

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2019 年 1 月 17 日 (17.01.2019)



(10) 国际公布号
WO 2019/011219 A1

(51) 国际专利分类号:
H04L 1/00 (2006.01)

(21) 国际申请号: PCT/CN2018/095051

(22) 国际申请日: 2018 年 7 月 9 日 (09.07.2018)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201710561850.2 2017年7月11日 (11.07.2017) CN
201710973473.3 2017年10月18日 (18.10.2017) CN

(71) 申请人: 上海交通大学 (SHANGHAI JIAO TONG UNIVERSITY) [CN/CN]; 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。

(72) 发明人: 徐异凌 (XU, Yiling); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。 张文军 (ZHANG, Wenjun); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。 柳宁 (LIU, Ning); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。 管云峰 (GUAN, Yunfeng); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。 王延峰 (WANG, Yanfeng); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。 孙军 (SUN, Jun); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。 姜志乾 (JIANG, Zhiqian); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。 陈浩 (CHEN, Hao); 中国上海市闵行区东川路800号, Shanghai 200240 (CN)。

(74) 代理人: 上海汉声知识产权代理有限公司 (SHANGHAI HANGSOME INTELLECTUAL

(54) Title: MEDIA CONTENT-BASED ADAPTIVE METHOD, DEVICE AND SYSTEM FOR FEC CODING AND DECODING OF SYSTEMATIC CODE, AND MEDIUM

(54) 发明名称: 基于媒体内容的自适应系统码FEC编译码方法、装置、系统及介质

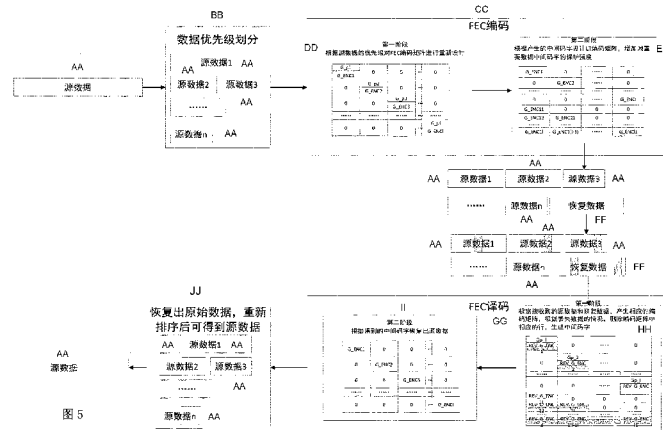


图 5

AA Source data
BB Division of data priorities
CC FEC coding
DD First stage: according to priorities of source data, redesign an FEC coding matrix
EE Second stage: design an LT matrix based on generated intermediate codewords, to increase the protection strength of an intermediate codeword of important data
FF Recovery data
GG FEC decoding
HH First stage: generate a corresponding coding matrix according to the received source data and the recovery data, delete a corresponding line in the coding matrix according to the situation of lost data, and generate intermediate codewords
II Second stage: recover source data according to the obtained intermediate codewords
JJ Recover original data, and rearrange to obtain source data

(57) Abstract: Provided are a media content-based adaptive method, device and system for FEC coding and decoding of a systematic code, and a medium. The method comprises: according to the importance of media content, dividing source data into N types of source data packets according to priorities; generating N types of intermediate codes according to the N types of source data packets and the priorities thereof; according to the N types of intermediate codes, setting recovery data of N types of source data according to a channel

PROPERTY LTD.); 中国上海市闵行区银都路
3828弄56号307室, Shanghai 201108 (CN)。

- (81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。
- (84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

- 包括国际检索报告 (条约第21条(3))。

condition, and generating N types of coded symbols of systematic codes; receiving the coded symbols, and sorting and arranging the coded symbols according to decoding requirements; and according to the number of received coded symbols, decoding same to obtain intermediate codewords according to different situations, and recovering corresponding source data packets according to intermediate codewords. According to the present invention, the quality of the media content is ensured to the greatest extent, and data redundancy caused by FEC is reduced. It is not necessary to shunt a source data stream, which reduces the complexity of FEC coding at a sending end and improves the efficiency of FEC coding. A coding scheme can be dynamically adjusted according to a change in a current network state, so that the adaptability is stronger.

(57) 摘要: 本发明提供了一种基于媒体内容的自适应系统码FEC编译码方法、装置、系统及介质, 包括: 根据媒体内容的重要程度, 将源数据按照优先级划分为N类源数据包; 根据N类源数据包及其优先级, 生成N类中间码; 根据N类中间码, 按照信道状况设定N类源数据的恢复数据, 生成N类系统码的编码符号; 接收所述编码符号, 根据译码需要对编码符号整理排序; 根据接收到的编码符号数量, 分情况译出中间码字, 并根据该中间码恢复出相应的源数据包。本发明最大限度保证媒体内容质量, 减少FEC造成的数据冗余; 不需要对源数据流进行分流, 降低了发送端FEC编码的复杂度, 提高了FEC编码的效率; 可以根据当前网络状态的变化动态的调整编码方案, 具有更强的适应性。

基于媒体内容的自适应系统码 FEC 编译码方法、装置、系统及介质

本申请要求申请日为 2017 年 10 月 18 日、申请号为 CN201710973473.3、专利名称为“基于媒体内容的自适应系统码 FEC 方法、装置及系统”的优先权；

本申请要求申请日为 2017 年 07 月 11 日、申请号为 CN201710561850.2、专利名称为“基于媒体内容的自适应系统码 FEC 编译码方法”的优先权。

技术领域

本发明涉及多媒体传输技术领域，具体地，涉及基于媒体内容的自适应系统码 FEC (Forward Error Correction, 前向纠错) 编译码方法、装置、系统及介质。

背景技术

在异构网络媒体服务系统中，内容通过因特网协议或广播协议分发到终端，因特网中使用 IP/TCP 或 UDP 报文来传输媒体数据，广播通过 MPEG2-TS 来传输内容。UDP 报文在经过多个网络设备后可能会出现丢失，广播 TS 流可能因为传输环境的影响，产生误码，从而造成终端侧的画面破损或者声音停顿。

FEC(Forward Error Correction, 前向纠错)技术是一种广泛应用于通信系统中的编码技术。通过服务器侧对媒体数据进行纠错编码，加入冗余信息一并发送，终端侧进行反向 FEC 解码，对丢失的报文进行恢复。以典型的分组码为例，其基本原理是：在发送端，通过将 k bit 信息作为一个分组进行编码，加入 $(n-k)$ bit 的冗余校验信息，组成长度为 n bit 的码字。码字经过信道到达接收端之后，如果错误在可纠错范围之内，通过译码即可检查并纠正错误 bit，从而抵抗信道带来的干扰，有效降低系统的误码率，提高通信系统的可靠性。但是 FEC 处理，是以冗余开销代价来降低系统的误码率，过度 FEC 编码对系统实时性和网络状态也会造成压力。

不等差错保护 (Unequal Error Protection, UEP) 是联合信源信道编码的一种。其核心思想是，依据码流的各部分数据的重要性不同，对各部分数据采用不同的信道保护机制，即对重要码流进行重点保护。尽管 UEP 降低了非重要码流的抗噪声性能，但有利于系统抗误码总体性能的提升。

作为一种前向纠错编码技术，数字喷泉码（Digital Fountain Code）在传输过程中，不需要反馈及自动重发机制，避免了信号往返的延时以及广播应用中的反馈爆炸问题。数字喷泉的基本思想是：发端将原始数据分割成 k 个数据符号，对这些数据符号进行编码，输出一个任意长度的编码符号码流，接收端只需正确地接收 n （ n 稍大于 k ）个编码符号就可以很大的概率恢复出所有的 k 个数据符号。数字喷泉码本身就具有 UEP 性能，可以实现对不同重要性的数据的保护。

发明内容

针对现有技术中的缺陷，本发明的目的是提供一种基于媒体内容的自适应系统码 FEC 编译码方法、装置、系统及介质。

根据本发明提供的一种基于媒体内容的自适应系统码 FEC 编译码方法，包括：

根据媒体内容的重要程度，将源数据按照优先级划分为 N 类源数据包； N 为大于 1 的整数；

根据源数据和 N 类源数据包的优先级，生成 N 类中间码；

根据所述 N 类中间码，按照信道状况设定 N 类源数据的恢复数据，生成 N 类编码符号；

接收所述编码符号，根据译码需要对所述编码符号进行整理排序，对所述编码符号进行译码处理；

根据接收到的编码符号情况，优先译出优先级高的中间码，并根据该中间码恢复出相应的源数据包。

较佳的，所述根据源数据和 N 类源数据包的优先级，生成 N 类中间码，包括：

构建 N 类源数据包对应的编码矩阵，记为 A_i ， $i=1,2,\dots,N$ ；其中：

$$A_i = \begin{bmatrix} G_{p_i} \\ G_{ENC_i} \end{bmatrix} \quad i = 1, 2, \dots, N$$

式中： A_i 表示第 i 个源数据包对应的编码矩阵， G_{p_i} 表示第 i 个源数据包对应的由 LDPC 矩阵和单位矩阵构成的联合矩阵， G_{ENC_i} 表示第 i 个源数据包对应的 LT 编码矩阵；

根据 N 类源数据包对应的编码矩阵构建源数据的编码矩阵 A ，所述源数据的编码矩阵如下：

$$A = \begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ G_{ENC_1} & 0 & \dots & 0 \\ 0 & G_{p_i} & \dots & 0 \\ 0 & G_{ENC_i} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_N} \\ 0 & 0 & \dots & G_{ENC_N} \end{bmatrix}$$

$$A \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_i \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} 0 \\ D_1 \\ 0 \\ \dots \\ D_i \\ \dots \\ 0 \\ D_N \end{bmatrix}$$

则

$$\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ \dots \\ D_N \end{bmatrix}$$

式中: $C_1, C_2, \dots, C_i, \dots, C_N$ 分别对应 N 类源数据包的中间码, $D_1, D_2, \dots, D_i, \dots, D_N$ 分别对应 N 类源数据包的数据, 上标-1为逆矩阵运算符。

较佳的, 所述根据所述 N 类中间码, 按照信道状况设定 N 类源数据的恢复数据, 包括:

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_N \\ R_1 \\ R_2 \\ \dots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC_1} & 0 & \dots & 0 \\ 0 & G_{ENC_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{ENC_N} \\ G_{ENC_{11}} & 0 & \dots & 0 \\ G_{ENC_{12}} & G_{ENC_{21}} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC_{1N}} & G_{ENC_{2(N-1)}} & \dots & G_{ENC_{N1}} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix}$$

式中: $R_1, R_2, \dots, R_i, \dots, R_N$ 分别对应 N 个源数据包的恢复数据, $G_{ENC_{1i}}$ 表示与 G_{ENC_1} 具有相同生成规则的 LT 编码矩阵; $G_{ENC_{i1}}$ 表示与 G_{ENC_i} 具有相同生成规则的 LT 编码矩阵; $i=1,2,\dots,N$ 。

较佳的, 当 N 的值取 2 时, 针对不同重要程度的数据可以产生相应的中间码如下:

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \end{bmatrix}$$

改变 LT 编码矩阵的结构, 得到 G_ENC_{11} 、 G_ENC_{12} 、 G_ENC_{21} 、 G_ENC_{12} 与 G_ENC_1 的生成方式相同; G_ENC_{21} 与 G_ENC_2 的生成方式相同; G_ENC_{11} 表示与 G_ENC_1 具有相同生成方式的 LT 编码矩阵; G_ENC_{21} 表示与 G_ENC_2 具有相同生成方式的 LT 编码矩阵; G_ENC_{12} 表示与 G_ENC_1 具有相同生成方式的 LT 编码矩阵;

得到的恢复数据 R_1 只与优先级为 1 的数据有关, 而恢复数据 R_2 与优先级 1 和 2 的数据均有关, 则得到:

$$\begin{bmatrix} D_1 \\ D_2 \\ R_1 \\ R_2 \end{bmatrix} = \begin{bmatrix} G_ENC_1 & 0 \\ 0 & G_ENC_2 \\ G_ENC_{11} & 0 \\ G_ENC_{12} & G_ENC_{21} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}。$$

较佳的, 当 N 的值取 2 时, 所述接收所述编码符号, 对所述编码符号进行译码处理, 包括:

在第一阶段解码时, 根据恢复矩阵 W 以及接收到的源数据包得到中间码字; 具体的:

$$W \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ 0 \\ REV_D_2 \\ REV_R_1 \\ REV_R_2 \end{bmatrix}$$

$$W = \begin{bmatrix} G_p_1 & 0 \\ REV_G_ENC_1 & 0 \\ 0 & G_p_2 \\ 0 & REV_G_ENC_2 \\ REV_G_ENC_{11} & 0 \\ REV_G_ENC_{12} & REV_G_ENC_{21} \end{bmatrix}$$

式中: $REV_G_ENC_1$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_1 中丢失包所对应的行后的 LT 编码矩阵; $REV_G_ENC_{11}$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_{11} 中丢失包所对应的行后的 LT 编码矩阵; $REV_G_ENC_{12}$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_{12} 中丢失包所对应的行后的 LT 编码矩阵; $REV_G_ENC_2$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_2 中丢失包所对应的行后的 LT 编码矩阵; $REV_G_ENC_{21}$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_{21} 中丢失包所对应的行后的 LT 编码矩阵;

$$\text{令 } REV_A = \begin{bmatrix} G_p_1 & 0 \\ REV_G_ENC_1 & 0 \\ REV_G_ENC_{11} & 0 \\ 0 & G_p_2 \\ 0 & REV_G_ENC_2 \\ REV_G_ENC_{12} & REV_G_ENC_{21} \end{bmatrix}; \text{ 并根据接收到的对应优先级下源数据}$$

包的数量以及接收到的数据包总数量进行解码；

接收到的编码符号总数小于源数据生成的编码符号的个数，并且 $REV_A_1 = \begin{bmatrix} G_{p_1} \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix}$ 的行数小于列数时； C_1 、 C_2 均无法解出；

接收到的编码符号总数小于源数据生成的编码符号的个数，且 $REV_A_1 = \begin{bmatrix} G_{p_1} \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix}$ 的行数大于等于列数时；若 $REV_A_1 = \begin{bmatrix} G_{p_1} \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix}$ 的行数大于等于列数，则 REV_A_1 可逆，对应的逆矩阵记为 $REV_A_1^{-1}$ ；则可得：

$$[C_1] = REV_A^{-1} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix} \text{ 其中：} C_2 \text{ 无解；}$$

