](✉), and Dezhi Li[1]

[1] School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China
ZYWang@hit.edu.cn
[2] Shenzhen Academy of Aerospace Technology, Shenzhen 518057, China

Abstract. A new growth degree encoding scheme (GDS) for online fountain codes is proposed to achieve a low overhead when the feedback is limited. When the feedback points are determined at the completion phase, the encoder sends coded symbols with growth degrees between the two feedback points, rather than symbols with fixed degrees. This increases the effective probability of the coded symbols, thereby reducing the overall overhead. We analyze the overhead of the proposed scheme to demonstrate the performance. Simulation results show that our proposed scheme has better overhead performance compared to the conventional online fountain codes with limited feedback.

Keywords: Online fountain codes · Feedback · Overhead analysis · Rateless codes

1 Introduction

Fountain codes, also called rateless codes, are a class of erasure correction codes that can generate an infinite number of coded symbols from a limited number of original symbols through the eXclusive-OR (XOR) operation. In 2002, Luby proposed the first practical realization of digital fountain, named LT codes [1]. In 2006, Raptor codes are proposed by Shokrollahi through cascading fixed-rate codes and LT codes [2]. Later, spinal codes [3] and online fountain codes (OFC) [4] were proposed as new classes of fountain codes to transmit data efficiently and reliably.

Fountain codes were initially designed to transmit information without feedback. However, as it evolved, it made sense to use a small amount of feedback to enhance its performance. In 2015, online fountain codes (OFC) [4] were first proposed by Cassuto and Shokrollahi. The online property means that once given an instantaneous decoding state, the encoder can find the optimal coding strategy efficiently. Online fountain codes as a class of fountain codes, which feedback the current decoding state of the receiver to the transmitter over the feedback

Q. Guo et al. (Eds.): WiSATS 2021, LNICST 410, pp. 456–467, 2022.
https://doi.org/10.1007/978-3-030-93398-2_43

channel, so that the transmitter can adjust the degree value of coding symbols according to the feedback information, thereby efficiently transmitting encoding symbols and reducing decoding overhead. Later, in [5] and [6], the full recovery performance of online fountain codes is improved by non-random selection of original symbols. The intermediate performance is important in many applications such as audio and video streaming. Therefore, the intermediate performance of online fountain codes were studied in recent years [7–9]. In [10] and [11], the unequal error protection (UEP) property and unequal recovery time (URT) property are studied in online fountain codes.

With the research on online fountain codes, they have been applied to many applications such as wireless sensor networks [12] and satellite broadcast system [13]. However, the existing methods are designed for unlimited feedback. In practical applications, due to the shortage of feedback resources, the feedback times are usually limited. Therefore, it is a meaningful research direction to study how to efficiently transmit data with online fountain codes under limited feedback scenario. In [14], the authors first studied the feedback strategies for online fountain codes with limited feedback and proposed two schemes to select the optimal feedback points. However, since the degree value of the symbols sent between the two feedback points is fixed, when there are too many optimal degree values are skipped by the adjacent feedback points, the overhead will be high. This explains why the feedback strategy has a large overhead when the number of feedbacks is small. To address this problem, we propose the growth degree encoding scheme (GDS), which sends coded symbols with growth degrees between adjacent feedback points. This increases the effective probability of coded symbols, thus reducing the overall overhead.

The rest of this paper is organized as follows. Section 2 introduces the online fountain codes with limited feedback. The proposed growth degree encoding scheme are provided in Sect. 3. Section 4 provides the overhead analysis for the proposed scheme. The simulation results are given in Sect. 5. Section 6 concludes this paper.

2 Online Fountain Codes with Limited Feedback

In this section, we briefly review the online fountain codes (OFC) and the online fountain scheme with limited feedback. For the online fountain codes, the uni-partite graph are introduced to represent the decoding state. As shown in Fig. 1, an example of the bi-partite and the corresponding uni-partite graph is presented, the original symbols, or called source symbols, are represented by circle nodes, while the output symbols, or called encoded symbols, are represented by square nodes. In the bi-partite graph, if a square node are neighbored with two circle nodes by edges, it means the corresponding encoded symbol are generated by the XOR of these two source symbols. While in the uni-partite graph, the nodes are only used to represent the source symbols. The edge between two nodes indicates that an encoded symbol generated by the XOR of the two source symbols is received by the decoder. Blacked node represents the source symbol that have been recovered. A component means a set that any two circle nodes in this set are connected by an edge. Obviously, a component is decoded when one input symbol in this set is colored black. Note that the size of a component is the number of source symbols in this component.

