A Filtering Dimension Reduction Decoding Algorithm for Underwater Acoustic Networks

Lijuan Wang[1,2], Xiujuan Du[1,2](✉), Chong Li[1,2], Duoliang Han[1,2], and Jianlian Zhu[1,2]

[1] School of Computer, Qinghai Normal University, Xining 810008, China
1041517271@qq.com, dxj@qhnu.edu.cn
[2] Key Laboratory of IoT of Qinghai Province, Xining, China

Abstract. Underwater acoustic channels are fragile. Reliable data transmission in underwater acoustic networks (UANs) faces tough challenges. Digital Fountain Code (DFC) is an efficient rateless error-correcting coding technique, in which the redundancy is not fixed and can be decided on the fly as the channel evolves. Thus, DFC is considered near-optimal for underwater acoustic channels. A recursive LT (RLT) code is a DFC tailored for underwater acoustic networks, which allows for lightweight implementation of an encoder and a decoder. Based on the analysis of the RLT algorithm, this paper proposes a filtering dimension reduction (FDR) decoding algorithm for underwater acoustic networks. The FDR decoding algorithm executes XOR operations on the encoded packets in a strict short ring of a generating a matrix to reduce the dimensions of the encoded packets, or generate 1-degree encoded packets. As a result, the FDR algorithm can increase the number of 1-degree encoded packets and reduce the decoding complexity. Moreover, the FDR algorithm can eliminate the waiting time for a traditional decoding algorithm to receive the 1-degree packets, and achieve fast decoding. Simulation results based on NS3 show that the decoding success rate of the FDR algorithm is higher than that of the RLT algorithm.

Keywords: Underwater acoustic network · Reliable transmission · Digital fountain code · Recursive LT · Filtering dimension reduction

1 Introduction

With the development of the wireless sensor network technology and people's increasing attention to underwater resources, marine environment, marine rights and interests, underwater acoustic networks (UANs) have attracted more and more research attention in recent years [1–5]. UANs use acoustic communication, and an acoustic channel is characterized by low bandwidth, long propagation delay, high error probability, Doppler effect and spatiotemporal variation, which make traditional transmission mechanisms inapplicable or inefficient in UANs. Therefore, UANs face great challenges for reliable transmission and call for new reliable transmission mechanisms. Digital fountain code (DFC) is of great significance to achieve reliable transmission in UANs [6–9].

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2019
Published by Springer Nature Switzerland AG 2019. All Rights Reserved
J. Zheng et al. (Eds.): ADHOCNETS 2019, LNICST 306, pp. 276–287, 2019.
https://doi.org/10.1007/978-3-030-37262-0_23

DFC is an efficient rateless error-correcting coding technique. The redundancy of DFC is not fixed and can be decided on the fly as the channel evolves. Thus, DFC is considered near-optimal for underwater acoustic channels. A recursive LT (RLT) code is a DFC tailored for underwater acoustic networks, which allows for lightweight implementation of an encoder and a decoder [10]. Based on the analysis of the encoding characteristics and shortcomings of the RLT algorithm, this paper proposes a filtering dimension reduction (FDR) decoding algorithm, which can reduce the decoding complexity, and eliminate the waiting time for a traditional decoding algorithm to receive the 1-degree packets. Simulation results based on NS3 show that the decoding success rate of the FDR algorithm is higher than that of the RLT algorithm.

The remainder of the paper is organized as follows. Related work is introduced in Sect. 2. An FDR filtering dimension reduction decoding algorithm is presented in Sect. 3. Simulation results are shown and analyzed in Sect. 4. Finally, the paper is concluded in Sect. 5.