式中： REV_D_1 表示接收到的优先级为1的源数据， REV_R_1 表示接收到的优先级为1源数据的恢复数据；

接收到的编码符号总数大于等于源数据生成的编码符号的个数，且 $REV_A_1 = \begin{bmatrix} G_{p_1} \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix}$ 的行数大于等于列数时；若 $REV_A_1 = \begin{bmatrix} G_{p_1} \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix}$ 的行数大于等于列数，则 REV_A_1 可逆，对应的逆矩阵记为 $REV_A_1^{-1}$ ；若 $REV_A_2 = \begin{bmatrix} G_{p_2} \\ REV_G_ENC_2 \\ REV_G_ENC_{21} \end{bmatrix}$ 的行数大于等于列数，则 REV_A_2 可逆，对应的逆矩阵记为 $REV_A_2^{-1}$ ；若 REV_A_1 和 REV_A_2 都可逆，则 REV_A 可逆，对应的逆矩阵记为 REV_A^{-1} ，记 $B_{1,1} = \begin{bmatrix} 0 \\ 0 \\ REV_G_ENC_{12} \end{bmatrix}$ ，则

$$REV_A^{-1} = \begin{bmatrix} REV_A_1^{-1} & 0 \\ REV_A_2^{-1} B_{1,1} REV_A_1^{-1} & REV_A_2^{-1} \end{bmatrix}; \text{ 得到:}$$

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = REV_A^{-1} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix} = \begin{bmatrix} REV_A_1^{-1} & 0 \\ REV_A_2^{-1} B_{1,1} REV_A_1^{-1} & REV_A_2^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix};$$

若 $REV_A_2 = \begin{bmatrix} G_{p_2} \\ REV_G_ENC_2 \\ REV_G_ENC_{21} \end{bmatrix}$ 的行数小于列数，则 REV_A_2 不存在逆矩阵，只能求出

$$[C_1] = REV_A^{-1} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix};$$

式中: REV_D_2 表示接收到的优先级为 2 的源数据, REV_R_2 表示接收到的优先级为 1 和 2 源数据的恢复数据;

接收到的编码符号总数大于等于源数据生成的编码符号的个数, 且 $REV_A_1 =$

$$\begin{bmatrix} G_p_1 \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix} \text{的行数小于列数, } REV_A_2 = \begin{bmatrix} G_p_2 \\ REV_G_ENC_2 \\ REV_G_ENC_{21} \end{bmatrix} \text{的行数大于等于列数时;}$$

使用高斯消元法直接对矩阵 REV_A 进行求解, 即解下列线性方程组:

$$\begin{bmatrix} G_p_1 & 0 \\ REV_G_ENC_1 & 0 \\ REV_G_ENC_{11} & 0 \\ 0 & G_p_2 \\ 0 & REV_G_ENC_2 \\ REV_G_ENC_{12} & REV_G_ENC_{21} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix}$$

得到 C_1 、 C_2 。

较佳的, 由于部分数据发生丢失, 需要根据接收到的编码符号, 生成恢复矩阵, 相当于擦除掉丢失的编码符号对应的矩阵行, 形成相应的恢复矩阵, 记为 REV_G_ENC ; 根据接收的数据包的数目不同, 分不同情况生成中间码字; 具体的, 包括如下步骤:

步骤 A1: 假设接收到 l 类优先级的数据包; 则得到如下的恢复矩阵方程:

$$\begin{bmatrix} G_p_1 & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ 0 & G_p_2 & \dots & 0 \\ 0 & REV_G_ENC_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_p_l \\ 0 & 0 & \dots & REV_G_ENC_l \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ REV_G_ENC_{1l} & REV_G_ENC_{2(l-1)} & \dots & REV_G_ENC_{l1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ 0 \\ REV_D_2 \\ \dots \\ 0 \\ REV_D_l \\ REV_R_1 \\ REV_R_2 \\ \dots \\ REV_R_l \end{bmatrix}$$

对恢复矩阵方程进行初等行变换, 得到:

$$\begin{bmatrix} G_p_1 & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_p_2 & \dots & 0 \\ 0 & REV_G_ENC_2 & \dots & 0 \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_p_l \\ 0 & 0 & \dots & REV_G_ENC_l \\ REV_G_ENC_{1l} & REV_G_ENC_{2(l-1)} & \dots & REV_G_ENC_{l1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \\ \dots \\ 0 \\ REV_D_l \\ REV_R_l \end{bmatrix}$$

式中: $REV_G_ENC_1$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_1 中丢失包

所对应的行后的 LT 编码矩阵; $REV_G_ENC_{11}$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_{11} 中丢失包所对应的行后的 LT 编码矩阵; $REV_G_ENC_{12}$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_{12} 中丢失包所对应的行后的 LT 编码矩阵; $REV_G_ENC_2$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_2 中丢失包所对应的行后的 LT 编码矩阵; $REV_G_ENC_{21}$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_{21} 中丢失包所对应的行后的 LT 编码矩阵; REV_D_1 表示接收到的优先级为 1 的源数据, REV_R_1 表示接收到的优先级为 1 源数据的恢复数据; REV_D_2 表示接收到的优先级为 2 的源数据, REV_R_2 表示接收到的优先级为 1 和 2 源数据的恢复数据; $REV_G_ENC_{1l}$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_{1l} 中丢失包所对应的行后的 LT 编码矩阵, REV_D_l 表示接收到的优先级为 l 的源数据, REV_R_l 表示接收到的优先级为 1 到 l 的源数据的恢复数据, C_l 表示生成的第 l 类中间码;

$$\text{步骤 A2: 令 } REV_A = \begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & REV_G_ENC_2 & \dots & \dots \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_l} \\ 0 & 0 & \dots & REV_G_ENC_l \\ REV_G_ENC_{1l} & REV_G_ENC_{2(l-1)} & \dots & REV_G_ENC_{l1} \end{bmatrix}, \text{ 令}$$

$R(i)$ 为接收到并经扩展后第 i 级编码符号数, $L(i)$ 为编码端每级符号长度, 其中: $1 \leq i \leq l$;

计算每级生成矩阵的秩:

$$\text{eachRank}(i) = \min(\text{Len}(R(i)), \text{sum}(L(1:i)));$$

式中: $\text{eachRank}(i)$ 表示优先级为 i 的数据所对应的子矩阵的秩, 其中 $i=2$ 时所对应

的子矩阵为 $\begin{bmatrix} 0 & G_{p_2} \\ 0 & REV_G_ENC_2 \\ REV_G_ENC_{12} & REV_G_ENC_{21} \end{bmatrix}$, $\text{Len}(R(i))$ 表示接收到的优先级为 i 的编码

符号数的长度, $\text{sum}(L(1:i))$ 表示优先级为 i 的子矩阵的列数, $\min()$ 表示取最小值运算;

计算累积矩阵的秩:

$$\text{cumRank}(i) = \min(\text{cumRank}(i-1) + \text{eachRank}(i), \text{sum}(L(1:i))),$$

式中: $\text{cumRank}(i)$ 表示优先级为 1 到优先级为 i 所对应的子矩阵秩之和,

$\text{cumRank}(i-1)$ 表示优先级为 1 到优先级为 $i-1$ 所对应的子矩阵秩之和;

其中 $\text{cumRank}(1) = \text{eachRank}(1)$;

依次判断 $\text{cumRank}(i) \geq \text{sum}(L(1:i))$: 若成立, $\text{cumFullRank}(i) = 1$; 否则

$\text{cumFullRank}(i) = 0$; $\text{cumFullRank}(i)$ 表示优先级为 1 到优先级为 i 所对应的子矩阵组成的矩阵满秩;

步骤 A3: 根据接收到的每级编码符号数, 分情况分别求中间码 C_1 、 $C_2 \dots C_l$, 设置 i 初始值为 1:

若满足 $\text{cumFullRank}(i) == 1 \& \& \text{cumFullRank}(1:i) > 0$, 使用 RFC 6330 译码方法解出 $C_1 \sim C_i$, 其中: $\& \&$ 表示与运算, $\text{cumFullRank}(1:i)$ 表示从优先级为 1 到优先级为 i 各子矩阵所组成的矩阵满秩; C_1 表示优先级为 1 的数据所对应的中间码字, C_i 表示优先级为 i 的数据所对应的中间码;

$$\text{设 } REV_A_{\Delta i} = \begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & REV_G_ENC_2 \dots & \dots & \dots \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_i} \\ 0 & 0 & \dots & REV_G_ENC_i \\ REV_G_ENC_{1i} & REV_G_ENC_{2(i-1)} & \dots & REV_G_ENC_{i1} \end{bmatrix}, \text{ 得到:}$$

$$REV_A_{\Delta i}^{-1} = \begin{bmatrix} A_1^{-1} & 0 & 0 & 0 \\ X_{1_2} & A_2^{-1} & 0 & 0 \\ \dots & \dots & \dots & 0 \\ X_{1_i} & X_{2_{(i-1)}} & \dots & A_i^{-1} \end{bmatrix}, \text{ 其中:}$$

$$X_{1_2} = A_2^{-1} B_{1_2} A_1^{-1}, \quad X_{a_b} = A_b^{-1} B_{a_b} A_a^{-1} + \sum_{k=a+1}^{b-1} A_b^{-1} B_{k_b} X_{a_k},$$

式中: $REV_A_{\Delta i}$ 表示到优先级为 i 时所有子矩阵组成的满秩矩阵, B_{1_2} 表示子矩阵

$$\begin{bmatrix} 0 \\ 0 \\ REV_G_ENC_{12} \end{bmatrix}, \quad X_{1_2} \text{表示对子矩阵 } B_{1_2} \text{求逆运算后的子矩阵, } X_{a_b} \text{表示子矩阵 } B_{a_b} \text{求逆}$$

运算后得到的子矩阵, B_{a_b} 表示子矩阵 $\begin{bmatrix} 0 \\ 0 \\ REV_G_ENC_{ab} \end{bmatrix}$, a 的取值范围为 $1 \sim i$, b 的取值

范围为 $2 \sim i$, A_i^{-1} 表示子矩阵 $\begin{bmatrix} G_{p_i} \\ REV_G_ENC_i \\ REV_G_ENC_{i1} \end{bmatrix}$ 的逆矩阵;

$$\text{令 } C_{i_temp} = REV_A_i^{-1} \begin{bmatrix} 0 \\ REV_D_i \\ REV_R_i \end{bmatrix}, \text{ 其中, } REV_A_i = \begin{bmatrix} G_{p_i} \\ REV_G_ENC_i \\ REV_G_ENC_{i1} \end{bmatrix}; \quad REV_A_i^{-1} \text{为}$$

REV_A_i 的逆矩阵, 则得到:

$$C_1 = C_{1_temp}$$

$$C_2 = C_{2_temp} + X_{1,2} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix} = C_{2_temp} + A_2^{-1} B_{1,2} C_1$$

$$C_3 = C_{3_temp} + X_{1,3} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix} + X_{2,3} \begin{bmatrix} 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix} = C_{2_temp} + A_3^{-1} B_{1,3} C_1 + A_3^{-1} B_{2,3} C_2$$

根据数学归纳法，得到：

$$C_i = C_{i_temp} + \sum_{j=1}^{i-1} X_{j,i} \begin{bmatrix} 0 \\ REV_D_j \\ REV_R_j \end{bmatrix} = C_{i_temp} + \sum_{j=1}^{i-1} A_i^{-1} B_{j,i} C_j$$

求得 C_1 、 $C_2 \dots C_i$ ，其中，不能解出 $C_{i+1} \dots C_l$ ；

若满足 $\text{cumFullRank}(i) == 1$ ，且不满足 $\text{cumFullRank}(1:i) > 0$ ，使用高斯消元译码方法解出 $C_1 \sim C_i$

$$\text{设 } REV_A_{\Delta i} = \begin{bmatrix} G_p_1 & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_p_2 & \dots & 0 \\ 0 & REV_G_ENC_2 \dots & \dots & \dots \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_p_i \\ 0 & 0 & \dots & REV_G_ENC_i \\ REV_G_ENC_{1i} & REV_G_ENC_{2(i-1)} & \dots & REV_G_ENC_{i1} \end{bmatrix}, \text{ 通过高斯消}$$

元法解线性方程组：

$$\begin{bmatrix} G_p_1 & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_p_2 & \dots & 0 \\ 0 & REV_G_ENC_2 \dots & \dots & \dots \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_p_i \\ 0 & 0 & \dots & REV_G_ENC_i \\ REV_G_ENC_{1i} & REV_G_ENC_{2(i-1)} & \dots & REV_G_ENC_{i1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_i \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \\ \dots \\ 0 \\ REV_D_i \\ REV_R_i \end{bmatrix}$$

求得 C_1 、 $C_2 \dots C_i$ ，其中，不能解出 $C_{i+1} \dots C_l$ ；

对于任一 i 值，若不满足 $\text{cumFullRank}(i) == 1$ 则相应 C_i 不能解出；直到 $i == 1$ ，也不满足 $\text{cumFullRank}(1:i) > 0$ 时，则 C_1 、 $C_2 \dots C_l$ 全部不能解出，译码失败；

步骤 A4：令 i 的值自减 1，返回执行步骤 A3。

较佳的，还包括：

解析源数据包中每个子数据包包头的重要性信息；

根据所述重要性信息，将所述源数据包中每个子数据包分别映射到不同的优先级中；

按照子数据包的优先级重新排列子数据包，得到中间数据包；

将所述中间数据包转化为多个符号 symbol，并对转化成多个符号 symbol 的中间数据包进行前向纠错 FEC 编码后，得到包含多个符号 symbol 的恢复数据包；

在所述恢复数据包中加入索引字段和/或优先级指示字段，所述索引字段用于指示恢复数据包中子数据包的顺序和/或指示恢复数据包中子数据包在所述优先级指示字段对应的优先级下的顺序；在所述中间数据包中也加入索引字段和/或优先级指示字段，所述索引字段用于指示中间数据包中子数据包的顺序和/或指示中间数据包中子数据包在所述优先级指示字段对应的优先级下的顺序；