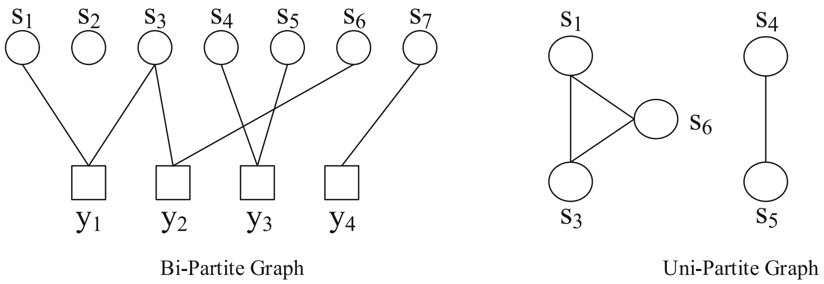

Fig. 1. A bi-partite graph for online fountain codes and the corresponding uni-partite graph.

The decoder receives coded symbols and updates the uni-partite graph. According to the uni-partite graph, the decoder can clear the current decoding state and get the optimal degree value at this time. When the optimal degree value changes, the decoder feeds back the decoding state to the transmitter. Then the transmitter modifies the degree value to transmit encoded symbols. At the decoder, an coded symbol is useful when it belongs to the following two cases.

- **Case 1:** A received symbol that degree m is the XOR of a single white symbol and $m-1$ black symbols.
- **Case 2:** A received symbol that degree m is the XOR of two white symbols and $m-2$ black symbols.

We assume there are k source symbols. The encoding process of online fountain codes is divided into two phases: build-up phase and completion phase.

Build-up Phase: The phase is consist of 2 steps. In the first step, the transmitter generate coded symbols with degree 2 continuously. The decoder receives the coded symbols and updates the uni-partite graph. Until the size of the largest component is $k\beta_0$, where β_0 is a predefined value that satisfies $0 < \beta_0 < 1$, the decoder sends feedback to inform the transmitter, then the first step ends at this time. In the second step, the transmitter sends the coded symbols with degree 1 until the largest component is colored black, which means all the source symbols in the largest component is recovered. Then the build-up phase ends.

Completion Phase: The transmitter generates degree-$\hat{m}$ coded symbols based on β until the decoding process is successful. β represents the recovery ratio of source symbols. The encoder selects an optimal value of $\hat{m}$ to maximize the probability of the coded symbol becoming a useful symbol. In other words, to maximize the sum of probabilities of Cases 1 and 2. Then the value of $\hat{m}$ satisfies:

$$\hat{m} = \arg\max_{m}[P_1(m, \beta) + P_2(m, \beta)] \tag{1}$$

The $P_1(m, \beta)$ and $P_2(m, \beta)$ can be calculated as follows:

$$P_1(m, \beta) = \binom{m}{1}\beta^{m-1}(1 - \beta) \tag{2}$$

$$P_2(m, \beta) = \binom{m}{2}\beta^{m-2}(1 - \beta)^2 \tag{3}$$

For the online fountain codes with limited feedback, the authors in [14] proposed two strategies to select optimal feedback points in all the feedback points. When the number of feedback times is determined, the heuristic table-lookup algorithm based on effective probability (HTLEP) and the heuristic table-lookup algorithm based on overhead (HLTO) are applied to select appropriate feedback points to reduce the overhead. However, since a fixed value of degree is selected between two feedback points, when the number of feedback times is small, the optimal degree value skipped between the two feedback points is too much. So the effective probability of the fixed value of degree will gradually decrease. This will lead to a high overhead.

3 Growth Degree Encoding Scheme for Online Fountain Codes

In this section, we will introduce the proposed growth degree encoding scheme for online fountain codes with limited feedback. For convenience, we refer to the limited feedback online fountain codes with fixed degree value between two feedback points as conventional scheme. At the completion phase, the degree value of the coded symbols sent by the conventional scheme is from the feedback, which is fixed until the next feedback. However, in our proposed scheme, the degree value between two feedback points is gradually growing. We use a set V to store the number of coded symbols that need to be sent for each skipped optimal degree, then the transmitter sends a fixed number of coded symbols according to the set V until the next feedback point is reached. If all the symbols in the set V are sent before the next feedback point is reached, the degree value is randomly selected according to the degree distribution for transmission until the next feedback is received. The specific design method is as follows.