2 Related Work

Existing reliable transmission mechanisms for UANs can be divided into three categories: retransmission-based, forward error correction code (FEC)-based, and hybrid approach-based. Digital fountain coding is a reliable coding technique based on forward error correction coding. Early reliable coding techniques based on FEC usually adopted network coding with multipath routing, as proposed by Guo et al. in [11]. However, multipath routing usually brings about collision and retransmission. In [12], Liu and Garcin proposed a packet-level FEC reliable transmission mechanism. In a packet-level FEC-based transmission mechanism, the redundancy transmitted is fixed prior to transmission, which is not applicable in UANs. In [13] Peng et al. proposed a reliable transmission mechanism SDRT for piecewise data. SDRT protocol adopts SVT code to improve encoding/decoding efficiency. Nevertheless, after pumping the packets within the window quickly into the channel, the sender sends the packets outside the window at a very slow rate until receiving a positive feedback from the receiver, which reduces channel utilization. The original ADELIN was proposed in [14], which determines the appropriate FEC assemblage according to the distance between nodes, and realizes reliable transmission for underwater data. In [15], Mo et al. put forward a coding-based multi-hop coordinated reliable transmission mechanism. However, the encoding vectors are generated randomly, and thus the success probability of recovering data packets from encoded packets cannot be guaranteed, and its decoding complexity is higher than other sparse codes. Furthermore, the multihop coordination mechanism requires time synchronization and is restricted in a string topology, where there is a single sender and a single receiver. In [10], Du et al. proposed a RLT (recursive LT) code, which is applicable to dynamic UANs with limited transmission time between two nodes. RLT reduces the overhead of encoding and decoding. An RLT code is a DFC tailored for underwater acoustic networks, which allows for lightweight implementation of an encoder and a decoder. In this paper, an FDR decoding algorithm is proposed to overcome the shortcomings of the RLT algorithm. Next we introduce the RLT algorithm in detail.

For a given parameter $(k, d, \Omega(d))$ of an RLT algorithm, k is the number of original packets, $d(d \in \{1,\ 2,\ 3,\ 4,\ k\})$ denotes the degree value of an encoded packet, $\Omega(d)$ is the degree distribution. The input packet is represented as $\{S_1, S_2, \ldots, S_k\}$, k input packets composes a set D. The sequence of encoded packets is represented as $\left\{Y_1, Y_2, \ldots, Y_j, \ldots, Y_n\right\}(n > k)$. The RLT coding process is described as follows.

(1) From the set D, successively XOR the k input packets to generate one encoded packet with degree k, and then duplicate the encoded packet to obtain $\lceil 1/(1-p_p) \rceil$ copies. Here, p_p denotes the error probability of a packet.
(2) From the set D, select $\lceil m/(1-p_p) \rceil$ distinct packets randomly to constitute a seed set U_1 and generate $\lceil m/(1-p_p) \rceil$ encoded packets with degree 1. Here, m is the expected number of encoded packets received successfully with degree 1. In reality, we can set $1 \leq m \leq \max(\lfloor k/4 \rfloor, 1)$.
(3) Let $U_2 = I - U_1$. From the set U_2, select uniformly $\lceil k/\left(2(1-p_p)\right) \rceil$ input packets at random, and do XOR operation respectively with one packet selected randomly from the set U_1. Thus, generate $\lceil k/\left(2(1-p_p)\right) \rceil$ encoded packets with degree 2.
(4) Let $U_3 = I - U_1 - U_2$. If $|U_3| > \lceil k/\left(6(1-p_p)\right) \rceil$, select $\lceil k/\left(6(1-p_p)\right) \rceil$ input packets at random from the set U_3; otherwise, from the set D, do XOR operation, respectively, with one packet from the set U_2 and another from the set U_1 to generate $\lceil k/\left(6(1-p_p)\right) \rceil$ encoded packets with degree 3.
(5) Let $U_4 = I - U_1 - U_2 - U_3$. If $|U_4| > \lceil (\xi + k/3 - m - 1)/\left(1-p_p\right) \rceil$, select randomly $\lceil (\xi + k/3 - m - 1)\ /\left(1-p_p\right) \rceil$ input packets from the set U_4; otherwise, from the set D, do XOR operation, respectively, with three packets from U_1, U_2, U_3 respectively, to generate $\lceil (\xi + k/3 - m - 1)\ /\left(1-p_p\right) \rceil$ encoded packets with degree 4.

3 FDR Decoding Algorithm

3.1 Analysis of RLT Coding

Consider a set of input (original) packets with each having the same number of bits. The RLT encoder takes input packets and generates a potentially infinite sequence of encoded packets. Each encoded packet is computed independent of others. More precisely, given k input packets $\{S_1, S_2, \ldots, S_k\}$, a sequence of encoded packets $\left\{Y_1, Y_2, \ldots, Y_j, \ldots, Y_n\right\}$ are generated, where $n > k$.