在所述恢复数据包和中间数据包中加入相应的指示信息，或者在发送恢复数据包和中间数据包时发送相应的指示信息，所述指示信息用于指示恢复数据包在各优先级下的子数据包，以及属于相应优先级的子数据包的数目或者所述子数据包占所述恢复数据包的比例；所述指示信息还用于指示中间数据包在各个优先级下的子数据包，以及属于相应优先级的子数据包的数目或者所述子数据包占所述中间数据包的比例；

向终端侧发送包含指示信息的恢复数据包、中间数据包，或者向终端侧发送恢复数据包、中间数据包以及相应的指示信息。

较佳的，所述解析源数据包中每个子数据包包头的重要性信息，包括：

读取每个子数据包包头中的优先 priority 字段，根据所述 priority 字段中的信息确定对应子数据包的优先级。

较佳的，所述根据所述重要性信息，将所述源数据包中每个子数据包分别映射到不同的预设优先级中，包括：

确定所述源数据包实际采用的优先级个数，以及各优先级下与所述 priority 字段的映射关系；

按照所述映射关系，将子数据包映射到相应的预设优先级中。

较佳的，所述将所述中间数据包转化为多个符号 symbol，并对转化成多个 symbol 的中间数据包进行前向纠错 FEC 编码后，得到包含多个 symbol 的恢复数据包，包括：

对所述中间数据包采用 RaptorQ 系统码的自适应 FEC 编码方式进行 FEC 编码，得到如下的中间码字：

$$\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_N \end{bmatrix}$$

式中： A_i 表示第 i 类源数据包对应的编码矩阵， $C_1, C_2, \dots, C_i, \dots, C_N$ 分别对应 N 类源数据包的中间码， $D_1, D_2, \dots, D_i, \dots, D_N$ 分别对应 N 类源数据包的数据，上标 -1 为逆矩阵运算符；

根据所述中间码字，得到恢复数据包，并在所述恢复数据包中附加 FEC payload ID，其中，所述恢复数据包中的恢复数据如下：

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_N \\ R_1 \\ R_2 \\ \dots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC1} & 0 & \dots & 0 \\ 0 & G_{ENC2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & G_{ENCN} \\ G_{ENC11} & 0 & \dots & 0 \\ G_{ENC12} & G_{ENC21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC1N} & G_{ENC2(N-1)} & \dots & G_{ENCN1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix}$$

式中： $R_1, R_2, \dots, R_i, \dots, R_N$ 分别对应 $1 \sim N$ 个源数据包的恢复数据， G_{ENCi} 表示第 i 个源数据包对应的 LT 编码矩阵， G_{ENC1i} 表示与 G_{ENC1} 具有相同生成规则的 LT 编码矩阵； G_{ENCi1} 表示与 G_{ENCi} 具有相同生成规则的 LT 编码矩阵； $i=1,2,\dots,N$ 。

根据本发明提供的一种基于媒体内容的自适应系统码 FEC 编译码方法，包括：

接收服务器发送的恢复数据包、中间数据包，或者接收恢复数据包、中间数据以及相应的指示信息；

解析出所述恢复数据包和中间数据包中包含的索引字段，并根据所述索引字段生成编码矩阵，其中，所述索引字段用于指示恢复数据包中子数据包的顺序，以及指示中间数据包在各个优先级下的子数据包；或者

通过恢复数据包和中间数据包的优先级指示字段直接获取索引信息，并根据所述索引信息生成编码矩阵，其中，所述索引信息用于指示恢复数据包中子数据包的顺序，以及指示中间数据包在各个优先级下的子数据包；

对所述编码矩阵进行前向纠错 FEC 译码，恢复出中间数据包；

解析所述中间数据包包头中的信息，重新排列所述中间数据包的子数据包的顺序，恢复出源数据包。

较佳的，所述解析出所述恢复数据包和中间数据包中包含的索引字段，包括：

通过以下公式得到源数据对应编码矩阵的相应行的索引字段：

$$\text{index} = \text{FEC payload ID} - \text{number of packets of all priorities before this packet}$$

式中：index 表示源数据对应编码矩阵的相应行索引，FEC payload ID 表示数据包包头的索引信息，number of packets of all priorities before this packet 表示该数据包之前的所有优先级的数据包总数。

较佳的，所述对所述编码矩阵进行前向纠错 FEC 译码，恢复出中间数据包，包括：

对所述编码矩阵进行初等行变换，并对接收到的相应恢复数据包中的符号 symbol 进行排序，根据恢复的数据包包头的索引信息 FEC_payload_ID 的值和对应优先级下符号 symbol 的数目确定该符号 symbol 所属的优先级和变换后的位置；或者

根据恢复数据包包头中的优先级字段和该优先级字段对应优先级下的索引信息，直接确定符号 symbol 所属的优先级和变换后的位置；

调整恢复数据包中的符号 symbol 的顺序，并通过高斯消元译码器和 RFC6330 译码器进行译码后，得到恢复的包含多个 symbol 的中间数据包；

将包含多个 symbol 的中间数据包转化为中间数据包。

根据本发明提供的一种基于媒体内容的自适应系统码 FEC 编译码装置，包括：

解析模块，用于解析源数据包中每个子数据包包头的重要性信息；

映射模块，用于根据所述重要性信息，将所述源数据包中每个子数据包分别映射到不同的预设优先级中；

排序模块，用于按照预设优先级重新排列子数据包，得到中间数据包；

编码模块，用于将所述中间数据包转化为多个符号 symbol，并对转化成多个 symbol 的中间数据包进行前向纠错 FEC 编码后，得到包含多个 symbol 的恢复数据包；

索引模块，用于在所述恢复数据包中加入索引字段和/或优先级指示字段，所述索引字段用于指示恢复数据包中子数据包的顺序和/或指示恢复数据包中子数据包在所述优先级指示字段对应的优先级下的顺序；在所述中间数据包中也加入索引字段和/或优先级指示字段，所述索引字段用于指示中间数据包中子数据包的顺序和/或指示中间数据包中子数据包在所述优先级指示字段对应的优先级下的顺序；；

指示信息生成模块，用于在所述恢复数据包和中间数据包中加入相应的指示信息，或者在发送恢复数据包和中间数据包时发送相应的指示信息；其中，所述指示信息用于指示恢复数据包在各优先级下的子数据包，以及属于相应优先级的子数据包的数目或者

所述子数据包占所述恢复数据包的比例；所述指示信息还用于指示中间数据包在各个优先级下的子数据包，以及属于相应优先级的子数据包的数目或者所述子数据包占所述中间数据包的比例；

发送模块，用于向终端侧发送所述恢复数据包、中间数据包和相应的信令信息，或者向终端侧发送恢复数据包、中间数据包以及相应的指示信息。

较佳的，所述编码模块，具体用于：

对所述中间数据包采用 RaptorQ 系统码的自适应 FEC 编码方式进行 FEC 编码，得到如下的中间码字：

$$\begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \vdots \\ 0 \\ D_N \end{bmatrix}$$

式中： A_i 表示第 i 类源数据包对应的编码矩阵， $C_1, C_2, \dots, C_i, \dots, C_N$ 分别对应 N 类源数据包的中间码， $D_1, D_2, \dots, D_i, \dots, D_N$ 分别对应 N 类源数据包的数据，上标 -1 为逆矩阵运算符；

根据所述中间码字，得到恢复数据包，并在所述恢复数据包中附加 FEC payload ID，其中，所述恢复数据包中的恢复数据如下：

$$\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_N \\ R_1 \\ R_2 \\ \vdots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC1} & 0 & \dots & 0 \\ 0 & G_{ENC2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & G_{ENCN} \\ G_{ENC11} & 0 & \dots & 0 \\ G_{ENC12} & G_{ENC21} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ G_{ENC1N} & G_{ENC2(N-1)} & \dots & G_{ENCN1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{bmatrix}$$

式中： $R_1, R_2, \dots, R_i, \dots, R_N$ 分别对应 $1 \sim N$ 个源数据包的恢复数据， G_{ENC_i} 表示第 i 个源数据包对应的 LT 编码矩阵， $G_{ENC_{1i}}$ 表示与 G_{ENC_1} 具有相同生成规则的 LT 编码矩阵； $G_{ENC_{i1}}$ 表示与 G_{ENC_i} 具有相同生成规则的 LT 编码矩阵； $i=1,2,\dots,N$ 。

根据本发明提供一种基于媒体内容的自适应系统码 FEC 编译码装置，包括：

接收模块，用于接收服务器发送的恢复数据包、中间数据包，或者接收恢复数据包、中间数据以及相应的指示信息；

编码矩阵生成模块，用于解析出所述恢复数据包和中间数据包中包含的索引字段，

并根据所述索引字段生成编码矩阵，其中，所述索引字段用于指示恢复数据包中子数据包的顺序，以及指示中间数据包在各个优先级下的子数据包；或者

通过恢复数据包和中间数据包的优先级指示字段直接获取索引信息，并根据所述索引信息生成编码矩阵，其中，所述索引信息用于指示恢复数据包中子数据包的顺序，以及指示中间数据包在各个优先级下的子数据包；，

译码模块，用于对所述编码矩阵进行前向纠错 FEC 译码，恢复出中间数据包；

恢复模块，用于解析所述中间数据包包头中的信息，重新排列所述中间数据包的子数据包的顺序，恢复出源数据包。

较佳的，所述编码矩阵生成模块，具体用于：

通过以下公式得到源数据对应编码矩阵的相应行的索引字段：

$$\text{index} = \text{FEC payload ID} - \\ \text{number of packets of all priorities before this packet}$$

式中：index 表示源数据对应编码矩阵的相应行索引，FEC payload ID 表示数据包包头的索引信息，number of packets of all priorities before this packet 表示该数据包之前的所有优先级的数据包总数。

较佳的，所述译码模块，具体用于：

对所述编码矩阵进行初等行变换，并对接收到的相应恢复数据包中的符号 symbol 进行排序，根据恢复的数据包包头的索引信息 FEC_payload_ID 的值和对应优先级下符号 symbol 的数目确定该符号 symbol 所属的优先级和变换后的位置；或者

根据恢复数据包包头中的优先级字段和该优先级字段对应优先级下的索引信息，直接确定符号 symbol 所属的优先级和变换后的位置；

调整恢复数据包中的符号 symbol 的顺序，并通过高斯消元译码器和 RFC6330 译码器进行译码后，得到恢复的包含多个 symbol 的中间数据包；

将包含多个 symbol 的中间数据包转化为中间数据包。

根据本发明提供一种基于媒体内容的自适应系统码 FEC 编译码系统，包括服务器和至少一个终端，其中，所述服务器用于执行权利要求 7-10 中任一项所述的基于媒体内容的自适应系统码 FEC 编译码方法，所述终端用于执行权利要求 11-13 任一项所述的基于媒体内容的自适应系统码 FEC 编译码方法。

根据本发明提供一种存储有计算机程序的计算机可读存储介质，所述计算机程序被处理器执行时实现权利要求 7 至 13 中任一项所述的基于图像的条纹布料检测定位裁剪方法的步骤。

与现有技术相比，本发明具有如下的有益效果：

本发明提供的基于媒体内容的自适应系统码 FEC 方法，通过根据媒体内容的重要性按照优先级排序，采用不等差错保护（UEP），最大限度保证媒体内容质量的同时，减少 FEC 造成的数据冗余；使系统码的编译码更加灵活，不需要对源数据流进行分流，因此降低了发送端 FEC 编码的复杂度，提高了 FEC 编码的效率；同时，可以根据当前网络状态的变化动态的调整编码方案，因此对时变网络具有更强的适应性。

附图说明

通过阅读参照以下附图对非限制性实施例所作的详细描述，本发明的其它特征、目的和优点将会变得更明显：

图 1 为针对媒体资源数据的 FEC 两层结构示意图；

图 2、图 3 为图像组中各帧依赖关系示意图；

图 4 为喷泉码的不等差错保护编码方案示意图；

图 5 为喷泉码不等差错保护的系统架构示意图；

图 6 为系统 RaptorQ 码编码矩阵结构示意图；

图 7 为系统 RaptorQ 码不等差错保护的编码矩阵结构示意图；

图 8 为 LT 编码矩阵不等差错保护结构示意图；

图 9 为多优先级不等差错保护系统喷泉码编码流程图；

图 10 为多优先级不等差错保护系统喷泉码译码流程图；

图 11 为基于媒体内容的自适应系统码 FEC 系统框架示意图；

图 12 为各优先级的映射关系示意图；

图 13 为按照优先级划分对数据包流重新排序示意图；

图 14 为 FEC 编码后对源数据包和恢复数据包的处理流程示意图；

图 15 为接收端解析信令信息和数据包信息示意图；

图 16 为根据接收到的数据包生成编码矩阵示意图；

图 17 为对接收到的数据包进行处理的流程示意图；

图 18 为恢复数据包流的初始顺序的结果示意图；

图 19 为接收端解析数据包包头信息示意图。

具体实施方式

下面结合具体实施例对本发明进行详细说明。以下实施例将有助于本领域的技术人员进一步理解本发明，但不以任何形式限制本发明。应当指出的是，对本领域的普通技术人员来说，在不脱离本发明构思的前提下，还可以做出若干变化和改进。这些都属于本发明的保护范围。

以下，对本申请中的部分用语进行解释说明，以便于本领域技术人员理解。

1) 不等差错保护 (Unequal Error Protection, UEP) 是联合信源信道编码的一种。其核心思想是，依据码流的各部分数据的重要性不同，对各部分数据采用不同的信道保护机制，即对重要码流进行重点保护。尽管 UEP 降低了非重要码流的抗噪声性能，但有利于系统抗误码总体性能的提升。

2) 数字喷泉码 (Digital Fountain Code) 在传输过程中，不需要反馈及自动重发机制，避免了信号往返的延时以及广播应用中的反馈爆炸问题。数字喷泉的基本思想是：发端将原始数据分割成 k 个数据符号，对这些数据符号进行编码，输出一个任意长度的编码符号流，接收端只需正确地接收 n (n 稍大于 k) 个编码符号就可以很大的概率恢复出所有的 k 个数据符号。