We assume the ratio of decoded symbols to all source symbols as β. With the increase of β, from Eq. (1), we can calculate the optimal degree $\hat{m}$ corresponding to each β that maximizes the useful probability. We call the β that changes the value of $\hat{m}$ each time as the degree transition points. Note that in the conventional scheme, since there is no limit to the number of feedback times, the set of degree transition points are the set of feedback points.

In the proposed GDS scheme, the transmitter get the selected feedback points from the set of degree transition points according to the HTLEP or HTLO algorithm. We denote by B and M the set of selected feedback points and the corresponding feedback degrees, respectively. When the number of feedback times is determined to be N, we can get:

$$B = \{\beta_0, \beta_1, \ldots, \beta_N, 1\} \tag{4}$$

$$M = \{m_0, m_1, \ldots, m_N, m_{max}\} \tag{5}$$

where β_i and m_i represents the ith feedback point and feedback degree, $1 \le i \le N$, m_0 represents the corresponding optimal degree when β is β_0. m_{max} represents the maximum degree in the scheme.

It can be seen from the above description that the transmitter sends coded symbols between every two adjacent feedback points. Some transition points will be skipped between any two adjacent feedback points, and the optimal degree value will change at these transition points. Without loss of generality, we assume the transmitter receives the feedback and knows the decoded symbols ratio β is β_i. Between the feedback points (β_i, β_{i+1}), the skipped transition points are denoted by $\{\beta_{i,1}, \beta_{i,2}, \ldots, \beta_{i,j}\}$ and the corresponding transition degree are denoted by $\{m_{i,1}, m_{i,2}, \ldots, m_{i,j}\}$. By combining the set of feedback points and transition points, we can get the following two sets:

$$B^i = \{\beta_{i,0}, \beta_{i,1}, \ldots, \beta_{i,j}, \beta_{i+1,0}\} \tag{6}$$

$$M^i = \{m_{i,0}, m_{i,1}, \ldots, m_{i,j}, m_{i+1,0}\} \tag{7}$$

where $\beta_{i,0}$ and $m_{i,0}$ are the β_i and m_i in the set of B and M, respectively. With these two sets, we can derive the set N^i. It is the set of coded symbols that needs to be sent between the two feedback points. The dimension of N^i is $\left|N^i\right| = \left|B^i\right| - 1 = j + 1$. We denote by $N^i(x)$ the xth element in this set. Then $N^i(x)$ can be get by the following equation.

$$N^i(x) = \frac{k[B^i(x+1) - B^i(x)]}{P(M^i(x), \frac{B^i(x+1)}{k})} \tag{8}$$

where $P(m, \beta)$ represents the sum of probabilities of the received coded symbol with degree m belongs to Cases 1 and 2. So it can be calculated as follows.

$$\begin{aligned} P(m, \beta) &= P_1(m, \beta) + P_2(m, \beta) \\ &= \binom{m}{1} \beta^{m-1}(1-\beta) + \binom{m}{2} \beta^{m-2}(1-\beta)^2 \end{aligned} \tag{9}$$

The value of $N^i(x)$ represents the theoretical upper bound of the number of coded symbols with degree $M^i(x)$ when the value of β is between the transition points $B^i(x)$ and $B^i(x+1)$.

After obtaining N^i and M^i, the transmitter first sends $N^i(1)$ coded symbols of degree $M^i(1)$. Then it increases the value of degrees and sends $N^i(2)$ coded symbol of degree $M^i(2)$. Continue in this way until $N^i(j+1)$ coded symbols of degree $M^i(j+1)$ have been sent or the feedback β_{i+1} is received.

If all symbols in the set N^i are sent and still no feedback is received, the calculated degree distribution is used to select the degree values to generate a new coded symbol until the next feedback is reached. We denote by $\Omega^i(x)$ the degree distribution between the feedback points β_i and β_{i+1}. The probability of being selected for a degree value $M^i(t)$ is the ratio of the number of symbols of degree $M^i(t)$ in N^i to the number of all symbols in N^i, i.e., $\frac{N^i(t)}{sum(N^i)}$, where $sum(N^i)$ represents the sum of the numbers in the set N^i. So the degree distribution can be calculated as follows.

$$\Omega^i(x) = \sum_{t=1}^{j+1} \frac{N^i(t)}{sum(N^i)} x^{M^i(t)} \tag{10}$$

4 Overhead Analysis

In this section, we provide analysis for the performance of conventional online fountain codes with limited feedback and the proposed GDS scheme based on the theoretical analysis framework in [5].