The generating matrix G_{kn} is a $k \times n - order$ binary matrix. Let $G_{kn} = [g_1, g_2, g_3, \ldots]$ be the generating matrix with n column vectors, $g_m = [g_{1m}, g_{2m}, \ldots, g_{km}]^T$, $m = 1,\ 2,\ 3,\ \ldots n$. The value of each vector member is 0 and 1. Thus, G_{kn} is expressed as

$$
\begin{array}{c}
S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ \cdots \\ S_{k-1} \\ S_k
\end{array}
\begin{bmatrix}
1 & 0 & 1 & 1 & 0 & 0 & 1 & \ldots & 0 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & \ldots & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & \ldots & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & \ldots & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & \ldots & 1 & 1 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & \ldots & 1 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 & 0 & \ldots & 0 & 0
\end{bmatrix}
$$
$$
\begin{array}{cccccccccc}
Y_1 & Y_2 & Y_3 & Y_4 & Y_5 & Y_6 & Y_7 & \ldots & Y_{n-1} & Y_n
\end{array}
$$

There may exist some "short rings" in the generating matrix of the RLT encoded packets. The definition and properties of a "short ring" are given below.

Definition 1. In the generating matrix, if there are two or more columns, of which the values of corresponding two or more rows are all "1", then the elements with value "1" in these rows and corresponding columns constitute a closed ring, which is called "short ring".

If the number of rows satisfying the definition of a "short ring" is two, then the short ring formed by these two rows is called $4-memberd$ ring. If the number of such rows is three, then the short ring is a $6-membered$ ring, and so on. Assuming that the number of rows satisfying the definition of a "short ring" is $k^{'}\left(2 \leq k^{'} < k\right)$, the short rings formed by them are $2k^{'}-membered$ ring. Here, the definition of "short ring" is introduced to explain the phenomenon of decoding termination existed in the RLT algorithm, which is shown in Fig. 1.

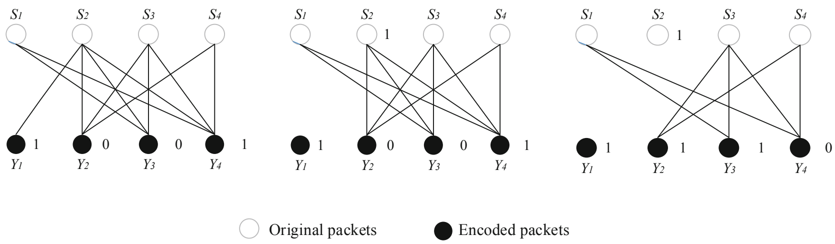

Fig. 1. Decoding termination of RLT code

In Fig. 1, it is seen that $Y_1 = S_2$, $Y_2 = S_2 \oplus S_3 \oplus S_4$, $Y_3 = S_1 \oplus S_2 \oplus S_3$, and $Y_4 = S_1 \oplus S_2 \oplus S_3 \oplus S_4$. According to the RLT decoding rules, we firstly find out the 1-degree encoded packet Y_1, then decode it to obtain S_2. Thus the line connecting Y_1 and S_2 can be removed. Next XOR operations for $\{Y_2,\ Y_3,\ Y_4\}$ and S_2 are performed. After that, the lines connecting $\{Y_2,\ Y_3,\ Y_4\}$ and S_2 are removed respectively. At this moment, the degree values of the remaining encoded packets $Y_2,\ Y_3,\ Y_4$ are 2, 2, 3. Without any 1-degree encoded packet, the decoding process of RLT is forced to terminate. The corresponding generating matrix of $Y_2,\ Y_3,\ Y_4$ is shown in Fig. 2.

