

基于混合纠错技术的北斗长报文可靠传输机制设计

张志峰, 李中学, 阮博

(陆军勤务学院 军事物流系, 重庆 401311)

摘要: 北斗卫星导航系统的短报文通信功能在传输容量上具有一定的限制, 并且无回执信息, 属于不可靠的通信方式; 为了提高长数据报文的传输可靠性, 通过对传输协议的改进, 提出了一套基于混合纠错技术的长报文可靠传输机制; 通过前向纠错技术和自动重传请求技术的结合, 能够对一定范围内的出错情况进行自纠正, 并对超出自纠能力的传输出错进行反馈重传, 从而提高传输的可靠性。

关键词: 长数据报文; 混合纠错技术; 传输协议; 可靠传输机制

Design of Reliable Transmission Mechanism for Beidou Long Message Based on Hybrid Error Correction Technology

Zhang Zhifeng, Li Zhongxue, Ruan Bo

(Dept. of Military Logistics, Army Logistics University of PLA, Chongqing 401311, China)

Abstract: The short message communication function of Beidou Satellite Navigation System has some limitations in transmission capacity, and there is no receipt information, which is an unreliable communication mode. In order to improve the transmission reliability of long data message, a set of reliable transmission mechanism of long message based on hybrid error correction technology is proposed by improving the transmission protocol. Through the combination of forward error correction and automatic retransmission request technology, it can correct the error in a certain range, and retransmit the transmission error beyond the self correction ability, thus improving the reliability of the transmission.

Keywords: long data message; hybrid error correction technology; transmission protocol; reliable transmission mechanism

0 引言

北斗短报文通信功能作为北斗卫星导航系统所特有的功能, 在各行业领域都有着广泛的研究和应用^[1]。由于北斗传输能力为 1 680 bit/次, 对于长数据报文不能一次发送完毕, 需要进行分包处理^[2-3]。据统计分析, 北斗短报文通信单数据包的传输成功率为 95.5%, 当报文分包数达到 6 包时, 实际测试的传输成功率只有 75.1%^[4]。因此, 需要设计一个合理的传输机制, 提高长报文传输的可靠性。

目前, 通信系统中解决数据丢包问题应用比较广泛的两种方法是自动重传请求 (ARQ) 技术和前向纠错 (FEC) 技术^[5-6]。ARQ 技术对于信道的占用率比较高, FEC 技术纠错能力具有一定的限值且无回执信息, 应用于北斗长报文通信的可靠传输中都具有一定的缺陷。因此, 本文采用混合纠错 (HEC) 技术将两者进行结合, 通过 FEC 技术进

行纠错的同时, 运用 ARQ 技术发送反馈信息, 并对 FEC 技术无法纠错的情况通过数据重传的方法解决^[7]。

在 FEC 技术的应用上, 本文所采用的是 Reed-Solomon (RS) 纠删码, 它的纠错能力很强, 可以纠正多个随机错误, 并且编码简单, 构造方便^[8]。在 ARQ 技术的应用上, 结合 RS 纠删码工作的特点, 以发送组为单元发送反馈信息并进行相应的重传操作, 减少了反馈的次数, 节约了信道空间。整个传输机制的设计思路是运用 HEC 技术, 在纠错能力范围内时进行自动纠错, 超过了纠错能力范围则通过反馈信息进行重传, 从而实现北斗长报文的可靠传输。

1 RS 编码技术

RS 编码技术编码过程如下: 假设要发送 k 个原始数据包, 通过 RS 编码后会生成 m 个冗余数据包, 将 $k+m$ 个数据包作为一个发送组, 令 $n=k+m$, 记作 RS (n, k)。在接收端只要任意接收到 k 个数据包, 就能够通过运算还原出发送组中的原始数据, 发送组中允许丢失的数据包不大于 $n-k$ 个^[9-11]。

1.1 编码

如图 1 所示, 将 RS 编码看作是矩阵运算。编码过程可以表示为 $C=G_0 * D$, 其中生成矩阵 G_0 是由 k 维单位矩阵 I 和 $m * k$ 维的范德蒙德矩阵 G 组成的 $(m+k) * k$ 维矩阵,