4.1 Performance Analysis of Online Fountain Codes with Limited Feedback

In this subsection, we analyze the performance of online fountain codes with limited feedback. First we introduce a lemma as follows.

Lemma 1 [5]. *Denote by $P(n)$ the probability that a received coded symbol belongs to Case 1 or Case 2 when the degree of the symbol is optimal, where n represents the number of recovered symbols. Then we can calculate the value of $P(n)$ as below.*

$$P(n) = P_1(\hat{m}, \beta_0 + \frac{n}{k}) + P_2(\hat{m}, \beta_0 + \frac{n}{k}) \tag{11}$$

where $\hat{m}$, P_1 and P_2 satisfy the Eqs. (1), (2) and (3), respectively.

To give the relationship between the number of received coded symbols and the number of recovered symbols, we introduce a new lemma.

Lemma 2 [5]. *Denote $T(s)$ as the number of coded symbols to transmit with unlimited feedback when s source symbols have been recovered. Then the value of $T(s)$ can be evaluated as follows.*

$$T(s) = \frac{1}{2}kc + \frac{1}{\beta_0} + (1 - \frac{1}{2}(1-\beta_0)c) \sum_{i=1}^{s-k\beta_0} \frac{1}{P(i)} \tag{12}$$

where $k\beta_0 < s \leq k$. c is the average degree of the source symbols when the build-up phase is over, which satisfies the following:

$$c = \frac{\ln(1-\beta_0)}{-\beta_0} \tag{13}$$

From Lemma 2, we can get the overhead performance analysis of the online fountain codes with limited feedback as follows.

Corollary 1. *Denote $T^N(s)$ as the number of coded symbols to transmit with N times of feedback when s source symbols have been recovered. The set of feedback points and the corresponding feedback degrees are B and M. Then the value of $T^N(s)$ can be evaluated as follows.*

$$T^N(s) = \frac{1}{2}kc + \frac{1}{\beta_0} + (1 - \frac{1}{2}(1-\beta_0)c) \sum_{i=1}^{s-k\beta_0} \sum_{m \in M} \frac{f(i,m)}{P(m, \beta_0 + \frac{i}{k})} \tag{14}$$

where $P(m, \beta_0 + \frac{i}{k})$ satisfies Eq. (9) and $f(i,m)$ is a function that takes a value of 1 when the value of β is between the feedback points and the degree value m is the corresponding degree point at the same time. For convenience, we assume the value of i and m that meet the conditions as the event E. Then we can get the value of $f(i,m)$ as follows.

$$f(i,m) = \begin{cases} 1 & (i,m) \in E \\ 0 & else \end{cases} \tag{15}$$

Proof. Since online fountain codes with limited feedback is designed for the completion phase, the relationship between the number of coded symbols sent by the transmitter at the build-up phase and the recovery rate is the same as the conventional online fountain codes. From Lemma 2, we know at the end of build-up phase, the transmitter needs to send $\frac{1}{2}kc + \frac{1}{\beta_0}$ coded symbols. And it can be seen from [5] that at the completion phase, n useful symbols, including the build-up edges, Case 1 symbols and Case 2 symbols, can recover n source symbols. Therefore, at the completion phase, we denote by $T_{ca1,2}(n)$ the number of Case 1 and Case 2 symbols required to recover n source symbols. Then it can be calculated as follows:

$$T_{ca1,2}(n) = (1 - \frac{1}{2}(1-\beta_0)c)n \tag{16}$$

In the online fountain codes with limited feedback, different from the conventional online fountain codes, the decoder only feeds back the corresponding degree value at a few fixed feedback points. The degree value between the two feedback points is constant. Therefore, the number of Cases 1 and 2 symbols n needs to be calculated separately between every two adjacent feedback points. Without loss of generality, we assume that the value of s/k is between the feedback points β_i and β_{i+1}. Then we first calculate the number of Cases 1 and 2 symbols required between the feedback points β_0 and β_1. We represent the value as n_0. Same as the analysis in [5], we can get:

$$n_0 = \sum_{t=1}^{k(\beta_1-\beta_0)-1} \frac{1}{P(m_0, \beta_0 + \frac{t}{k})} \tag{17}$$