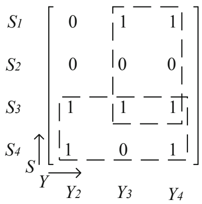

Fig. 2. An example of short rings

In Fig. 2, the generating matrix contains two $4-memberd$ short rings. The values of both Y_3 and Y_4 columns are "1" in the first row as well as the third row. Similarly, the values of both Y_2 and Y_4 columns are "1" in the third row as well as the fourth row. Given (n, k, Ω_x) of RLT, k is the number of original packets, n denotes the number of encoded packets. The proportion of column vectors with weight i of the generating matrix $k \times n - order\ G$ to the total column vectors is Ω_i. Thus, the probability of a column vector with weight i in G can be calculated as

$$P_i = \Omega_i / \binom{k}{i} \tag{1}$$

The probability that a column vector with weight j and a column vector with weight i constitute a $4-memberd$ short ring is given by

$$P_{r \cdot 4}(i,j) = \Omega_i \Omega_j \binom{k-2}{j-2} / \left(\binom{k}{k-j} \binom{k}{j} \right) \tag{2}$$

Next we give the definition of a "strict short ring".

Definition 2. In the generating matrix, if there are two columns, of which the values are all "1" in two or more corresponding rows, and the values are all "0" in other rows, then these rows and the corresponding columns constitute a closed ring, which is called "strict short ring". Assuming that there are $k'(2 \le k' < k)$ rows satisfying the definition of a "strict short ring", the short rings formed by them are strict $(2k') - membered$ rings.

The probability that a column vector with weight $j(j > 2)$ and a column vector with weight 2 constitute a strict $4-membered$ short ring can be calculated as

$$P_{strict(r \cdot 4)}(2,j) = \Omega_2 \Omega_j \binom{k-2}{j-2} / \left(\binom{k}{k-j} \binom{k}{2} \right) \tag{3}$$

Accordingly, the probability that a column vector with weight j and a column vector with weight $m(m < j)$ constitute a strict $2m - membered$ short ring can be calculated as

$$P_{strict(r\cdot 2m)}(m,j) = \Omega_m\Omega_j\binom{k-m}{j-m}/\left(\binom{k}{k-j}\binom{k}{m}\right) \tag{4}$$

Then the probability of a strict $4-memberd$ short ring in the order-generating matrix is given by

$$P_{strict(r\cdot 4)} = \binom{n}{2}\sum\nolimits_{j=2}^{\Omega_k}\Omega_2\Omega_j\binom{k-2}{j-2}/\binom{k}{k-j}\binom{k}{2} \tag{5}$$

Accordingly, the probability of a strict $2m-memberd$ short ring in the generating matrix can be approximately expressed as

$$P_{strict(r\cdot 2m)} = \binom{n}{2}\sum\nolimits_{m=2}^{\Omega_{k-1}}\sum\nolimits_{j=3}^{\Omega_k}\Omega_m\Omega_j\binom{k-m}{j-m}/\left(\binom{k}{k-j}\binom{k}{m}\right) \tag{6}$$

3.2 FDR Decoding Algorithm

In Fig. 1, the generating matrix contains two $4-memberd$ strict short rings. Accordingly, we analyze the original packet set. The set of the original packets corresponding to Y_2 is $\{S_3, S_4\}$. The set of the original packets corresponding to Y_3 is $\{S_1, S_3\}$. The set of the original packets corresponding to Y_4 is $\{S_1, S_3, S_4\}$. The above three sets have the following inclusion relationships: $\{S_3, S_4\} \subsetneq \{S_1, S_3, S_4\}$ and $\{S_1, S_3\} \subsetneq \{S_1,\ S_3,\ S_4\}$. Here, $S_1 = Y_2 \oplus Y_4$ can be obtained by XOR operation of Y_2 and Y_4. S_4 can be obtained by XOR operation of Y_3 and Y_4. S_3 can be obtained by XOR operation of Y_3 and S_1. According to the RLT decoding rules, decoding operations of S_2 is $C_{S2} = 1$, while the decoding operations of S_1, S_3, S_4 is infinite, which can be defined as $C_{S1}, C_{S3}, C_{S4} \to \infty$. Therefore, the decoding operations of all original packets are $C_{all} = C_{S1} + C_{S2} + C_{S3} + C_{S4} \to \infty$. But through XOR operations between the decoded packets S_1, S_3, S_4 can be decoded. The decoding operations of S_1, S_2, S_3, S_4 are $C'_{S1} = 1$, $C'_{S2} = 1$, $C'_{S3} = 2$, $C'_{S4} = 1$. The decoding operations of all original packets are $C'_{all} = C'_{S1} + C'_{S2} + C'_{S3} + C'_{S4} = 5 \ll C_{all}$.