收稿日期: 2018-05-23; 修回日期: 2018-07-16。

基金项目: 军委后勤保障部综合计划局重点科研项目 (BS215R012)。

作者简介: 张志峰 (1993-), 男, 山东聊城人, 在读硕士研究生, 主要从事信息系统与应用方向的研究。

李中学 (1965-), 男, 四川岳池人, 博士, 教授, 硕士研究生导师, 主要从事信息系统与应用方向的研究。

编码前矩阵 D 由 k 个原始数据包 $d_1, d_2, d_3, \dots, d_k$ 组成, 编码后矩阵 C 由 k 个原始数据包和 m 个冗余数据包 $c_1, c_2, c_3, \dots, c_m$ 组成。

$$\begin{matrix} I \leftarrow \\ \\ G \leftarrow \end{matrix} \begin{matrix} \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \\ \dots \\ \begin{matrix} 0 & 0 & 0 & 0 \\ G_{11} & G_{12} & G_{13} & G_{14} \\ G_{21} & G_{22} & G_{23} & G_{24} \\ \dots & \dots & \dots & \dots \\ G_{m1} & G_{m2} & G_{m3} & G_{m4} \end{matrix} \end{matrix} \times \begin{matrix} \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix} \\ \dots \\ \begin{matrix} 1 \\ G_{1k} \\ G_{2k} \\ \dots \\ G_{mk} \end{matrix} \end{matrix} = \begin{matrix} \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ \dots \\ d_k \end{matrix} \\ \dots \\ \begin{matrix} c_1 \\ c_2 \\ \dots \\ c_m \end{matrix} \end{matrix}$$

图 1 RS 编码

1.2 解码

如图 2 所示, 接收端收到任意不少于 k 个数据包时, 取其中的 k 个数据包组成接收矩阵 C' , 根据矩阵 C' 可以知道接收到的数据包在发送组中的位置, 并从生成矩阵 G_0 中提取出对应的行, 组成新的矩阵 G_0' , 则有等式 $C' = G_0' * D$ 成立。由此可推出 $D = (G_0')^{-1} * C'$ 成立, 如图 3 所示, 从而实现了原始数据包中丢失数据包的恢复, 其中 $(G_0')^{-1}$ 为 G_0' 的逆矩阵。

$$\begin{matrix} I \leftarrow \\ \\ G \leftarrow \end{matrix} \begin{matrix} \begin{matrix} - & - & - & - \\ 0 & 1 & 0 & 0 \\ - & - & - & - \\ 0 & 0 & 0 & 1 \end{matrix} \\ \dots \\ \begin{matrix} 0 & 0 & 0 & 0 \\ G_{21} & G_{22} & G_{23} & G_{24} \\ \dots & \dots & \dots & \dots \\ G_{m1} & G_{m2} & G_{m3} & G_{m4} \end{matrix} \end{matrix} \times \begin{matrix} \begin{matrix} - \\ 0 \\ - \\ 0 \end{matrix} \\ \dots \\ \begin{matrix} 1 \\ G_{2k} \\ \dots \\ G_{mk} \end{matrix} \end{matrix} = \begin{matrix} \begin{matrix} - \\ d_2 \\ - \\ d_4 \\ \dots \\ d_k \end{matrix} \\ \dots \\ \begin{matrix} c_2 \\ \dots \\ c_m \end{matrix} \end{matrix}$$

图 2 接收端接收情况

$$\begin{matrix} \begin{matrix} \times & \times \\ 1 & 0 \\ \times & \times \\ 0 & 1 \end{matrix} \\ \dots \\ \begin{matrix} 0 & 0 \end{matrix} \end{matrix} \times \begin{matrix} \begin{matrix} \times & \times \\ 0 & 0 \\ \times & \times \\ 0 & 0 \end{matrix} \\ \dots \\ \begin{matrix} 1 & 0 \end{matrix} \end{matrix} = \begin{matrix} \begin{matrix} d_2 \\ d_4 \\ \dots \\ d_k \\ c_2 \\ \dots \\ c_m \end{matrix} \\ \dots \\ \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ \dots \\ d_k \end{matrix} \end{matrix}$$