We denote by n_i the number of Cases 1 and 2 symbols required between the feedback points β_i and β_{i+1}. Based on the Eq. (17), we can get $n_1, n_2, \ldots, n_{i-1}$ according to the above analysis. In addition, the value of n_i can be calculated by

$$n_i = \sum_{t=k(\beta_i-\beta_0)}^{s-k\beta_0} \frac{1}{P(m_i, \beta_0 + \frac{t}{k})} \tag{18}$$

Therefore, the number of Cases 1 and 2 symbols needed to recover s symbols is the sum of $n_0, n_1, \ldots, n_i$. From Eq. (16), at the completion phase, the number of coded symbols required to recover s symbols is

$$T^N(s) = \frac{1}{2}kc + \frac{1}{\beta_0} + (1 - \frac{1}{2}(1-\beta_0)c)\sum_{t=0}^{i} n_t \tag{19}$$

Through derivation, it is obvious that Eq. (14) and Eq. (19) are equivalent. □

4.2 Performance Analysis of the Proposed GDS Scheme

In this subsection, we analyze the theoretical performance of the proposed GDS scheme over lossy channels. We denote by ϵ the channel erasure probability. Note that when the channel is lossless channels, the value of ϵ is 0.

We present the performance analysis of our proposed GDS scheme through the following corollary.

Corollary 2. *Denote $T_p^N(s)$ as the number of coded symbols to transmit in the GDS scheme with N times of feedback when s source symbols have been recovered. The set of feedback points and the corresponding feedback degrees are B and M. The set B^i, M^i and N^i, Ω^i have been calculated when $0 \le i \le N$. Then the value of $T_p^N(s)$ can be evaluated as follows.*

$$\begin{aligned} T_p^N(s) = &\frac{kc}{2(1-\epsilon)} + \frac{1}{\beta_0(1-\epsilon)} + (1 - \frac{1}{2}(1-\beta_0)c) \\ &\cdot [\sum_{\beta_i=B(1)}^{B(b-1)} G_{\beta_i}(k\beta_{i+1} - k\beta_i) + G_{B(b)}(s - k\beta_{B(b)})] \end{aligned} \tag{20}$$

where $B(b)$ represents that the value of s/k is between the value of $B(b)$ and $B(b+1)$ in B, i.e., $s/k \in [B(b), B(b+1)]$. We assume $G_{\beta_i}(x)$ is a function that satisfies the equation: $G_{\beta_i}(x) = \sum_{t=1}^{x} g(\beta_i, t)$. And $g(\beta_i, t)$ is a function that satisfies the following cases:

- *The value is $\frac{1}{P(M^i(1), \beta_i + \frac{t}{k})}$, when $0 \le G_{\beta_i}(t-1) \le N^i(1)(1-\epsilon)$.*
- *The value is $\frac{1}{P(M^i(r), \beta_i + \frac{t}{k})}$, when $\sum_{h=1}^{r} N^i(h)(1-\epsilon) < G_{\beta_i}(t-1) \le \sum_{h=1}^{r+1} N^i(h)(1-\epsilon)$ $\forall r > 1$.*

– *The value is* $\sum_{h=1}^{j+1} \frac{\Omega^i(h)}{P(M^i(h), \beta_i + \frac{t}{k})}$, *when* $G_{\beta_i}(t-1) > sum(N^i)(1-\epsilon)$.

Proof. Because the proposed GDS scheme is designed for the completion phase, the performance analysis of the build-up phase is similar to the conventional online fountain codes. And based on the Eq. (16) in the proof of Corollary 1, we know n represents the number of Cases 1 and 2 symbols, which is the focus of the analysis.

In the proposed GDS scheme, we need to analyze the number of Cases 1 and 2 symbols for every two adjacent feedback points. Without loss of generality, we analyze the number for β_i and β_{i+1}. The value of N^i can be calculated by (8). The transmitter sends $N^i(1)(1-\epsilon)$ coded symbols with degree $M^i(1)$, the probability of being Cases 1 and 2 symbols is $\frac{1}{P(M^i(1), \beta_i + \frac{t}{k})}$. So we can prove the first condition of $g(\beta_i, t)$. In the same way, we can prove the second condition.