A strict short-ring seems to be worthless for the traditional decoding technique. However, if the decoding algorithm is changed, the strict short-ring can play an active role in the decoding process and the contribution of a strict short-ring would not be ignored. What's more, a strict short-ring may even become the key to decoding the remaining packets in the "stop set".

Conclusion 1. If the two columns constituting a short ring have different degrees, and the values of the column with a smaller degree in other rows are all "0", then the two encoded packets corresponding to the short-ring can be XOR operated, and the generated encoded packet, which is called quadratic encoded packet, has a degree equal to the degree-difference of the two packets involving XOR operation. Thus, the degrees of packets are filtered and reduced by FDR decoding. If the degree-difference is 1, the degree of quadratic encoded packets is 1 and an original packet is recovered.

In RLT, a receiving node starts a decoding process after receiving a certain number of encoded packets. FDR eliminates the waiting time of the RLT algorithm, and thus achieves fast decoding. After encoding, the 1-degree encoded packets are sent first, and the FDR decoder starts immediately the decoding process upon receiving a packet no matter whether the degree value of the encoded packet is 1 or not. As in Fig. 1, after receiving two encoded packets, Y_4 and Y_3, the receiver compares the original packet ID sets. If one set is another set's subset, the receiver also starts the decoding process and decodes the two encoded packets. Therefore, we obtain the following conclusion.

Conclusion 2. When an FDR receiver receives encoded packets, it can start the decoding process as long as there exists a true inclusion relationship between the two sets of original packets. It does not have to wait for the encoded packet with 1-degree, which reduces the decoding time to some extent.

Based on the above two conclusions, this paper proposes a filtering dimension reduction decoding algorithm (FDR), and introduces the design and decoding process of the FDR decoder. Firstly, we define some parameters used in the FDR decoding algorithm.

Definition 3. The encoder takes k original packets $S : S = \{S_1, S_2, \ldots, S_k\}$ and generates n encoded packets Y: $Y = \{Y_1, Y_2, \ldots, Y_i \ldots, Y_n\}$. k original packets are encoded into n encoded packets Y. The degree of encoded packet Y_i is defined as $d(Y_i)$. The FDR algorithm divides the encoded packets into two types, the encoded packets generated by the sender and the quadratic packets generated by XOR operations between encoded packets. Quadratic packets is defined as Y_{sec}.T represents the set of original packets that generate an encoded packet.

According to the degree of encoded packets, we adopt the idea of layering to design the decoder. There are five degree values of the encoded packets, which are $d = 1$, $d = 2$, $d = 3$, $d = 4$, $d = k$. Respectively, the decoder is designed into five layers, l_1, l_2, l_3, l_4, l_k which is illustrated in Fig. 3.