数字喷泉码本身就具有 UEP 性能，可以实现对不同重要性的数据的保护。相比与传统的固定码率的信道编码方法，数字喷泉码具有以下明显的优势：

a、理想的可扩展性，由于单向广播没有反馈，发送方不受用户数量增长的任何影响。使得发送方能够为任意数量的用户提供服务。

b、适应时变信道，高效利用信道容量。用户的译码性能与信道的删除概率和带宽无关。当信道丢包率较高、状况不好时，不会对接收端的译码造成影响，即接收端接收足够数量的编码数据就可以正常译码，具有更强的适应性。

c、编译码复杂度低。在理想情况下，喷泉码生成每个编码符号具有线性编译码复杂度，有利于简化收发端编译码器的设计和软件化实现。

d、对异质用户的适用性能良好。喷泉码的无码率特性使得具有不同丢包率或带宽的用户之间互不影响，优质用户不受劣质用户牵制。除此之外，数字喷泉码可以支持中断续传、异步接入等多种服务模式。

如图 1 所示：针对媒体资源数据的 FEC 两层结构，第一层将源数据包块 source packet block 分为较多的小块分别做 FEC 保护，第二层是一个整块做 FEC 保护。第一层划分较细致可以提供较小的时延，第二层保证了恢复性能和较小的冗余，但是这种方式需要进

行两次 FEC 编码，降低了编码效率，同时缺乏灵活性。

在一个数据流中可以根据其内容分成不同类型的数据包，不同类型的数据包具有不同的重要程度。例如，数据包中的 I、B、P 帧的重要程度如图 2 图 3 所示。一个图像组中各帧依赖关系，说明一个图像组中不同帧的依赖程度和重要性不同，I 帧是最重要的，B、P 帧的重要程度较低。因此可以根据数据内容重要性进行不等差错保护。具体考虑传输信源数据具有两个重要等级的情况，将信源数据按上述标准进行划分，并将其按照重要程度重新进行排序，通过改变系统喷泉码编码矩阵的结构，达到对不同内容的不等差错保护。其系统的基本架构如图 5 所示。经过系统喷泉码的不等差错保护编码后产生的数据如图 4 所示。

实施例 1

本实施例以 RaptorQ 系统码为例，系统 RaptorQ 码的编码矩阵的结构如图 6 所示。其中，D 表示源数据，C 表示中间码，G_LDPC 表示 LDPC 矩阵，I_S、I_H 表示单位矩阵，G_ENC 表示 LT 编码矩阵，则编码矩阵可以如下表示。

$$G_p = \begin{bmatrix} G_{LDPC,1} & I_S & G_{LDPC,2} \\ G_{HDPC} & & I_H \end{bmatrix} \quad (1)$$

$$A = \begin{bmatrix} G_p \\ G_{ENC} \end{bmatrix} \quad (2)$$

在进行 FEC 编码时，由源数据和图 6 所示的编码矩阵产生中间码。

$$C = A^{-1} \times \begin{bmatrix} 0 \\ D \end{bmatrix} \quad (3)$$

再根据编码矩阵中的 LT 矩阵和得到的中间码产生源数据和恢复数据。其中 R 表示编码后得到的恢复数据，G_ENC' 是与 G_ENC 相同生成方式产生的 LT 矩阵。

$$\begin{bmatrix} D \\ R \end{bmatrix} = \begin{bmatrix} G_{ENC} \\ G_{ENC'} \end{bmatrix} \times C \quad (4)$$

采用上述方法对源数据进行编码，不能对源数据进行优先级划分并采用不等差错保护编码，因此需要重新设计编码矩阵 A 以及 LT 矩阵 G_ENC 的矩阵结构。以源数据中具有两种不同优先级的数据为例，设优先级分别为优先级 1 和 2，将源数据中的不同优先级的数据区分开，对其分配不同的冗余，已达到不等差错保护的效果。

编码过程：

其编码矩阵结构如图 7 所示。其编码矩阵如公式(6)所示，其中 G_ENC₁ 与 G_ENC₂ 是由不同规则产生的编码矩阵，为了保证对重要数据的保护可以提高 G_ENC₁ 矩阵的度分布。

$$A_i = \begin{bmatrix} G_{p_i} \\ G_{ENC_i} \end{bmatrix} \quad i = 1, 2 \quad (5)$$

$$A = \begin{bmatrix} G_{p_1} & 0 \\ G_{ENC_1} & 0 \\ 0 & G_{p_2} \\ 0 & G_{ENC_2} \end{bmatrix} \quad (6)$$

这样，针对不同重要程度的数据可以产生相应的中间码。

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \end{bmatrix} \quad (7)$$

为了达到对不同重要程度数据的不等差错保护，还需要改变 LT 编码矩阵的结构如图 8 所示，增加对重要数据保护的冗余数据。其中 G_{ENC_1}' 是与 G_{ENC_1} 具有生成方式相同， G_{ENC_2}' 与 G_{ENC_2} 的生成方式相同。得到的恢复数据 R_1 只与优先级为 1 的数据有关，而恢复数据 R 与优先级 1 和 2 的数据均有关。由此增加了重要数据的冗余，提高了其保护强度。其中若由于信道条件的限制，数据总的冗余度有限，生成 R_1 和 R_2 的比例由 $[G_{ENC_1}' \ 0]$ 与 $[G_{ENC_1}'' \ G_{ENC_2}']$ 的行数比确定，可以增加 $[G_{ENC_1}' \ 0]$ 的行数进一步提高优先级为 1 的数据的保护强度。

$$\begin{bmatrix} D_1 \\ D_2 \\ R_1 \\ R_2 \end{bmatrix} = \begin{bmatrix} G_{ENC_1} & 0 \\ 0 & G_{ENC_2} \\ G_{ENC_1}' & 0 \\ G_{ENC_1}'' & G_{ENC_2}' \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad (8)$$

译码过程：

由于在经过有损信道后，部分数据包发生了丢失，根据接收到的符号 ISI，生成恢复矩阵，需要擦除掉丢失符号 ISI 对应的矩阵行，例如，对于矩阵 $REV_G_ENC_1$ ，根据接收到的 REV_D_1 和发送端的 D_1 可以得到丢失数据的 ISI，从而删除 G_{ENC_1} 相应的行。与编码过程类似，进行第一阶段解码时，根据恢复矩阵以及接收到的数据包得到中间码字。

$$\begin{bmatrix} G_{p_1} & 0 \\ REV_G_ENC_1 & 0 \\ 0 & G_{p_2} \\ 0 & REV_G_ENC_2 \\ REV_G_ENC_1' & 0 \\ [REV_G_ENC_1'' \ REV_G_ENC_2'] \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ 0 \\ REV_D_2 \\ REV_R_1 \\ REV_R_2 \end{bmatrix} \quad (9)$$

对上面矩阵方程进行初等行变换，得到以下形式：

$$\begin{bmatrix} G_{p1} & 0 \\ REV_G_ENC_1 & 0 \\ REV_G_ENC_1' & 0 \\ 0 & G_{p2} \\ 0 & REV_G_ENC_2 \\ _REV_G_ENC_1'' & REV_G_ENC_2' \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix} \quad (10)$$

令 $REV_A = \begin{bmatrix} G_{p1} & 0 \\ REV_G_ENC_1 & 0 \\ REV_G_ENC_1' & 0 \\ 0 & G_{p2} \\ 0 & REV_G_ENC_2 \\ _REV_G_ENC_1'' & REV_G_ENC_2' \end{bmatrix}$, 根据接收到的各优先级下数据包的数量以及接收到的数据包总数量分多种情况进行解码。

接收到的编码符号总数小于源数据编码符号的个数, 并且 $REV_A_1 = \begin{bmatrix} G_{p1} \\ REV_G_ENC_1 \\ REV_G_ENC_1' \end{bmatrix}$

的行数小于列数; C_1 、 C_2 均不可求出。

接收到的编码符号总数小于源数据编码符号的个数, 但是

$REV_A_1 = \begin{bmatrix} G_{p1} \\ REV_G_ENC_1 \\ REV_G_ENC_1' \end{bmatrix}$ 的行数大于等于列数;

若 $REV_A_1 = \begin{bmatrix} G_{p1} \\ REV_G_ENC_1 \\ REV_G_ENC_1' \end{bmatrix}$ 的行数大于等于列数, 则 REV_A_1 可逆, 其逆矩阵记为

$REV_A_1^{-1}$; 则可得:

$$[C_1] = REV_A^{-1} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix} \quad (11)$$

但 C_2 不能求出。

接收到的编码符号总数大于源数据编码符号的个数, $REV_A_1 = \begin{bmatrix} G_{p1} \\ REV_G_ENC_1 \\ REV_G_ENC_1' \end{bmatrix}$ 的

行数大于等于列数;

若 $REV_A_1 = \begin{bmatrix} G_{p1} \\ REV_G_ENC_1 \\ REV_G_ENC_1' \end{bmatrix}$ 的行数大于等于列数, 则 REV_A_1 可逆, 其逆矩阵记为

$REV_A_1^{-1}$; 若 $REV_A_2 = \begin{bmatrix} G_{p2} \\ REV_G_ENC_2 \\ REV_G_ENC_2' \end{bmatrix}$ 的行数大于等于列数, 则 REV_A_2 可逆, 其

逆矩阵记为 $REV_A_2^{-1}$ 。若 REV_A_1 和 REV_A_2 都可逆，则 REV_A 可逆，其逆矩阵记为

$$REV_A^{-1}, \quad \text{记} \quad B_{1,1} = \begin{bmatrix} 0 \\ 0 \\ REV_G_ENC_1'' \end{bmatrix}, \quad \text{易知}$$

$$REV_A^{-1} = \begin{bmatrix} REV_A_1^{-1} & 0 \\ REV_A_2^{-1} B_{1,1} REV_A_1^{-1} & REV_A_2^{-1} \end{bmatrix}。 \text{则可得:}$$

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = REV_A^{-1} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix} = \begin{bmatrix} REV_A_1^{-1} & 0 \\ REV_A_2^{-1} B_{1,1} REV_A_1^{-1} & REV_A_2^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix} \quad (12)$$

但若 $REV_A_2 = \begin{bmatrix} G_p_2 \\ REV_G_ENC_2' \\ REV_G_ENC_2 \end{bmatrix}$ 的行数小于列数，则 REV_A_2 不存在逆矩阵，只能求

$$\text{出} [C_1] = REV_A^{-1} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix}。$$

接收到的编码符号总数大于等于源数据编码符号的个数，

$$REV_A_1 = \begin{bmatrix} G_p_1 \\ REV_G_ENC_1' \\ REV_G_ENC_1 \end{bmatrix} \text{的行数小于列数，但} REV_A_2 = \begin{bmatrix} G_p_2 \\ REV_G_ENC_2' \\ REV_G_ENC_2 \end{bmatrix} \text{的行数大于等}$$

于列数；

使用高斯消元法直接对矩阵 REV_A 进行求解，即解下列线性方程组：

$$\begin{bmatrix} G_p_1 & 0 \\ REV_G_ENC_1 & 0 \\ REV_G_ENC_1' & 0 \\ 0 & G_p_2 \\ 0 & REV_G_ENC_2' \\ REV_G_ENC_1'' & REV_G_ENC_2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix} \quad (13)$$

易求得 C_1 、 C_2 。

由上述解码过程的第一阶段可以得到，任何情况下都首先保证 C_1 的正确解出，由于中间码字 C_1 和 C_2 作为在解码第二阶段中恢复出源数据的重要数据，提高 C_1 的解码成功率可以进一步提高源数据正确恢复的概率。

根据得到的中间码字，生成系统码编码符号

$$\begin{bmatrix} D_1 \\ D_2 \end{bmatrix} = \begin{bmatrix} G_ENC_1 & 0 \\ 0 & G_ENC_2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad (14)$$

若第一阶段没有求出 C_1 、 C_2 ，则译码失败，无法译出源符号；若只求出 C_1 ，则可以

保证优先的源码数据 D_1 可以正确译出；若 C_1 、 C_2 都求出，则利用上式可以得出最终的源码 D_1 和 D_2 。

同时，该方案不仅适用于两个优先级的场景，可扩展到多个优先级的源数据上，其具体过程如下。

编码过程：

首先根据源数据以及源数据的优先级，生成中间码字。假设源数据划分成 l 个数据包，得到：

$$\begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ G_{ENC_1} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & G_{ENC_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_l} \\ 0 & 0 & \dots & G_{ENC_l} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_l \end{bmatrix} \quad (15)$$

由式（16）可得到中间码字 $C_1, C_2, \dots, C_l$ 。

$$\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_l^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_l \end{bmatrix} \quad (16)$$

根据得到的中间码字生成源数据和恢复数据。

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_l \\ R_1 \\ R_2 \\ \dots \\ R_l \end{bmatrix} = \begin{bmatrix} G_{ENC_1} & 0 & \dots & 0 \\ 0 & G_{ENC_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{ENC_l} \\ G_{ENC_{11}} & 0 & \dots & 0 \\ G_{ENC_{12}} & G_{ENC_{21}} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC_{1l}} & G_{ENC_{2(l-1)}} & \dots & G_{ENC_{l1}} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} \quad (17)$$

整个编码流程如图 9 所示。将源数据和恢复数据由发送端发送给接收端，由于信道的衰减，部分数据可能会发生丢失，接收端接收到数据后，根据接收到的数据情况进行译码恢复出源数据。

译码过程：

由于部分数据发生丢失，需要根据接收到的符号 ISI，生成恢复矩阵，相当于擦除掉丢失符号 ISI 对应的矩阵行，形成相应的矩阵，记为 REV_G_ENC 。根据接收数据的数目不同，分不同情况生成中间码字。

$$\begin{bmatrix}
G_{p_1} & 0 & \dots & 0 \\
REV_G_ENC_1 & 0 & \dots & 0 \\
0 & G_{p_2} & \dots & 0 \\
0 & REV_G_ENC_2 & \dots & 0 \\
\dots & \dots & \dots & \dots \\
0 & 0 & \dots & G_{p_l} \\
0 & 0 & \dots & REV_G_ENC_l \\
REV_G_ENC_{11} & 0 & \dots & 0 \\
REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\
\dots & \dots & \dots & \dots \\
REV_G_ENC_{1l} & REV_G_ENC_{2(l-1)} & \dots & REV_G_ENC_{l1}
\end{bmatrix}
\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ 0 \\ REV_D_2 \\ \dots \\ 0 \\ REV_D_l \\ REV_R_1 \\ REV_R_2 \\ \dots \\ REV_R_l \end{bmatrix} \quad (18)$$