图 3 RS 解码

1.3 范德蒙德 (Vandermonde) 矩阵

由 RS 解码的过程可知, 要使解码成功, G_0' 必须为 k 维可逆矩阵, 所以要求生成矩阵 G_0 的任意一个 $k * K$ 维子矩阵必须可逆。由于矩阵 I 为单位阵, 必然存在逆矩阵, 因此要求矩阵 G 存在逆矩阵。在线性代数中有一种矩阵称为范德蒙德矩阵, 它的任意的子方阵均为可逆方阵。一个 m 行 n 列的范德蒙德矩阵定义如式 (1) 所示, 其中 a_i 均不相同, 且不为 0。

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ a_1^1 & a_2^1 & a_3^1 & \dots & a_n^1 \\ a_1^2 & a_2^2 & a_3^2 & \dots & a_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1^{m-1} & a_2^{m-1} & a_3^{m-1} & \dots & a_n^{m-1} \end{bmatrix} \quad (1)$$

令 $a_1, a_2, a_3 \dots a_n$ 分别为 $1, 2, 3 \dots n$, 则得到范德蒙德矩阵为:

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{m-1} & 3^{m-1} & \dots & n^{m-1} \end{bmatrix}$$

2 传输机制设计

2.1 设计思路

如图 4 所示, 发送端对于长报文首先进行分包处理, 将其切分成符合北斗发送规则的数据包; 然后对数据包进行分组处理, 将其按照发送组中原始数据包的个数 k 进行分组, 以便按照 RS 编码技术对数据包进行处理; 最后对划分成组的数据包进行 RS 编码, 生成 m 个冗余数据包, 并和 k 个原始数据包组成一个发送组进行发送。

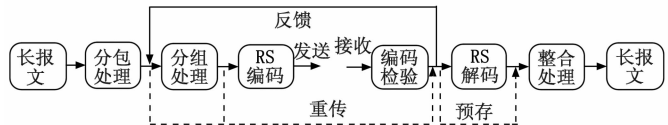


图 4 设计思路

接收端在接收完一个发送组的数据包后, 首先对接收到的数据包进行编码检验, 并根据检验结果向发送端发送反馈信息; 然后根据检验结果决定接下来的处理, 当丢失的数据包数在纠错能力之内时, 通过 RS 解码技术进行数据包的还原并存放于接收端存储空间中, 当超过纠错范围时, 将已经接收到的原始数据包存放在接收端存储空间中, 并对丢失的原始数据包进行重传处理; 最后将存储空间中的原始数据包进行整合处理, 还原出长报文内容。

2.2 传输协议设计

为了便于传输机制的工作, 在北斗短报文现有通信协议的基础上需要添加必要的传输约定关键字进行协议拓展, 如图 5 所示。

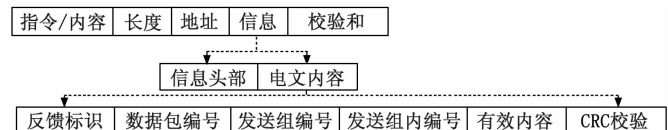


图 5 协议拓展结构

具体定义如下:

- 1) 反馈标识。占用 1 个比特, 用于表示该数据包是否为反馈数据包。设定若反馈标识为 0, 则为反馈数据包; 若反馈标识为 1, 则不是反馈数据包。由于北斗短报文通信本质上为不可靠通信方式, 为了提高通信可靠性, 必不可少的需要添加反馈信息。
- 2) 数据包编号。占用 1 个字节, 用于对分包后的数据

包进行编号。长报文数据经过分包处理后, 在接收端需要重新整合, 为保证数据整合的完整性, 按照先后顺序对数据包进行编号, 整合时按照编号顺序进行。

3) 发送组编号。占用 4 个比特, 用于对发送组进行编号。本文将几个数据包看作是一个发送组进行处理, 因此会形成多个发送组, 为避免接收端混淆组内信息, 需要对每一个发送组进行编号。

4) 发送组内编号。占用 3 个比特, 用于对一个发送组内的数据包进行编号。本文采用的是 RS (8, 5) 编码, 每个发送组内有 8 个数据包, 正好可用 3 个比特位编号。

5) CRC 校验。占用 1 个字节, 用于对数据包内数据进行检验。为了保证接收端接收到的数据包在传输过程中数据没有出错, 需要进行 CRC 校验, 只有检验没有错误的数据包解码才有意义。