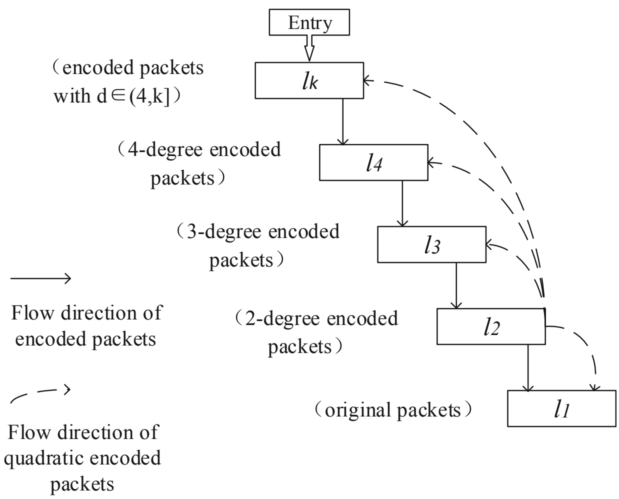

Fig. 3. Design of decoder

Each layer of the decoder stores encoded packets with a degree value equal to the value of the layer. It should be noted that the encoded packets at the layer include not only the encoded packets from the senders, but also the quadratic encoded packets generated by XOR operation from the decoder of the receiving node. For example, l_2 stores all 2-degree encoded packets. One packet in l_2 may be the 2-degree encoded packets received from the sender, or a 2-degree quadratic packet generated by the XOR operation at a higher layer. In addition, it should be noted that l_k stores all encoded packets with d-degree value, where $d \in (4, k]$. The encoded packets are stored in the form of $key - value$. key represents the set T, the IDs of input packets, and $value$ corresponds to the encoded packet. For example, $Y_i = S_1 \oplus S_2 \oplus S_3$, and the ID of S_1, S_2, S_3 are known to be $\{0,\ 1,\ 2\} - Y_i$. Therefore, the $key - value$ stored is as $\{0, 1, 2\} - Y_i$. The encoded packets go downstairs through the decoder. Once it satisfies the XOR condition with a certain packet at one of the layers, or in other words, once all the original packets that participate in the lower-degree encoded packet are also involved in the higher-degree encoded packet, they do XOR operations. In this way, the encoded packet with a higher degree value is updated to the quadratic packet, and its degree value is reduced. By filtering high-degree encoded packets and reducing their degree values, FDR speeds up the decoding process to a certain extent. It is not difficult to imagine that the encoded packets with higher or lower degree values have a higher probability to do XOR operations with other encoded packets. Theoretically, it is impossible for the key value of each layer to have a true inclusion relationship with one key value at the lower layer except l_1. The FDR decoder is in the decoding state from the receipt of the first encoded packet until the l_1 layer contains all the original packets.

4 Numerical Results

In this section, we evaluate the performance of the FDR decoding algorithm through numerical results. In order to eliminate the impact of packet loss and packet collision on the decoding success probability as much as possible, we focus on single-hop communication between a source node and a destination node. Through NS3 simulations and MATLAB experiments, we compare the FDR algorithm and the RLT algorithm in terms of the successful decoding probability, and demonstrate the stability of the FDR algorithm.

4.1 Simulation Settings and Parameters

In this section, we evaluate the performance of the FDR decoding algorithm through simulation experiments. All simulations are performed using Network Simulator 3, and a two-dimensional regular hexagonal topology of seven nodes is used, as shown in Fig. 4.

In Fig. 4, six nodes are located at the vertex of hexagonal network topology as source nodes. The remaining one is located at the center of the network as a sink node. The direction of data flow is from a source node to the sink node. The communication between a source node and the sink node is single-hop and the transmission range is r. In order to eliminate the packet collision, we control the six source nodes to send packets to the sink node at different time by setting different packet interval.

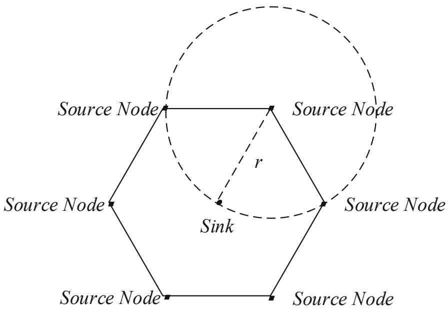

Fig. 4. Hexagonal network topology

The parameters used in the simulation experiment have a big impact on the experimental results. The values of parameters such as "simulation time, bandwidth, transmission power and receiving power" are given in Table 1. The data generated at the application layer by a source node is divided into several blocks, each block composed of about k data packets (here $k = 60$). Each data packet consists of three parts: head fields, load and FCS check.

Table 1. Experimental parameters.

Parameters	Value
Simulation time/s	6000
Size of data block/B	2975–8955
Length of load/B	200
Bandwidth/kbps	10
Route protocol	LB-AGR
MAC protocol	RCHF
Transmitting power/W	2
Receiving power/W	0.75
Range/m	1500

4.2 Simulation Results

The successful decoding probability is defined in formula (7), where $N_{total-trans}$ represents the total number of times that a source node sends data packets to the sink node, and $N_{retrans}$ denotes the number of times that the sink node restores the original data packets successfully. The difference between $N_{total-trans}$ and $N_{retrans}$ is $N_{one-time-succ}$, which represents the number of times that the sink node decodes successfully at the first time. The successful decoding probability of one single hop is defined as $N_{one-time-succ}$ divided by $N_{total-trans}$.

$$P_{Dec-succ-hop} = \frac{N_{Total-trans} - N_{retrans}}{N_{Total-trans}} = \frac{N_{one-time-succ}}{N_{Total-trans}} \quad (7)$$

The simulation results are shown in Fig. 5. The horizontal axis of Fig. 5 represents the number of encoded packets sent out by a source node each time, and the vertical axis represents the successful decoding probability of one transmission. In the experiment, 4 × 4 sets of experimental data using the FDR algorithm with different *k* values are also counted in $N_{total-trans}$ and $N_{retrans}$, as shown in Fig. 6(a), (b), (c), and (d). Figure 5 shows the successful decoding probability of one transmission with the FDR algorithm and the RLT algorithm, respectively.