对上面矩阵方程进行初等行变换，得到以下形式：

$$\begin{bmatrix}
G_{p_1} & 0 & \dots & 0 \\
REV_G_ENC_1 & 0 & \dots & 0 \\
REV_G_ENC_{11} & 0 & \dots & 0 \\
0 & G_{p_2} & \dots & 0 \\
0 & REV_G_ENC_2 & \dots & \dots \\
REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\
\dots & \dots & \dots & \dots \\
0 & 0 & \dots & G_{p_l} \\
0 & 0 & \dots & REV_G_ENC_l \\
REV_G_ENC_{1l} & REV_G_ENC_{2(l-1)} & \dots & REV_G_ENC_{l1}
\end{bmatrix}
\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \\ \dots \\ 0 \\ REV_D_l \\ REV_R_l \end{bmatrix} \quad (19)$$

$$\text{设 } REV_A = \begin{bmatrix}
G_{p_1} & 0 & \dots & 0 \\
REV_G_ENC_1 & 0 & \dots & 0 \\
REV_G_ENC_{11} & 0 & \dots & 0 \\
0 & G_{p_2} & \dots & 0 \\
0 & REV_G_ENC_2 & \dots & \dots \\
REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\
\dots & \dots & \dots & \dots \\
0 & 0 & \dots & G_{p_l} \\
0 & 0 & \dots & REV_G_ENC_l \\
REV_G_ENC_{1l} & REV_G_ENC_{2(l-1)} & \dots & REV_G_ENC_{l1}
\end{bmatrix}, \quad \text{设 } R(i) (1 \leq$$

$i \leq l$) 为接收到并经扩展后第 i 级符号数， $L(i)$ 为编码端每级符号长度，对于 $i = 1:l$ ，计算每级生成矩阵的秩 $\text{eachRank}(i) = \min(\text{Len}(R(i)), \text{sum}(L(1:i)))$ ；再计算累积矩阵的秩 $\text{cumRank}(i) = \min(\text{cumRank}(i-1) + \text{eachRank}(i), \text{sum}(L(1:i)))$ ，其中 $\text{cumRank}(1) = \text{eachRank}(1)$ 。依次判断 $\text{cumRank}(i) \geq \text{sum}(L(1:i))$ ：若成立， $\text{cumFullRank}(i) = 1$ ；否则 $\text{cumFullRank}(i) = 0$ 。以下根据接收到的每级符号数分情况分别求中间码 $C_1, C_2 \dots C_l$ ，设置 i 初始值为 1：

若满足 $\text{cumFullRank}(i) == 1 \&\& \text{cumFullRank}(1:i) > 0$ ，使用 RFC 6330 译码方法解出 $C(1) \sim C(i)$

$$\text{设 } REV_A_{\Delta i} = \begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & REV_G_ENC_2 \dots & \dots & \dots \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_i} \\ 0 & 0 & \dots & REV_G_ENC_i \\ REV_G_ENC_{1i} & REV_G_ENC_{2(i-1)} & \dots & REV_G_ENC_{i1} \end{bmatrix}, \text{ 并且易知}$$

$$REV_A_{\Delta i}^{-1} = \begin{bmatrix} A_1^{-1} & 0 & 0 & 0 \\ X_{1,2} & A_2^{-1} & 0 & 0 \\ \dots & \dots & \dots & 0 \\ X_{1,i} & X_{2,(i-1)} & \dots & A_{i,1}^{-1} \end{bmatrix}, \text{ 其中 } X_{1,2} = A_2^{-1}B_{1,2}A_1^{-1}, X_{a,b} =$$

$A_b^{-1}B_{a,b}A_a^{-1} + \sum_{k=a+1}^{b-1} A_b^{-1}B_{k,b}X_{a,k}$, 进而中间码可以通过下式求得:

$$C_{i_temp} = REV_A_i^{-1} \begin{bmatrix} 0 \\ REV_D_i \\ REV_R_i \end{bmatrix},$$

$$C_1 = C_{1_temp}$$

$$C_2 = C_{2_temp} + X_{1,2} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix} = C_{2_temp} + A_2^{-1}B_{1,2}C_1$$

$$C_3 = C_{3_temp} + X_{1,3} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix} + X_{2,3} \begin{bmatrix} 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix} = C_{2_temp} + A_3^{-1}B_{1,3}C_1 + A_3^{-1}B_{2,3}C_2$$

... (数学归纳法)

$$C_i = C_{i_temp} + \sum_{j=1}^{i-1} X_{j,i} \begin{bmatrix} 0 \\ REV_D_j \\ REV_R_j \end{bmatrix} = C_{i_temp} + \sum_{j=1}^{i-1} A_i^{-1}B_{j,i}C_j$$

求得 $C_1, C_2 \dots C_i$, 而其他 $C_{i+1} \dots C_l$ 不能求得。

若满足 $\text{cumFullRank}(i) == 1$, 但不满足 $\text{cumFullRank}(1:i) > 0$, 使用高斯消元译码方法解出 $C(1) \sim C(i)$

$$\text{设 } REV_A_{\Delta i} = \begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & REV_G_ENC_2 \dots & \dots & \dots \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_i} \\ 0 & 0 & \dots & REV_G_ENC_i \\ REV_G_ENC_{1i} & REV_G_ENC_{2(i-1)} & \dots & REV_G_ENC_{i1} \end{bmatrix}, \text{ 可以通过高}$$

斯消元法解线性方程组:

$$\begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & REV_G_ENC_2 \dots & \dots & \dots \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_i} \\ 0 & 0 & \dots & REV_G_ENC_i \\ REV_G_ENC_{1i} & REV_G_ENC_{2(i-1)} & \dots & REV_G_ENC_{i1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_i \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \\ \dots \\ 0 \\ REV_D_i \\ REV_i \end{bmatrix} \quad (20)$$

求得 C_1 、 $C_2 \dots C_i$ ，而其他 $C_{i+1} \dots C_l$ 不能求得。

对于某 i 值，若不满足 $\text{cumFullRand}(i) == 1$ 则相应 C_i 不能解出。直到 $i == 1$ ，上述两个条件都不能满足，则 C_1 、 $C_2 \dots C_l$ 全部不能解出，译码失败。

令 $i=i-1$ 继续循环，即 $i=l-1:1$ ，重复以上过程。

整个译码流程如图 10 所示。

本实施例中的方法，能够有效地节省资源，如果在接收端，人为的根据情况舍弃已经接收的数据（例如 B 帧）则造成了传输资源的浪费，上述方案从源端解决问题，让不想要的包在传输过程中更大概率丢掉，而更大程度的保护了重要的包。

本实施例中的方法，提供个性化的传输方案。可以根据信道状况、用户体验等设计不等差错保护的方案，使得视频传输更加灵活、细致。同时，尽管发送端的编码方式灵活，接收端都可以根据列表信息正确的恢复出原始的数据。

本实施例中的方法，提供更加灵活的编码矩阵的设计。可以根据实际媒体的应用场景，基于媒体内容进行数据的优先级划分，可以根据实际需求更加灵活的设计 FEC 编码矩阵，达到不等差错保护的效果。

本实施例中的方法，提供更加灵活的译码方式。根据接收到的各优先级数据的数目，可以采用不同的译码算法，在有限的带宽资源下，可以最大程度的恢复优先级高的数据，

提高对对优先级较高的数据的保护强度。

实施例 2

可以对源数据包中每个子数据包分别映射到不同的预设优先级，或者将实施例 1 与实施例 2 结合。具体如下：

图 11 为基于媒体内容的自适应系统码 FEC 系统框架示意图，包括服务器和至少一个终端，首先服务器侧根据实际应用场景和媒体内容划分出多个优先级，解析数据包包头中关于该数据包的重要性信息，将每个数据包映射到不同的优先级下。其次，根据优先级情况和映射关系，将所有需要进行前向纠错编码的数据包（FEC）按照优先级从高到低重新排列。对重新排序后的源数据，根据实际的信道情况，采用不等差错保护的 FEC 编码方式进行编码，生成恢复数据（repair symbols）。并在并在 FEC 编码结束后，在源数据包和恢复数据包中加入指示当前数据包顺序的索引字段。统计各优先级下的源数据和恢复数据，生成相应的信令信息，在信令中指示出各优先级下的数据包数目或所占比例，以及各优先级下对应的恢复数据的数目或所占比例，以便接收端在接收到数据后可以正确的恢复数据。最后，将源数据包、恢复数据包以及信令发送到终端（客户端或者是接收端）。

相应的，终端侧首先解析信令信息以及接收到的数据包中的索引字段，得到每个优先级下接收到的源数据包和恢复数据包的数目以及相应的数据包索引。其次，根据得到的数据包索引信息，在接收端生成编码矩阵。最后，根据生成的编码矩阵，分不同的情况进行 FEC 译码，并恢复出源数据包。由于恢复出的源数据包的顺序是重新排序后的顺序，通过解析数据包包头中源数据包顺序，重新排列源数据包，得到源数据包的初始数据。

下面以具体的实施例对本发明的技术方案进行详细说明。下面这几个具体的实施例可以相互结合，对于相同或相似的概念或过程可能在某些实施例不再赘述。

实施例 2.1

如图 12 所示，可以包括如下步骤：

S1：解析编码封装后的数据包的包头，读取 DU header 中的 priority 字段，根据实际应用确定实际采用的优先级个数以及各优先级下与 priority 字段的映射关系。

S2：根据划分的优先级以及映射关系，对原始的源数据流按照优先级由高到低进行重新排序，并统计各优先级下数据包的数目，如图 13 所示。

S3：对重新排序后的源数据流采用系统 RaptorQ 系统码的自适应 FEC 编码方案进行

FEC 编码，生成恢复数据。

具体地，对所述中间数据包采用 RaptorQ 系统码的自适应 FEC 编码方式进行 FEC 编码，得到如下的中间码字：

$$\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_N \end{bmatrix}$$

式中： A_i 表示第 i 类源数据包对应的编码矩阵， $C_1, C_2, \dots, C_i, \dots, C_N$ 分别对应 N 类源数据包的中间码， $D_1, D_2, \dots, D_i, \dots, D_N$ 分别对应 N 类源数据包的数据，上标 -1 为逆矩阵运算符；

根据所述中间码字，得到恢复数据包，并在所述恢复数据包中附加 FEC payload ID，其中，所述恢复数据包中的恢复数据如下：

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_N \\ R_1 \\ R_2 \\ \dots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC1} & 0 & \dots & 0 \\ 0 & G_{ENC2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & G_{ENCN} \\ G_{ENC11} & 0 & \dots & 0 \\ G_{ENC12} & G_{ENC21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC1N} & G_{ENC2(N-1)} & \dots & G_{ENCN1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix}$$

式中： $R_1, R_2, \dots, R_i, \dots, R_N$ 分别对应 $1 \sim N$ 个源数据包的恢复数据， G_{ENCi} 表示第 i 个源数据包对应的 LT 编码矩阵， G_{ENC1i} 表示与 G_{ENC1} 具有相同生成规则的 LT 编码矩阵； G_{ENCi1} 表示与 G_{ENCi} 具有相同生成规则的 LT 编码矩阵； $i=1,2,\dots,N$ 。

并根据从 FEC 编码器输出的数据流（重新排序后的数据流），生成源数据包和恢复数据包，加上 FEC payload ID，如图 14 所示。

S4：生成 AL-FEC message，在信令中指示出各优先级下的数据包数目，以及各优先级下对应的恢复数据包数目，以便接收端在接收到数据后可以正确的恢复数据。具体信令如表 1 所示。

表 1 信令信息对照

语法	值	比特位数	备注
if (fec_coding_structure == 0100) {			
number_of__priority	N8		uimbsf
for(i=0;i<N8;i++) {			
fec_ priority_ value		8	uimbsf
number_of_source_symbol		16	uimbsf
number_of_repair_symbol		16	uimbsf
}			
if (private_fec_flag == 1) {			
private_flag	N7	1	bslbf
private_field_length		7	bslbf
private_field		N7*8	uimbsf
}			
repair_flow_id		8	uimbsf
fec_code_id_for_repair_flow		8	uimbsf
maximum_k_for_repair_flow		24	uimbsf
maximum_p_for_repair_flow		24	uimbsf
protection_window_time		32	uimbsf
protection_window_size		32	uimbsf
}			

表 1 中，number_of_priority：指示优先级的个数。fec_priority_value：8 位，指示各优先级对应的值，该值最多可与 DU header 中 priority 所定义的优先级相同，也可以对应多个 DU header 中的 priority。number_of_source_symbol：16 位，指示在优先级为 fec_priority_value 所指示的值下，源符号的数目。number_of_repair_symbol：16 位，指示在优先级为 fec_priority_value 所指示的值下，恢复符号的数目。private_fec_flag：指

示位, 指明是否使用私有的 FEC 编码方案。private_flag: 指示位, 指明是否存在一个 private_field 用来描述所使用的私有 FEC 编码方案。private_field_length: 长度字段, 用于描述私有 FEC 编码方案的字段的长度。private_field: 用于描述私有 FEC 方案的详细信息。priority_id: 优先级 id, 用于指示 MMT 包的优先级。fec_code_id_for_repair_flow: 用于描述所使用的 FEC 编码方案。repair_flow_id: 8 位整数, 用于指示生成的 FEC repair flow, 与 FEC repair 包的包头中的 packet id 有对应关系。maximum_k_for_repair_flow: 24 位整数, 描述在一个 source symbol 块中 source symbol 的最大数目。maximum_p_for_repair_flow: 24 位整数, 描述在一个 repair symbol 块中 repair symbol 的最大数目。protection_window_time: 保护窗时间, 指示在 FEC 编码中发送第一个 source 或者 repair 包与发送最后一个 source 或者 repair 包之间的最大时间差, 单位为毫秒。protection_window_size: 保护窗值, 指示在 FEC 编码流中发送第一个 FEC 包的负载与发送最后一个 FEC 包负载之间的最大计数值。

如图 15 所示, 根据 FEC 各优先级下的数据包数目, 通过以下公式得到源数据对应编码矩阵的相应行的索引:

$$index = \text{FEC payload ID} - \text{number of packets of all priorities before this packet}.$$

式中: index 表示源数据对应编码矩阵的相应行索引, FEC payload ID 表示数据包包头的索引信息, number of packets of all priorities before this packet: 表示该数据包之前的所有优先级的数据包总数。