2.3 发送端处理

如图 6 所示, 发送端对于需要发送的长数据报文, 首先根据所用设备的传输容量限值以及添加的关键字大小, 确定分包数据量的大小, 对长报文进行分包处理。然后依照 RS 编码的规则, 以 5 个原始数据包为一组, 对数据包进行分组处理, 以便进行 RS 编码的操作。最后以发送组为单位, 进行 RS 编码处理后在发送组中加入 3 个冗余数据包, 并将处理好的发送组发送出去。

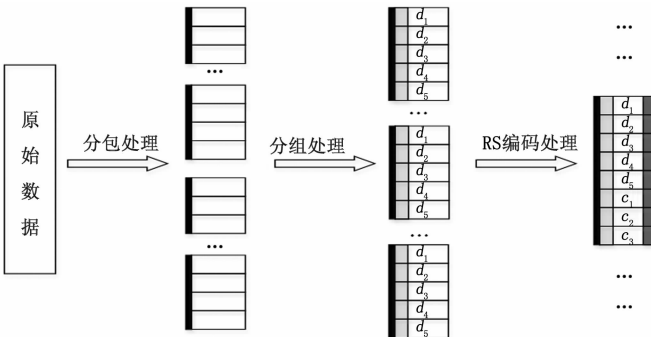


图 6 发送端处理过程

本文所采用的是 RS (8, 5) 编码方式, 即每个发送组中包含 5 个原始数据包和 3 个冗余数据包, 接收端只要接收到其中的任意 5 个数据包即可通过 RS 解码技术还原出所有的原始数据包。按照这种方法进行 RS 编码的空间冗余率为 1.6, 在牺牲一部分的传输信道空间的前提下, 能够为北斗短报文通信功能增加一定的传输纠错能力。

值得注意的是, 在分包的过程中, 对于末尾数据包数据量不足的情况用 0 补足, 保持分包后每个数据包内的数据量相同, 以便进行 RS 编码处理。由于 RS 编码的丢包恢复能力有限, 本文中只允许每个发送组中丢失的数据包数不大于 3 个, 否则就超出了纠错范围。为了保证数据的可靠传输, 在这种情况下就需要根据反馈数据包所反馈回来的信息进行数据包的重传。

2.4 接收端处理

接收端在接收到数据包后, 首先通过 CRC 检验码对数据包编码进行检验, 对于检验出错的数据包没有解码的必要, 将其按照数据包丢失的情况进行处理。根据检验结果

决定数据包进行 RS 解码还是反馈重传。

当一个发送组内接收到的数据包数不少于 5 个时, 反馈数据包向发送端发送该发送组发送成功的消息, 并进行 RS 编码处理。当接收到的数据包恰好包含所有原始数据包 d_1, d_2, d_3, d_4, d_5 时, 不必进行 RS 解码, 直接按照发送组内编号进行数据包整合即可。当接收到的数据包中原始数据包出现丢包现象时, 选取所有正确接收到的原始数据包和相应数量的冗余数据包, 组成一个由 5 个数据包组成的数据包组, 通过 RS 解码技术还原出所有的原始数据包, 进而进行数据包整合处理。

当一个发送组内接收到的数据包数少于 5 个时, 超过了 RS 编码的纠错范围, 此时就需要通过反馈重传对数据包进行重新发送。根据发送组内编号确定接收成功的数据包的类型。对于接收成功的原始数据包暂时保存在接收端存储空间中, 以便进行后续的数据包整合处理, 对于接受成功的冗余数据包, 由于所在的发送组不会进行 RS 编码处理, 因此没有存在的价值, 直接删除即可。

对于丢失的原始数据包, 在反馈信息中说明其数据包编号即可, 当发送端接收到反馈数据包后, 在存储空间中找到相对应的原始数据包, 重新进行分组以及 RS 编码后通过另一个发送组重新进行发送。特殊情况下, 当一个发送组内的 5 个原始数据包全部丢失时, 可以在反馈信息中说明发送组编号, 接收端接收到反馈数据包后, 在存储空间中找到相对应的发送组, 对整个发送组进行重传, 这样就不必重新进行分组以及 RS 编码, 减少对数据包的处理。