For the third condition, when the symbols in set N^i have been sent, the transmitter sends symbols according to the degree distribution Ω^i. For every degree value, we get the product of the number of symbols with the degree value becomes Cases 1 and 2 symbols and the probability that the degree value is selected. Then we sum all the products to get the expected value. So we prove the third condition. □

5 Simulation Results

In this section, we first verify the proposed analysis by comparing with the simulation results. We assume the $k = 1000$ and $\beta_0 = 0.5$. We compare the performance for the three feedback points selection strategies, HTLEP, HTLO and EVEN. The EVEN strategy is a simple feedback point selection strategy that equals the number of degrees skipped between every two feedback points.

Figure 2 shows the overhead performance of the OFC with limited feedback and the proposed GDS scheme. For the feedback points selection strategies, HTLEP, HTLO and EVEN, we perform analysis and simulations over the lossless channels when the number of feedback times is 1 to 5. As shown in Fig. 2, the theoretical analysis matches with the simulation results, which demonstrate the accuracy of our proposed analysis. When the channel is lossless channels, we also observe that compared with the conventional encoding scheme, the GDS scheme can effectively reduce the overhead, especially when the number of feedback times is extremely small.

In Tables 1 and 2, we compare the overhead performance of conventional scheme and GDS scheme over the lossy channel. We set the channel erasure $\epsilon = 0.1$ and $\epsilon = 0.2$. As can been seen from the tables, the proposed GDS scheme still has good overhead performance over lossy channel. Even with the simple EVEN feedback selection strategy, online fountain codes with limited feedback can achieve very low overhead by using the GDS scheme.

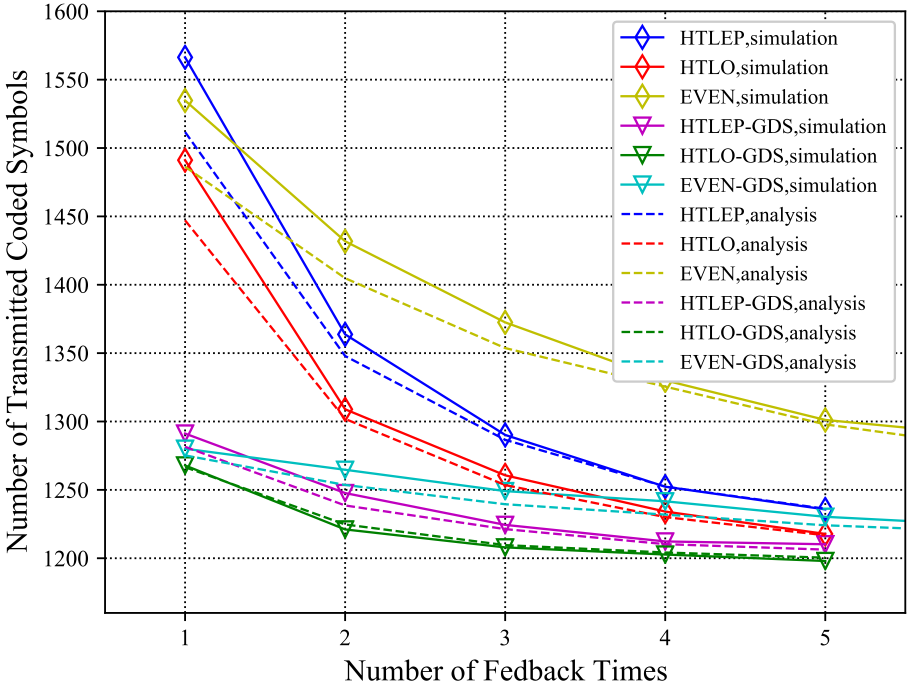

Fig. 2. Analysis and simulation results of the conventional OFC with limited feedback and the proposed GDS scheme. We set $k = 1000$, $\beta_0 = 0.5$ and $\epsilon = 0$. We simulate the three feedback strategies HTLEP, HTLO and EVEN when the number of feedback times is 1 to 5.

Table 1. Overhead performance of the conventional OFC with limited feedback and the GDS scheme when $\epsilon = 0.1$.