现以 4 个优先级, 每个优先级的数目为 100 为例说明, 如图 16 所示。source_FEC_payload_ID 是根据重新排序后的源数据包的顺序排序的, 优先级为 1 的可以直接根据当前的 source_FEC_payload_ID 确定生成编码矩阵的 ISI (索引)。当优先级大于 1 时, 在解析包头时, 根据其 source_FEC_payload_ID 的值和各优先级下源符号的数目, 可以确定其属于哪一优先级, 例如 source_FEC_payload_ID = 108, 可以确定属于优先级 2, 有上述的公式可得其 ISI = 8。对于恢复数据包, 其可以根据包头中的 repair_FEC_payload_ID 确定其在生成矩阵的 ISI。由于发送端与接收端采用相同的 FEC 编码机制, 根据上述两个索引可以生成相应的编码矩阵。

根据生成的 FEC 编码矩阵, 对编码矩阵进行初等行变换, 并对相应的接收到的数据包进行排序, 根据 repair_FEC_payload_ID 的值和各优先级下恢复符号的数目, 确定其属于哪一优先级, 从而调整恢复符号在符号块中的位置, 实现相同的变化, 如图 17 所示。根据接收到的不同优先级的数据包的数目不同, 分多种情况进行解码, 其流程图

如图 10 所示。最终得到恢复后的源数据，再根据包头中的 packet_sequence_number，将得到的源数据包恢复为初始的顺序，具体如图 18 所示。

实施例 2.2：如图 12 所示，可以包括如下步骤：

S1：解析编码封装后的数据包的包头，读取 DU header 中的 priority 字段，根据实际应用确定实际采用的优先级个数以及各优先级下与 priority 字段的映射关系。

S2：根据划分的优先级以及映射关系，对原始的源数据流按照优先级由高到低进行重新排序，并统计各优先级下数据包的数目，如图 13 所示。

S3：对重新排序后的源数据流采用系统 RaptorQ 系统码的自适应 FEC 编码方案进行 FEC 编码，生成恢复数据。

具体地，对所述中间数据包采用 RaptorQ 系统码的自适应 FEC 编码方式进行 FEC 编码，得到如下的中间码字：

$$\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_N \end{bmatrix}$$

式中： A_i 表示第 i 类源数据包对应的编码矩阵， $C_1, C_2, \dots, C_i, \dots, C_N$ 分别对应 N 类源数据包的中间码， $D_1, D_2, \dots, D_i, \dots, D_N$ 分别对应 N 类源数据包的数据，上标 -1 为逆矩阵运算符；

根据所述中间码字，得到恢复数据包，并在所述恢复数据包中附加 FEC payload ID，其中，所述恢复数据包中的恢复数据如下：

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_N \\ R_1 \\ R_2 \\ \dots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC1} & 0 & \dots & 0 \\ 0 & G_{ENC2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & G_{ENCN} \\ G_{ENC11} & 0 & \dots & 0 \\ G_{ENC12} & G_{ENC21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC1N} & G_{ENC2(N-1)} & \dots & G_{ENCN1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix}$$

式中： $R_1, R_2, \dots, R_i, \dots, R_N$ 分别对应 $1 \sim N$ 个源数据包的恢复数据， G_{ENCi} 表示第 i 个源数据包对应的 LT 编码矩阵， G_{ENC1i} 表示与 G_{ENC1} 具有相同生成规则的 LT 编码矩阵； G_{ENCi1} 表示与 G_{ENCi} 具有相同生成规则的 LT 编码矩阵； $i=1,2,\dots,N$ 。

S4：在数据包包头扩展中生成相应的指示信息，指示出各优先级下的数据包数目，

以及各优先级下对应的恢复数据包数目,以便接收端在接收到数据后可以正确的恢复数据。具体如表 2 所示。

表 2

										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
+-----																																							

表 2 中，**type**: 16 位，指示当前包头扩展的类型，此处为指示用于自适应前向纠错编码的中间数据包或恢复数据包。**length**: 16 位，用于指示 **header_extension** 的长度。**FEC_priority**: 8 位，用于指示当前数据包所属的优先级。**sub_encoded_symbol_id**: 16 位，用于指示该优先级下中间数据包或恢复数据包数目的索引。**N**: 8 位，用于指示优先级的总个数。**SSBL[i]**: 24 位，**i** 表示第 **i** 个优先级，用于指示各优先级的源数据包数目。**RSBL[i]**: 24 位，**i** 表示第 **i** 个优先级，用于指示各优先级的恢复数据包数目。

如图 19 所示, 根据 FEC 各优先级下的数据包数目, 通过以下公式得到源数据对应编码矩阵的相应行的索引:

index =
FEC payload ID – number of packets of all priorities before this packet.

式中：index 表示源数据对应编码矩阵的相应行索引，FEC payload ID 表示数据包包的索引信息，number of packets of all priorities before this packet: 表示该数据包之前的所有优先级的数据包总数。

根据生成的 FEC 编码矩阵, 对编码矩阵进行初等行变换, 并对相应的接收到的数据包进行排序, 根据 repair_FEC_payload_ID 的值和各优先级下恢复符号的数目, 确定其属于哪一优先级, 从而调整恢复符号在符号块中的位置, 实现相同的变化, 如图 17 所示。根据接收到的不同优先级的数据包的数目不同, 分多种情况进行解码, 其流程图如图 10 所示。最终得到恢复后的源数据, 再根据包头中的 packet_sequence_number, 将得到的源数据包恢复为初始的顺序, 具体如图 18 所示。

本实施例中，服务器侧通过对数据流中不同数据进行重要程度的划分，对不同重要

程度的数据给予不同的保护。通过发送的数据包包头信息和信令信息进行标识。采用系统码的方式，根据信道状况和数据的重要程度，自适应地调整系统喷泉码的编码矩阵，以达到对不同重要程度数据的不等差错保护。终端侧解析信令信息和包头信息，同时根据接收到的不同优先级数据的数量，采用灵活的译码算法，实现灵活编译码，并根据包头信息最终恢复源数据流。因此可以达到如下的有益效果：

1) 节省资源，例如在终端侧（接收端），人为的根据情况舍弃已经接收的数据（例如 B 帧）则造成了传输资源的浪费，本实施例中的方法从源端解决问题，让不想要的包在传输过程中更大概率被丢弃，从而更大程度的保护了重要的数据包。

2) 实现个性化的传输方案，例如可以根据信道状况、用户体验等设计不等差错保护的方案，使得视频传输更加灵活、细致。另外，尽管发送端的编码方式灵活，接收端都可以根据列表信息正确的恢复出原始的数据。

3) 实现更加灵活的编码矩阵的设计，例如可以根据实际媒体的应用场景，基于媒体内容进行数据的优先级划分，根据实际需求更加灵活的设计 FEC 编码矩阵，达到不等差错保护的效果。

4) 实现更加灵活的译码方式，例如根据接收到的各优先级数据的数目，可以采用不同的译码算法，在有限的带宽资源下，可以最大程度的恢复优先级高的数据，提高对对优先级较高的数据的保护强度。

需要说明的是，本发明提供的所述基于媒体内容的自适应系统码 FEC 编译码方法中的步骤，可以利用所述基于媒体内容的自适应系统码 FEC 系统中对应的模块、装置、单元等予以实现，本领域技术人员可以参照所述系统的技术方案实现所述方法的步骤流程，即，所述系统中的实施例可理解为实现所述方法的优选例，在此不予赘述。

此外，本发明可以应用于计算机可读存储介质上，在计算机可读存储介质存储计算机程序，被处理器执行时实现上述实施例所述的基于媒体内容的自适应系统码 FEC 编译码方法的步骤。

本领域技术人员知道，除了以纯计算机可读程序代码方式实现本发明提供的系统及其各个装置、模块、单元以外，完全可以通过将方法步骤进行逻辑编程来使得本发明提供的系统及其各个装置、模块、单元以逻辑门、开关、专用集成电路、可编程逻辑控制器以及嵌入式微控制器等的形式来实现相同功能。所以，本发明提供的系统及其各项装置、模块、单元可以被认为是一种硬件部件，而对其内包括的用于实现各种功能的装置、

模块、单元也可以视为硬件部件内的结构；也可以将用于实现各种功能的装置、模块、单元视为既可以是实现方法的软件模块又可以是硬件部件内的结构。

以上对本发明的具体实施例进行了描述。需要理解的是，本发明并不局限于上述特定实施方式，本领域技术人员可以在权利要求的范围内做出各种变化或修改，这并不影响本发明的实质内容。在不冲突的情况下，本申请的实施例和实施例中的特征可以任意相互组合。

权利要求书

1、一种基于媒体内容的自适应系统码 FEC 编译码方法，其特征在于，包括：

根据媒体内容的重要程度，将源数据按照优先级划分为 N 类源数据包；N 为大于 1 的整数；

根据源数据和 N 类源数据包的优先级，生成 N 类中间码；

根据所述 N 类中间码，按照信道状况设定 N 类源数据的恢复数据，生成 N 类编码符号；

接收所述编码符号，根据译码需要对所述编码符号进行整理排序，对所述编码符号进行译码处理；

根据接收到的编码符号情况，优先译出优先级高的中间码，并根据该中间码恢复出相应的源数据包。

2、根据权利要求 1 所述的基于媒体内容的自适应系统码 FEC 编译码方法，其特征在于，所述根据源数据和 N 类源数据包的优先级，生成 N 类中间码，包括：

构建 N 类源数据包对应的编码矩阵，记为 A_i , $i=1,2,\dots,N$ ；其中：

$$A_i = \begin{bmatrix} G_p_i \\ G_ENC_i \end{bmatrix} \quad i = 1, 2, \dots, N$$

式中： A_i 表示第 i 个源数据包对应的编码矩阵， G_p_i 表示第 i 个源数据包对应的由 LDPC 矩阵和单位矩阵构成的联合矩阵， G_ENC_i 表示第 i 个源数据包对应的 LT 编码矩阵；

根据 N 类源数据包对应的编码矩阵构建源数据的编码矩阵 A ，所述源数据的编码矩阵如下：

$$A = \begin{bmatrix} G_p_1 & 0 & \dots & 0 \\ G_ENC_1 & 0 & \dots & 0 \\ 0 & G_p_i & \dots & 0 \\ 0 & G_ENC_i & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_p_N \\ 0 & 0 & \dots & G_ENC_N \end{bmatrix}$$

$$A \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_i \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} 0 \\ D_1 \\ 0 \\ \dots \\ D_i \\ \dots \\ 0 \\ D_N \end{bmatrix}$$

则

$$\begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \vdots \\ 0 \\ D_N \end{bmatrix}$$

式中: $C_1, C_2, \dots, C_i, \dots, C_N$ 分别对应 N 类源数据包的中间码, $D_1, D_2, \dots, D_i, \dots, D_N$ 分别对应 N 类源数据包的数据, 上标⁻¹为逆矩阵运算符。

3、根据权利要求 2 所述的基于媒体内容的自适应系统码 FEC 编译码方法, 其特征在于, 所述根据所述 N 类中间码, 按照信道状况设定 N 类源数据的恢复数据, 包括:

$$\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_N \\ R_1 \\ R_2 \\ \vdots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC_1} & 0 & \dots & 0 \\ 0 & G_{ENC_2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & G_{ENC_N} \\ G_{ENC_{11}} & 0 & \dots & 0 \\ G_{ENC_{12}} & G_{ENC_{21}} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ G_{ENC_{1N}} & G_{ENC_{2(N-1)}} & \dots & G_{ENC_{N1}} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{bmatrix}$$

式中: $R_1, R_2, \dots, R_i, \dots, R_N$ 分别对应 N 个源数据包的恢复数据, $G_{ENC_{1i}}$ 表示与 G_{ENC_1} 具有相同生成规则的 LT 编码矩阵; $G_{ENC_{i1}}$ 表示与 G_{ENC_i} 具有相同生成规则的 LT 编码矩阵; $i=1,2,\dots,N$ 。

4、根据权利要求 3 所述的基于媒体内容的自适应系统码 FEC 编译码方法, 其特征在于, 当 N 的值取 2 时, 针对不同重要程度的数据可以产生相应的中间码如下:

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \end{bmatrix}$$

改变 LT 编码矩阵的结构, 得到 $G_{ENC_{11}}, G_{ENC_{12}}, G_{ENC_{21}}, G_{ENC_{12}}$ 与 G_{ENC_1} 的生成方式相同; $G_{ENC_{21}}$ 与 G_{ENC_2} 的生成方式相同; $G_{ENC_{11}}$ 表示与 G_{ENC_1} 具有相同生成方式的 LT 编码矩阵; $G_{ENC_{21}}$ 表示与 G_{ENC_2} 具有相同生成方式的 LT 编码矩阵; $G_{ENC_{12}}$ 表示与 G_{ENC_1} 具有相同生成方式的 LT 编码矩阵;

得到的恢复数据 R_1 只与优先级为 1 的数据有关, 而恢复数据 R_2 与优先级 1 和 2 的数据均有关, 则得到:

$$\begin{bmatrix} D_1 \\ D_2 \\ R_1 \\ R_2 \end{bmatrix} = \begin{bmatrix} G_ENC_1 & 0 \\ 0 & G_ENC_2 \\ G_ENC_{11} & 0 \\ G_ENC_{12} & G_ENC_{21} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}。$$

5、根据权利要求4所述的基于媒体内容的自适应系统码 FEC 编译码方法，其特征在于，当N的值取2时，所述接收所述编码符号，对所述编码符号进行译码处理，包括：

在第一阶段解码时，根据恢复矩阵 W 以及接收到的源数据包得到中间码字；具体的：

$$W \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ 0 \\ REV_D_2 \\ REV_R_1 \\ REV_R_2 \end{bmatrix}$$

$$W = \begin{bmatrix} G_p_1 & 0 \\ REV_G_ENC_1 & 0 \\ 0 & G_p_2 \\ 0 & REV_G_ENC_2 \\ REV_G_ENC_{11} & 0 \\ REV_G_ENC_{12} & REV_G_ENC_{21} \end{bmatrix}$$