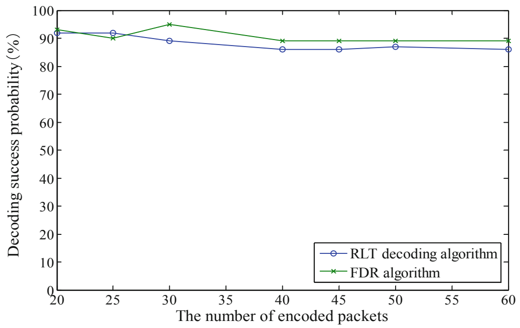

Fig. 5. Comparison in terms of the successful decoding probability

It can be seen that the successful decoding probability with the FDR algorithm is generally higher than that with the RLT algorithm. When the number of encoded packets is small, such as n = 20, the successful decoding probability with the two decoding algorithms is almost equal, e.g., RLT reaches 92% and FDR can reach 93%. The mutation occurs at n = 25, where the successful decoding probability with FDR is 90%, slightly lower than that with RLT, which is 92%. When n = 30, the performance of the FDR algorithm is significantly better than that of RLT, i.e., the former is 95% and the latter 89%. When the number of encoded packets exceeds 40, the successful decoding probability with the RLT algorithm is basically 86%, while that with the FDR algorithm is about 89%. Generally speaking, the performance of the FDR algorithm is better than that of the RLT algorithm.

Figure 6(a) shows four groups of experimental data when the number of original packets *k* is 25 and the number of encoded packets *n* is 32. Take group 3 experiment as an example. The total number of times that a source node sends data packets to the sink node is 143, but only 10 of them are the number of times with secondary successful decoding.

Figure 6(b) shows four groups of experimental data when the number of original packets is 30 and the number of encoded packets is 38. Figure 6(c) shows four groups of experimental data when the number of original packets is 35 and the number of encoded packets is 44. Figure 6(d) shows four groups of experimental data when the

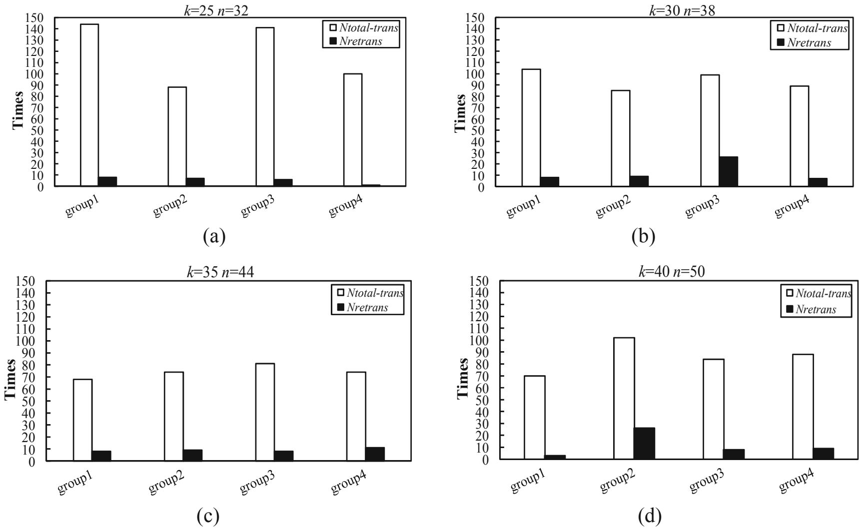

Fig. 6. $N_{total-trans}$ and $N_{retrans}$ under different k by the FDR algorithm

number of original packets is 40 and the number of encoded packets is 50. The statistical data are consistent with the trend of the successful decoding probability in Fig. 5.