3 实验结果与分析

本文采用 matlab 软件对所提出的传输机制进行模拟仿真, 主要是验证 RS 纠错码对于数据的恢复效果, 其主要的编程及解释如下所示。

1) 本文采用的是 RS (8, 5) 纠错码, 编码前矩阵 D 由 d_1, d_2, d_3, d_4, d_5 组成, 用语句 “ $d = 1 + \text{round}(7 * \text{rand}(1, 30))$ ” 生成一个包含 30 个元素的行向量, 元素为 1-8 之间的任意整数, 以此来模拟分包后的数据包。

% 编码前矩阵 D

$d_1 = 1 + \text{round}(7 * \text{rand}(1, 30))$;

$d_2 = 1 + \text{round}(7 * \text{rand}(1, 30))$;

$d_3 = 1 + \text{round}(7 * \text{rand}(1, 30))$;

$d_4 = 1 + \text{round}(7 * \text{rand}(1, 30))$;

$d_5 = 1 + \text{round}(7 * \text{rand}(1, 30))$;

$D = [d_1; d_2; d_3; d_4; d_5]$

2) 只有当正确接收到的数据包数不少于 5 个时, RS 纠错码技术才能够使用, 本实验模拟的是不包含特殊情况的需要通过 RS 解码方能回复原始数据的情况。通过 “randperm” 语句产生任意排序的 8 个整数, 代表着一个发送组内的 8 个数据包, 然后取前 6 个数表示接收端正确接收到的 6 个数据包, 再通过对这 6 个数的升序排列, 取前 5 个数所对应的行向量组成接收矩阵 C 。以及生成矩阵中形成的新矩阵 g_0 。通过这一操作可以保证随机性的要求, 囊括了所有可能出现的情况, 使得模拟结果更具说服力。

% 模拟正确接收到任意六个数据包的情况

```

out=randperm(8)%返回一行包含从 1 到 8 的整数
    %完成的是不重复的重排采样
data1=out(1:6)
data=sort(data1)%升序排列
C1=C(data(1),:)
C2=C(data(2),:)
C3=C(data(3),:)
C4=C(data(4),:)
C5=C(data(5),:)
C0=[C1;C2;C3;C4;C5] %接收矩阵 C0
g1=G0(data(1),:)
g2=G0(data(2),:)
g3=G0(data(3),:)
g4=G0(data(4),:)
g5=G0(data(5),:)
g0=[g1;g2;g3;g4;g5] %从生成矩阵 G0 中取出相应的行,
组成新矩阵 g0
    
```

3) 按照 RS 纠错码的工作原理还原出编码前矩阵 d ，并比较还原出的矩阵和编码前矩阵是否一致，以验证 RS 编码的效果。

```

g=inv(g0) %对新矩阵 g0 取逆
d=g * C0 %还原出编码前矩阵
if sum(d==D)~=0
    display('数据恢复完全')
else
    display('数据未能恢复完全')
end
    
```

通过实验可以看出 RS 纠错码技术在满足应用条件的情况下对于数据能够很好的进行恢复。但是值得注意的是，本实验仅仅是对于 RS 编码的可行性进行了简单的模拟，受限于设备的原因，并没有对整个的可靠传输机制进行实验，因此本实验还有很大的改进之处。

4 传输机制的改进

4.1 生成矩阵的改进

生成矩阵 G_0 中的矩阵 G 考虑到必须存在逆矩阵，本文采用的是范德蒙德矩阵，其实还可选用柯西矩阵（如式（2）所示）。范德蒙德矩阵求逆运算的复杂度为 $o(n^3)$ ，相比而言，柯西矩阵求逆运算的复杂度为 $o(n^2)$ ，并且柯西矩阵通过有限域转换，将伽罗华有限域 $GF(2^w)$ 域中的元素转换成二进制矩阵，降低了乘法运算的复杂度。

$$\begin{bmatrix}
 \frac{1}{x_0 + y_0} & \frac{1}{x_0 + y_1} & \frac{1}{x_0 + y_2} & \dots & \frac{1}{x_0 + y_n} \\
 \frac{1}{x_1 + y_0} & \frac{1}{x_1 + y_1} & \frac{1}{x_1 + y_2} & \dots & \frac{1}{x_1 + y_n} \\
 \frac{1}{x_2 + y_0} & \frac{1}{x_2 + y_1} & \frac{1}{x_2 + y_2} & \dots & \frac{1}{x_2 + y_n} \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 \frac{1}{x_m + y_0} & \frac{1}{x_m + y_1} & \frac{1}{x_m + y_2} & \dots & \frac{1}{x_m + y_n}
 \end{bmatrix} \quad (2)$$