式中： $REV_G_ENC_1$ 表示根据接收到的编码符号的丢包情况，去除 G_ENC_1 中丢失包所对应的行后的 LT 编码矩阵； $REV_G_ENC_{11}$ 表示根据接收到的编码符号的丢包情况，去除 G_ENC_{11} 中丢失包所对应的行后的 LT 编码矩阵； $REV_G_ENC_{12}$ 表示根据接收到的编码符号的丢包情况，去除 G_ENC_{12} 中丢失包所对应的行后的 LT 编码矩阵； $REV_G_ENC_2$ 表示根据接收到的编码符号的丢包情况，去除 G_ENC_2 中丢失包所对应的行后的 LT 编码矩阵； $REV_G_ENC_{21}$ 表示根据接收到的编码符号的丢包情况，去除 G_ENC_{21} 中丢失包所对应的行后的 LT 编码矩阵；

$$\text{令 } REV_A = \begin{bmatrix} G_p_1 & 0 \\ REV_G_ENC_1 & 0 \\ REV_G_ENC_{11} & 0 \\ 0 & G_p_2 \\ 0 & REV_G_ENC_2 \\ REV_G_ENC_{12} & REV_G_ENC_{21} \end{bmatrix}；\text{并根据接收到的对应优先级下源数据}$$

包的数量以及接收到的数据包总数量进行解码；

接收到的编码符号总数小于源数据生成的编码符号的个数，并且 $REV_A_1 = \begin{bmatrix} G_p_1 \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix}$ 的行数小于列数时； C_1 、 C_2 均无法解出；

接收到的编码符号总数小于源数据生成的编码符号的个数，且 $REV_A_1 =$

$\begin{bmatrix} G_{p1} \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix}$ 的行数大于等于列数时；若 $REV_A_1 = \begin{bmatrix} G_{p1} \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix}$ 的行数大于等于

列数，则 REV_A_1 可逆，对应的逆矩阵记为 $REV_A_1^{-1}$ ；则可得：

$$[C_1] = REV_A^{-1} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix} \text{ 其中：} C_2 \text{ 无解；}$$

式中： REV_D_1 表示接收到的优先级为 1 的源数据， REV_R_1 表示接收到的优先级为 1 源数据的恢复数据；

接收到的编码符号总数大于等于源数据生成的编码符号的个数，且 $REV_A_1 =$

$\begin{bmatrix} G_{p1} \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix}$ 的行数大于等于列数时；若 $REV_A_1 = \begin{bmatrix} G_{p1} \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix}$ 的行数大于等于

列数，则 REV_A_1 可逆，对应的逆矩阵记为 $REV_A_1^{-1}$ ；若 $REV_A_2 = \begin{bmatrix} G_{p2} \\ REV_G_ENC_2 \\ REV_G_ENC_{21} \end{bmatrix}$ 的行

数大于等于列数，则 REV_A_2 可逆，对应的逆矩阵记为 $REV_A_2^{-1}$ ；若 REV_A_1 和 REV_A_2 都

可逆，则 REV_A 可逆，对应的逆矩阵记为 REV_A^{-1} ，记 $B_{1,1} = \begin{bmatrix} 0 \\ 0 \\ REV_G_ENC_{12} \end{bmatrix}$ ，则

$$REV_A^{-1} = \begin{bmatrix} REV_A_1^{-1} & 0 \\ REV_A_2^{-1} B_{1,1} REV_A_1^{-1} & REV_A_2^{-1} \end{bmatrix}； \text{ 得到：}$$

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = REV_A^{-1} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix} = \begin{bmatrix} REV_A_1^{-1} & 0 \\ REV_A_2^{-1} B_{1,1} REV_A_1^{-1} & REV_A_2^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix}；$$

若 $REV_A_2 = \begin{bmatrix} G_{p2} \\ REV_G_ENC_2 \\ REV_G_ENC_{21} \end{bmatrix}$ 的行数小于列数，则 REV_A_2 不存在逆矩阵，只能求出

$$[C_1] = REV_A^{-1} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix}；$$

式中： REV_D_2 表示接收到的优先级为 2 的源数据， REV_R_2 表示接收到的优先级为 1 和 2 源数据的恢复数据；

接收到的编码符号总数大于等于源数据生成的编码符号的个数，且 $REV_A_1 =$

$\begin{bmatrix} G_{p1} \\ REV_G_ENC_1 \\ REV_G_ENC_{11} \end{bmatrix}$ 的行数小于列数， $REV_A_2 = \begin{bmatrix} G_{p2} \\ REV_G_ENC_2 \\ REV_G_ENC_{21} \end{bmatrix}$ 的行数大于等于列数时；

使用高斯消元法直接对矩阵 REV_A 进行求解，即解下列线性方程组：

$$\begin{bmatrix} G_{p1} & 0 \\ REV_G_ENC_1 & 0 \\ REV_G_ENC_{11} & 0 \\ 0 & G_{p2} \\ 0 & REV_G_ENC_2 \\ _REV_G_ENC_{12} & REV_G_ENC_{21} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix}$$

得到 C_1 、 C_2 。

6、根据权利要求3所述的基于媒体内容的自适应系统码 FEC 编译码方法，其特征在于，由于部分数据发生丢失，需要根据接收到的编码符号，生成恢复矩阵，相当于擦除掉丢失的编码符号对应的矩阵行，形成相应的恢复矩阵，记为 REV_G_ENC ；根据接收的数据包的数目不同，分不同情况生成中间码字；具体的，包括如下步骤：

步骤 A1：假设接收到 l 类优先级的数据包；则得到如下的恢复矩阵方程：

$$\begin{bmatrix} G_{p1} & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ 0 & G_{p2} & \dots & 0 \\ 0 & REV_G_ENC_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{pl} \\ 0 & 0 & \dots & REV_G_ENC_l \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ REV_G_ENC_{1l} & REV_G_ENC_{2(l-1)} & \dots & REV_G_ENC_{l1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ 0 \\ REV_D_2 \\ \dots \\ 0 \\ REV_D_l \\ REV_R_1 \\ REV_R_2 \\ \dots \\ REV_R_l \end{bmatrix}$$

对恢复矩阵方程进行初等行变换，得到：

$$\begin{bmatrix} G_{p1} & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_{p2} & \dots & 0 \\ 0 & REV_G_ENC_2 & \dots & 0 \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{pl} \\ 0 & 0 & \dots & REV_G_ENC_l \\ REV_G_ENC_{1l} & REV_G_ENC_{2(l-1)} & \dots & REV_G_ENC_{l1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \\ \dots \\ 0 \\ REV_D_l \\ REV_R_l \end{bmatrix}$$

式中： $REV_G_ENC_1$ 表示根据接收到的编码符号的丢包情况，去除 G_ENC_1 中丢失包所对应的行后的 LT 编码矩阵； $REV_G_ENC_{11}$ 表示根据接收到的编码符号的丢包情况，去除 G_ENC_{11} 中丢失包所对应的行后的 LT 编码矩阵； $REV_G_ENC_{12}$ 表示根据接收到的编码符号的丢包情况，去除 G_ENC_{12} 中丢失包所对应的行后的 LT 编码矩阵； $REV_G_ENC_2$ 表示根据接收到的编码符号的丢包情况，去除 G_ENC_2 中丢失包所对应的

行后的 LT 编码矩阵; $REV_G_ENC_{21}$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_{21} 中丢失包所对应的行后的 LT 编码矩阵; REV_D_1 表示接收到的优先级为 1 的源数据, REV_R_1 表示接收到的优先级为 1 源数据的恢复数据; REV_D_2 表示接收到的优先级为 2 的源数据, REV_R_2 表示接收到的优先级为 1 和 2 源数据的恢复数据; $REV_G_ENC_{1l}$ 表示根据接收到的编码符号的丢包情况, 去除 G_ENC_{1l} 中丢失包所对应的行后的 LT 编码矩阵, REV_D_l 表示接收到的优先级为 l 的源数据, REV_R_l 表示接收到的优先级为 1 到 l 的源数据的恢复数据, C_l 表示生成的第 l 类中间码;

$$\text{步骤 A2: 令 } REV_A = \begin{bmatrix} G_p_1 & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_p_2 & \dots & 0 \\ 0 & REV_G_ENC_2 & \dots & \dots \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_p_l \\ 0 & 0 & \dots & REV_G_ENC_l \\ REV_G_ENC_{1l} & REV_G_ENC_{2(l-1)} & \dots & REV_G_ENC_{l1} \end{bmatrix}, \text{ 令}$$

$R(i)$ 为接收到并经扩展后第 i 级编码符号数, $L(i)$ 为编码端每级符号长度, 其中: $1 \leq i \leq l$;

计算每级生成矩阵的秩:

$$\text{eachRank}(i) = \min(\text{Len}(R(i)), \text{sum}(L(1:i)));$$

式中: $\text{eachRank}(i)$ 表示优先级为 i 的数据所对应的子矩阵的秩, 其中 $i=2$ 时所对应

的子矩阵为 $\begin{bmatrix} 0 & G_p_2 \\ 0 & REV_G_ENC_2 \\ REV_G_ENC_{12} & REV_G_ENC_{21} \end{bmatrix}$, $\text{Len}(R(i))$ 表示接收到的优先级为 i 的编码

符号数的长度, $\text{sum}(L(1:i))$ 表示优先级为 i 的子矩阵的列数, $\min()$ 表示取最小值运算;

计算累积矩阵的秩:

$$\text{cumRank}(i) = \min(\text{cumRank}(i-1) + \text{eachRank}(i), \text{sum}(L(1:i))),$$

式中: $\text{cumRank}(i)$ 表示优先级为 1 到优先级为 i 所对应的子矩阵秩之和,

$\text{cumRank}(i-1)$ 表示优先级为 1 到优先级为 $i-1$ 所对应的子矩阵秩之和;

其中 $\text{cumRank}(1) = \text{eachRank}(1)$;

依次判断 $\text{cumRank}(i) \geq \text{sum}(L(1:i))$: 若成立, $\text{cumFullRank}(i) = 1$; 否则 $\text{cumFullRank}(i) = 0$; $\text{cumFullRank}(i)$ 表示优先级为 1 到优先级为 i 所对应的子矩阵组成的矩阵满秩;

步骤 A3: 根据接收到的每级编码符号数, 分情况分别求中间码 C_1 、 $C_2 \dots C_l$, 设置 i 初始值为 1:

若满足 $\text{cumFullRank}(i) == 1 \&\& \text{cumFullRank}(1:i) > 0$, 使用 RFC 6330 译码方法解出 $C_1 \sim C_i$, 其中: $\&\&$ 表示与运算, $\text{cumFullRank}(1:i)$ 表示从优先级为 1 到优先级为 i 各子矩阵所组成的矩阵满秩; C_1 表示优先级为 1 的数据所对应的中间码字, C_i 表示优先级为 i 的数据所对应的中间码;

$$\text{设 } REV_A_{\Delta i} = \begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & REV_G_ENC_2 \dots & \dots & \dots \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_i} \\ 0 & 0 & \dots & REV_G_ENC_i \\ REV_G_ENC_{i1} & REV_G_ENC_{2(i-1)} & \dots & REV_G_ENC_{i1} \end{bmatrix}, \text{ 得到:}$$

$$REV_A_{\Delta i}^{-1} = \begin{bmatrix} A_1^{-1} & 0 & 0 & 0 \\ X_{1,2} & A_2^{-1} & 0 & 0 \\ \dots & \dots & \dots & 0 \\ X_{1,i} & X_{2(i-1)} & \dots & A_i^{-1} \end{bmatrix}, \text{ 其中:}$$

$$X_{1,2} = A_2^{-1} B_{1,2} A_1^{-1}, \quad X_{a,b} = A_b^{-1} B_{a,b} A_a^{-1} + \sum_{k=a+1}^{b-1} A_b^{-1} B_{k,b} X_{a,k},$$

式中: $REV_A_{\Delta i}$ 表示到优先级为 i 时所有子矩阵组成的满秩矩阵, $B_{1,2}$ 表示子矩阵

$$\begin{bmatrix} 0 \\ 0 \\ REV_G_ENC_{12} \end{bmatrix}, \quad X_{1,2} \text{表示对子矩阵 } B_{1,2} \text{求逆运算后的子矩阵, } X_{a,b} \text{表示子矩阵 } B_{a,b} \text{求逆}$$

运算后得到的子矩阵, $B_{a,b}$ 表示子矩阵 $\begin{bmatrix} 0 \\ 0 \\ REV_G_ENC_{ab} \end{bmatrix}$, a 的取值范围为 $1 \sim i$, b 的取值

范围为 $2 \sim i$, A_i^{-1} 表示子矩阵 $\begin{bmatrix} G_{p_i} \\ REV_G_ENC_i \\ REV_G_ENC_{i1} \end{bmatrix}$ 的逆矩阵;

$$\text{令 } C_{i_temp} = REV_A_i^{-1} \begin{bmatrix} 0 \\ REV_D_i \\ REV_R_i \end{bmatrix}, \text{ 其中, } REV_A_i = \begin{bmatrix} G_{p_i} \\ REV_G_ENC_i \\ REV_G_ENC_{i1} \end{bmatrix}; \quad REV_A_i^{-1} \text{为}$$

REV_A_i 的逆矩阵, 则得到:

$$C_1 = C_{1_temp}$$

$$C_2 = C_{2_temp} + X_{1,2} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix} = C_{2_temp} + A_2^{-1} B_{1,2} C_1$$

$$C_3 = C_{3_temp} + X_{1,3} \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \end{bmatrix} + X_{2,3} \begin{bmatrix} 0 \\ REV_D_2 \\ REV_R_2 \end{bmatrix} = C_{2_temp} + A_3^{-1} B_{1,3} C_1 + A_3^{-1} B_{2,3} C_2$$

根据数学归纳法, 得到:

$$C_i = C_{i_temp} + \sum_{j=1}^{i-1} X_{j_i} \begin{bmatrix} 0 \\ REV_D_j \\ REV_R_j \end{bmatrix} = C_{i_temp} + \sum_{j=1}^{i-1} A_i^{-1} B_{j_i} C_j$$

求得 C_1 、 $C_2 \dots C_i$, 其中, 不能解出 $C_{i+1} \dots C_l$;

若满足 $\text{cumFullRank}(i) == 1$, 且不满足 $\text{cumFullRank}(1:i) > 0$, 使用高斯消元译码方法解出 $C_1 \sim C_i$

$$\text{设 } REV_A_{\Delta i} = \begin{bmatrix} G_p_1 & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_p_2 & \dots & 0 \\ 0 & REV_G_ENC_2 \dots & \dots & \dots \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_p_i \\ 0 & 0 & \dots & REV_G_ENC_i \\ REV_G_ENC_{1i} & REV_G_ENC_{2(i-1)} & \dots & REV_G_ENC_{i1} \end{bmatrix}, \text{ 通过高斯消}$$