5 Conclusions

In this paper, we proposed a filtering dimension reduction (FDR) decoding algorithm for underwater acoustic networks (UANs), which is more suitable for reliable transmission of underwater communication. The FDR algorithm executes XOR operations on the encoded packets to reduce the dimensions of the encoded packets or directly generate some 1-degree packets. As a result, the FDR algorithm can increase the number of 1-degree encoded packets and reduce the decoding complexity. Moreover, the FDR algorithm can eliminate the waiting time for a traditional decoding algorithm to receive the 1-degree packets, and achieve fast decoding. The simulation results show that the successful decoding probability with the FDR algorithms is higher than that with the FDR algorithm.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (Grant No. 61962052), the IoT Innovation Team Foundation of Qinghai Office of Science and Technology, the Key Lab of IoT of Qinghai, the Hebei IoT Monitoring Center (No. 3142016020) and Key Projects of Hebei Province (Grant No. 19270318D).

References

1. Hao, K., Jin, Z.G., Shen, H.F.: An efficient and reliable geographic routing protocol based on partial network coding for underwater sensor networks. Sensors **15**(6), 12720–12735 (2015)
2. Du, X.J., Huang, K.J., Lan, S.L., Feng, Z.X., Liu, F.: LB-AGR: level-based adaptive geo-routing for underwater sensor network. J. China Univ. Posts Telecommun. **21**(1), 54–59 (2014)
3. Wang, Y., Li, Q., Zhang, G.P., Liang, G.L., Ma, S.L.: Phase-difference ambiguity resolution for USBL underwater acoustic positioning systems. Acta Electron. Sin. **45**(11), 2787–2794 (2017)
4. Du, X.J., Peng, C.Y., Liu, X.X.: Hierarchical code assignment algorithm and state-based CDMA protocol for UWSN. China Commun. **12**(3), 50–61 (2015)
5. Jiang, P., Feng, Y., Wu, F.: Fruit fly inspired underwater sensor network deployment algorithn. Acta Electron. Sin. **45**(6), 1403–1407 (2017)
6. Zhang, X., Chang, Y., Song, K., Li, C.G., Yang, L.X.: A novel parameter estimation algorithm for underwater acoustic channels. Acta Electron. Sin. **47**(2), 509–512 (2019)
7. Liu, K.N., Zhao, Z.H., Wang, Y.: AdaCode: adaptive codes in IEEE 802.11n wireless long-distance links. Acta Electron. Sin. **46**(12), 2942–2949 (2018)
8. Ma, H.Y., Fan, X.K., Wang, S.M., Dong, Y.M.: Secure hierarchical hybrid encryption communication protocol based on quantum teleportation for underwater sensor networks. J. Softw. **25**(1), 39–46 (2014)
9. Zhu, J., Liu, J., Zhao, H., Xu, Y.: Research on reliable protocol based on randon-walk model in UWSN. Chin. J. Comput. **39**(5), 1007–1020 (2016)
10. Du, X., Li, K., Liu, X.: RLT code based handshake-free reliable MAC protocol for underwater sensor networks. J. Sens. **15**, 12720–12735 (2016)
11. Guo, Z., Xie, P., Cui, J.H., Wang, B.: On applying network coding to underwater sensor networks. In: Proceedings of the 1st ACM International Workshop on Underwater Networks, Los Angeles, CA, USA, p. 109 (2006)
12. Liu, B., Garcin, F., Ren, F.: A study of forward error correction schemes for reliable transport in underwater sensor networks. In: Proceedings of IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pp. 197–205 (2008)
13. Xie, P., Zhou, Z., Peng, Z., Cui, J.H., Shi, Z.: SDRT: a reliable data transport protocol for underwater sensor networks. Ad Hoc Netw. **8**(7), 708–722 (2010)
14. Liu, B., Chen, H., Lei, X., Ren, F., Sezaki, K.: Internode distance-based redundancy reliable transport in underwater sensor networks. EURASIP J. Wirel. Commun. Netw. **2010**, 1–16 (2010)
15. Mo, H., Peng, Z., Zhou, Z., Zuba, M.: Coding based multi-hop coordinated reliable data transfer for underwater acoustic networks: design, implementation and tests. In: Globecom Workshops (2013)