4.2 数据包交织编码

北斗短报文通信受工作环境等因素的影响，有时候出现丢包的情况呈现出连续性的特点，采用 RS 编码提高容错

性并不能达到理想的效果。为了弥补 RS 纠错码对于连续性丢包问题恢复能力差的弊端，最大限度的使用 FEC 技术进行数据的恢复，考虑在 RS 编码的基础上进行发送组之间的交织编码。通过将连续丢包的情况离散分布到不同的发送组上，进一步提高 RS 纠错码的容错性能。由于发送组间交织时并没有添加多余的冗余信息，因此不会降低发送组的编码效率。

交织编码根据交织方式的不同，可分为线性交织、卷积交织和伪随机交织。其中线性交织编码是一种比较常见的形式。所谓线性交织编码器，是指把纠错编码器输出信号均匀分成 m 个码组，每个码组由 n 段数据组成，这样就够成一个 $n \times m$ 的矩阵，称为交织矩阵。如图 9 所示，数据以横排顺序存入，经过交织处理后以竖列的顺序输出，这样就完成了对数据的交织编码。在接收时通过上述过程的逆向重复，即先按竖列存入存储器，再横排读出，就可恢复成原来的 RS 编码，但在传输时的突发错误被分散了。

5 结束语

本文使用混合纠错技术来达到提高北斗长报文传输可靠性的目的，通过理论分析以及模拟实验的验证，证明了其可行性。通过自动重传请求技术和前向纠错技术的综合应用，在提高通信功能容错性能的同时，尽可能节省了传输信道空间，对于北斗短报文功能的广泛应用具有一定的贡献。

值得注意的是，本文所提出的可靠传输机制仅适用于长数据报文传输的情况，由于增加了冗余信息，对信道的占用率会增加，当所需传输的数据信息并不多时，相比于传统的反馈重传方式其优势并不明显。因此下一步考虑将可靠传输机制应用到短数据报文的传输中，以及运用相应的压缩技术对数据进行处理，以便进一步节约传输信道空间。

参考文献:

- [1] 李源, 钱向阳, 李绍雄. 北斗卫星导航系统的发展现状与前景 [J]. 通讯世界, 2016 (14): 217-218.
- [2] 龚振宇. 北斗远程通信的设计与实现 [D]. 成都: 电子科技大学, 2015.
- [3] 颜晓星, 车明, 高小娟. 基于北斗卫星的可靠远程通信系统设计 [J]. 计算机工程, 2017, 43 (3): 62-68.
- [4] 谷军霞, 王春芳, 宋之光. 北斗短报文通信信道性能测试与统计分析 [J]. 气象科技, 2015, 43 (3): 458-463.
- [5] 唐懿芳, 钟达夫. 基于数据冗余的 BDS 长报文传输机制改进算法 [J]. 指挥控制与仿真, 2016, 38 (1): 61-65.
- [6] 成方林, 张翼飞, 刘佳佳. 基于“北斗”卫星导航系统的长报文通信协议 [J]. 海洋技术, 2008 (1): 26-28+31.
- [7] 林晓峰. 自适应 FEC 丢包恢复技术的研究 [D]. 北京: 北京邮电大学, 2010.
- [8] 刘琴. 基于 RS 纠错码的数据安全存储与安全删除方法研究 [D]. 武汉: 武汉理工大学, 2015.
- [9] 徐胜霞. RS 码技术的研究 [D]. 南京: 南京邮电大学, 2012.
- [10] 刘会. 单向链路下高可靠数据传输方法研究 [D]. 北京: 北京邮电大学, 2017.
- [11] 周亮. 分布式存储系统上的 RS 纠错码研究与应用 [D]. 成都: 成都理工大学, 2017.