	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$
HTLEP	1754.34	1507.09	1433.26	1391.31	1379.14
HTLO	1651.6	1458.16	1398.66	1370.12	1353.07
EVEN	1720.37	1587.89	1516.88	1475.83	1451.15
HTLEP-GDS	1409.42	1369	1349.03	1343.61	1338.21
HTLO-GDS	1391.22	1352.81	1337.48	1336.77	1334.86
EVEN-GDS	1400.67	1384.88	1376.45	1365.33	1355.98

Table 2. Overhead performance of the conventional OFC with limited feedback and the GDS scheme when $\epsilon = 0.2$.

	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$
HTLEP	1955.83	1701.6	1613.87	1570.31	1551.93
HTLO	1859.36	1634.92	1576.42	1548.86	1524.73
EVEN	1930.94	1793.89	1722.36	1664.51	1635.85
HTLEP-GDS	1555.27	1525.12	1517.77	1506.44	1502.45
HTLO-GDS	1545.87	1516.67	1503.21	1500.56	1499.83
EVEN-GDS	1557.6	1529.9	1527.61	1521.84	1517.53

6 Conclusions

In this paper, we proposed growth degree encoding scheme for online fountain codes with limited feedback to achieve a low overhead. Between the two feedback points, the transmitter initially sends a fixed number of coded symbols with growth degree values. When all the coded symbols are transmitted, if no feedback is received at this time, the coded symbols are generated according to the calculated degree distribution until the feedback information is received. We also analyzed the overhead performance of the conventional OFC with limited feedback and the proposed scheme. Both the theoretical analysis and the simulation results showed that the proposed scheme can achieve a significant overhead reduction when the number of feedback times is small.

Acknowledgements. This work was supported in part by Beijing Natural Science Foundation under Grant L182032, by the National Natural Science Foundation of China (No. 61601147), and by the Development Foundation of CECT54 (Grant No. XX17641C008).

References

1. Luby, M.: LT codes. In: the 43rd Annual IEEE Symposium on Foundations of Computer Science, pp. 271–280 (2002)
2. Shokrollahi, A.: Raptor codes. IEEE Trans. Inf. Theory **52**(6), 2551–2567 (2006)
3. Perry, J., Iannucci, P.A., Fleming, K.E., Balakrishnan, H., Shah, D.: Spinal codes. SIGCOMM Comput. Commun. Rev. **42**(4), 49–60 (2012)
4. Cassuto, Y., Shokrollahi, A.: Online fountain codes with low overhead. IEEE Trans. Inf. Theory **61**(6), 3137–3149 (2015)
5. Huang, J., Fei, Z., Cao, C., Xiao, M., Jia, D.: Performance analysis and improvement of online fountain codes. IEEE Trans. Commun. **66**(12), 5916–5926 (2018)
6. Huang, T., Yi, B.: Improved online fountain codes based on shaping for left degree distribution. AEU-Int. J. Electron. C. **79**, 9–15 (2017)
7. Zhao, Y., Zhang, Y., Lau, F.C.M., Yu, H., Zhu, Z.: Improved online fountain codes. IET Commun. **12**(18), 2297–2304 (2018)

8. Shi, P., Wang, Z., Li, D., Xiang, W.: Zigzag decodable online fountain codes with high intermediate symbol recovery rates. IEEE Trans. Commun. **68**(11), 6629–6641 (2020)
9. Huang, J., Fei, Z., Cao, C., Xiao, M.: Design and analysis of online fountain codes for intermediate performance. IEEE Trans. Commun. **68**(9), 5313–5325 (2020)
10. Huang, J., Fei, Z., Cao, C., Xiao, M., Jia, D.: On-line fountain codes with unequal error protection. IEEE Commun. Lett. **21**(6), 1225–1228 (2017)
11. Cai, P., Zhang, Y., Pan, C., Song, J.: Online fountain codes with unequal recovery time. IEEE Commun. Lett. **23**(7), 1136–1140 (2019)
12. Yi, B., Xiang, M., Huang, T., Huang, H., Qiu, K., Li, W.: Data gathering with distributed rateless coding based on enhanced online fountain codes over wireless sensor networks. AEU-Int. J. Electron. Commun. **92**, 86–92 (2018)
13. Huang, J., Fei, Z., Cao, C., Xiao, M., Xie, X.: Reliable broadcast based on online fountain codes. IEEE Commun. Lett. **25**(2), 369–373 (2021)
14. Cai, P., Zhang, Y., Wu, Y., Chang, X., Pan, C.: Feedback strategies for online fountain codes with limited feedback. IEEE Commun. Lett. **24**(9), 1870–1874 (2020)