元法解线性方程组:

$$\begin{bmatrix} G_p_1 & 0 & \dots & 0 \\ REV_G_ENC_1 & 0 & \dots & 0 \\ REV_G_ENC_{11} & 0 & \dots & 0 \\ 0 & G_p_2 & \dots & 0 \\ 0 & REV_G_ENC_2 \dots & \dots & \dots \\ REV_G_ENC_{12} & REV_G_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_p_i \\ 0 & 0 & \dots & REV_G_ENC_i \\ REV_G_ENC_{1i} & REV_G_ENC_{2(i-1)} & \dots & REV_G_ENC_{i1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_i \end{bmatrix} = \begin{bmatrix} 0 \\ REV_D_1 \\ REV_R_1 \\ 0 \\ REV_D_2 \\ REV_R_2 \\ \dots \\ 0 \\ REV_D_i \\ REV_R_i \end{bmatrix}$$

求得 C_1 、 $C_2 \dots C_i$, 其中, 不能解出 $C_{i+1} \dots C_l$;

对于任一 i 值, 若不满足 $\text{cumFullRank}(i) == 1$ 则相应 C_i 不能解出; 直到 $i == 1$, 也不满足 $\text{cumFullRank}(1:i) > 0$ 时, 则 C_1 、 $C_2 \dots C_l$ 全部不能解出, 译码失败;

步骤 A4: 令 i 的值自减 1, 返回执行步骤 A3。

7、根据权利要求 1 所述的基于媒体内容的自适应系统码 FEC 编译码方法, 其特征在于, 还包括:

解析源数据包中每个子数据包包头的重要性信息;

根据所述重要性信息, 将所述源数据包中每个子数据包分别映射到不同的优先级中;

按照子数据包的优先级重新排列子数据包, 得到中间数据包;

将所述中间数据包转化为多个符号 symbol，并对转化成多个符号 symbol 的中间数据包进行前向纠错 FEC 编码后，得到包含多个符号 symbol 的恢复数据包；

在所述恢复数据包中加入索引字段和/或优先级指示字段，所述索引字段用于指示恢复数据包中子数据包的顺序和/或指示恢复数据包中子数据包在所述优先级指示字段对应的优先级下的顺序；在所述中间数据包中也加入索引字段和/或优先级指示字段，所述索引字段用于指示中间数据包中子数据包的顺序和/或指示中间数据包中子数据包在所述优先级指示字段对应的优先级下的顺序；

在所述恢复数据包和中间数据包中加入相应的指示信息，或者在发送恢复数据包和中间数据包时发送相应的指示信息，所述指示信息用于指示恢复数据包在各优先级下的子数据包，以及属于相应优先级的子数据包的数目或者所述子数据包占所述恢复数据包的比例；所述指示信息还用于指示中间数据包在各个优先级下的子数据包，以及属于相应优先级的子数据包的数目或者所述子数据包占所述中间数据包的比例；

向终端侧发送包含指示信息的恢复数据包、中间数据包，或者向终端侧发送恢复数据包、中间数据包以及相应的指示信息。

8、根据权利要求 7 所述的基于媒体内容的自适应系统码 FEC 编译码方法，其特征在于，所述解析源数据包中每个子数据包包头的重要性信息，包括：

读取每个子数据包包头中的优先 priority 字段，根据所述 priority 字段中的信息确定对应子数据包的优先级。

9、根据权利要求 8 所述的基于媒体内容的自适应系统码 FEC 编译码方法，其特征在于，所述根据所述重要性信息，将所述源数据包中每个子数据包分别映射到不同的预设优先级中，包括：

确定所述源数据包实际采用的优先级个数，以及各优先级下与所述 priority 字段的映射关系；

按照所述映射关系，将子数据包映射到相应的预设优先级中。

10、根据权利要求 7 所述的基于媒体内容的自适应系统码 FEC 编译码方法，其特征在于，所述将所述中间数据包转化为多个符号 symbol，并对转化成多个 symbol 的中间数据包进行前向纠错 FEC 编码后，得到包含多个 symbol 的恢复数据包，包括：

对所述中间数据包采用 RaptorQ 系统码的自适应 FEC 编码方式进行 FEC 编码，得到如下的中间码字：

$$\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_N \end{bmatrix}$$

式中： A_i 表示第 i 类源数据包对应的编码矩阵， $C_1, C_2, \dots, C_i, \dots, C_N$ 分别对应 N 类源数据包的中间码， $D_1, D_2, \dots, D_i, \dots, D_N$ 分别对应 N 类源数据包的数据，上标 -1 为逆矩阵运算符；

根据所述中间码字，得到恢复数据包，并在所述恢复数据包中附加 FEC payload ID，其中，所述恢复数据包中的恢复数据如下：

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_N \\ R_1 \\ R_2 \\ \dots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC1} & 0 & \dots & 0 \\ 0 & G_{ENC2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & G_{ENCN} \\ G_{ENC11} & 0 & \dots & 0 \\ G_{ENC12} & G_{ENC21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC1N} & G_{ENC2(N-1)} & \dots & G_{ENCN1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix}$$

式中： $R_1, R_2, \dots, R_i, \dots, R_N$ 分别对应 $1 \sim N$ 个源数据包的恢复数据， G_{ENCi} 表示第 i 个源数据包对应的 LT 编码矩阵， G_{ENC1i} 表示与 G_{ENC1} 具有相同生成规则的 LT 编码矩阵； G_{ENCi1} 表示与 G_{ENCi} 具有相同生成规则的 LT 编码矩阵； $i=1,2,\dots,N$ 。

11、一种基于媒体内容的自适应系统码 FEC 编译码方法，其特征在在于，包括：

接收服务器发送的恢复数据包、中间数据包，或者接收恢复数据包、中间数据以及相应的指示信息；

解析出所述恢复数据包和中间数据包中包含的索引字段，并根据所述索引字段生成编码矩阵，其中，所述索引字段用于指示恢复数据包中子数据包的顺序，以及指示中间数据包在各个优先级下的子数据包；或者

通过恢复数据包和中间数据包的优先级指示字段直接获取索引信息，并根据所述索引信息生成编码矩阵，其中，所述索引信息用于指示恢复数据包中子数据包的顺序，以及指示中间数据包在各个优先级下的子数据包；

对所述编码矩阵进行前向纠错 FEC 译码，恢复出中间数据包；

解析所述中间数据包包头中的信息，重新排列所述中间数据包的子数据包的顺序，恢复出源数据包。

12、根据权利要求 11 所述的基于媒体内容的自适应系统码 FEC 编译码方法，其特征在于，所述解析出所述恢复数据包和中间数据包中包含的索引字段，包括：

通过以下公式得到源数据对应编码矩阵的相应行的索引字段：

$$\text{index} = \text{FEC payload ID} - \\ \text{number of packets of all priorities before this packet}$$

式中：index 表示源数据对应编码矩阵的相应行索引，FEC payload ID 表示数据包包头的索引信息，number of packets of all priorities before this packet 表示该数据包之前的所有优先级的数据包总数。

13、根据权利要求 11 所述的基于媒体内容的自适应系统码 FEC 编译码方法，其特征在于，所述对所述编码矩阵进行前向纠错 FEC 译码，恢复出中间数据包，包括：

对所述编码矩阵进行初等行变换，并对接收到的相应恢复数据包中的符号 symbol 进行排序，根据恢复的数据包包头的索引信息 FEC_payload_ID 的值和对应优先级下符号 symbol 的数目确定该符号 symbol 所属的优先级和变换后的位置；或者

根据恢复数据包包头中的优先级字段和该优先级字段对应优先级下的索引信息，直接确定符号 symbol 所属的优先级和变换后的位置；

调整恢复数据包中的符号 symbol 的顺序，并通过高斯消元译码器和 RFC6330 译码器进行译码后，得到恢复的包含多个 symbol 的中间数据包；

将包含多个 symbol 的中间数据包转化为中间数据包。

14、一种基于媒体内容的自适应系统码 FEC 编译码装置，其特征在于，包括：

解析模块，用于解析源数据包中每个子数据包包头的重要性信息；

映射模块，用于根据所述重要性信息，将所述源数据包中每个子数据包分别映射到不同的预设优先级中；

排序模块，用于按照预设优先级重新排列子数据包，得到中间数据包；

编码模块，用于将所述中间数据包转化为多个符号 symbol，并对转化成多个 symbol 的中间数据包进行前向纠错 FEC 编码后，得到包含多个 symbol 的恢复数据包；

索引模块，用于在所述恢复数据包中加入索引字段和/或优先级指示字段，所述索引字段用于指示恢复数据包中子数据包的顺序和/或指示恢复数据包中子数据包在所述优先级指示字段对应的优先级下的顺序；在所述中间数据包中也加入索引字段和/或优先级指示字段，所述索引字段用于指示中间数据包中子数据包的顺序和/或指示中间数据包中子数据包在所述优先级指示字段对应的优先级下的顺序；；

指示信息生成模块，用于在所述恢复数据包和中间数据包中加入相应的指示信息，

或者在发送恢复数据包和中间数据包时发送相应的指示信息；其中，所述指示信息用于指示恢复数据包在各优先级下的子数据包，以及属于相应优先级的子数据包的数目或者所述子数据包占所述恢复数据包的比例；所述指示信息还用于指示中间数据包在各个优先级下的子数据包，以及属于相应优先级的子数据包的数目或者所述子数据包占所述中间数据包的比例；

发送模块，用于向终端侧发送所述恢复数据包、中间数据包和相应的信令信息，或者向终端侧发送恢复数据包、中间数据包以及相应的指示信息。

15、根据权利要求 14 所述的基于媒体内容的自适应系统码 FEC 编译码装置，其特征在于，所述编码模块，具体用于：

对所述中间数据包采用 RaptorQ 系统码的自适应 FEC 编码方式进行 FEC 编码，得到如下的中间码字：

$$\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_N \end{bmatrix}$$

式中： A_i 表示第 i 类源数据包对应的编码矩阵， $C_1, C_2, \dots, C_i, \dots, C_N$ 分别对应 N 类源数据包的中间码， $D_1, D_2, \dots, D_i, \dots, D_N$ 分别对应 N 类源数据包的数据，上标 -1 为逆矩阵运算符；

根据所述中间码字，得到恢复数据包，并在所述恢复数据包中附加 FEC payload ID，其中，所述恢复数据包中的恢复数据如下：

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_N \\ R_1 \\ R_2 \\ \dots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC1} & 0 & \dots & 0 \\ 0 & G_{ENC2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & G_{ENCN} \\ G_{ENC11} & 0 & \dots & 0 \\ G_{ENC12} & G_{ENC21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC1N} & G_{ENC2(N-1)} & \dots & G_{ENCN1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix}$$

式中： $R_1, R_2, \dots, R_i, \dots, R_N$ 分别对应 $1 \sim N$ 个源数据包的恢复数据， G_{ENCi} 表示第 i 个源数据包对应的 LT 编码矩阵， G_{ENC1i} 表示与 G_{ENC1} 具有相同生成规则的 LT 编码矩阵； G_{ENCi1} 表示与 G_{ENCi} 具有相同生成规则的 LT 编码矩阵； $i=1,2,\dots,N$ 。

16、一种基于媒体内容的自适应系统码 FEC 编译码装置，其特征在于，包括：

接收模块，用于接收服务器发送的恢复数据包、中间数据包，或者接收恢复数据包、中间数据以及相应的指示信息；

编码矩阵生成模块，用于解析出所述恢复数据包和中间数据包中包含的索引字段，并根据所述索引字段生成编码矩阵，其中，所述索引字段用于指示恢复数据包中子数据包的顺序，以及指示中间数据包在各个优先级下的子数据包；或者

通过恢复数据包和中间数据包的优先级指示字段直接获取索引信息，并根据所述索引信息生成编码矩阵，其中，所述索引信息用于指示恢复数据包中子数据包的顺序，以及指示中间数据包在各个优先级下的子数据包；；

译码模块，用于对所述编码矩阵进行前向纠错 FEC 译码，恢复出中间数据包；

恢复模块，用于解析所述中间数据包包头中的信息，重新排列所述中间数据包的子数据包的顺序，恢复出源数据包。

17、根据权利要求 16 所述的基于媒体内容的自适应系统码 FEC 编译码装置，其特征在于，所述编码矩阵生成模块，具体用于：

通过以下公式得到源数据对应编码矩阵的相应行的索引字段：

$$index = FEC \text{ payload ID} - \\ \text{number of packets of all priorities before this packet}$$

式中：index 表示源数据对应编码矩阵的相应行索引，FEC payload ID 表示数据包包头的索引信息，number of packets of all priorities before this packet 表示该数据包之前的所有优先级的数据包总数。

18、根据权利要求 16 所述的基于媒体内容的自适应系统码 FEC 编译码装置，其特征在于，所述译码模块，具体用于：

对所述编码矩阵进行初等行变换，并对接收到的相应恢复数据包中的符号 symbol 进行排序，根据恢复的数据包包头的索引信息 FEC_payload_ID 的值和对应优先级下符号 symbol 的数目确定该符号 symbol 所属的优先级和变换后的位置；或者

根据恢复数据包包头中的优先级字段和该优先级字段对应优先级下的索引信息，直接确定符号 symbol 所属的优先级和变换后的位置；

调整恢复数据包中的符号 symbol 的顺序，并通过高斯消元译码器和 RFC6330 译码器进行译码后，得到恢复的包含多个 symbol 的中间数据包；

将包含多个 symbol 的中间数据包转化为中间数据包。

19、一种基于媒体内容的自适应系统码 FEC 编译码系统，其特征在于，包括服务器和至少一个终端，其中，所述服务器用于执行权利要求 7-10 中任一项所述的基于媒

体内容的自适应系统码 FEC 编译码方法，所述终端用于执行权利要求 11-13 任一项所述的基于媒体内容的自适应系统码 FEC 编译码方法。

20、一种存储有计算机程序的计算机可读存储介质，其特征在于，所述计算机程序被处理器执行时实现权利要求 7 至 13 中任一项所述的基于图像的条纹布料检测定位裁剪方法的步骤。

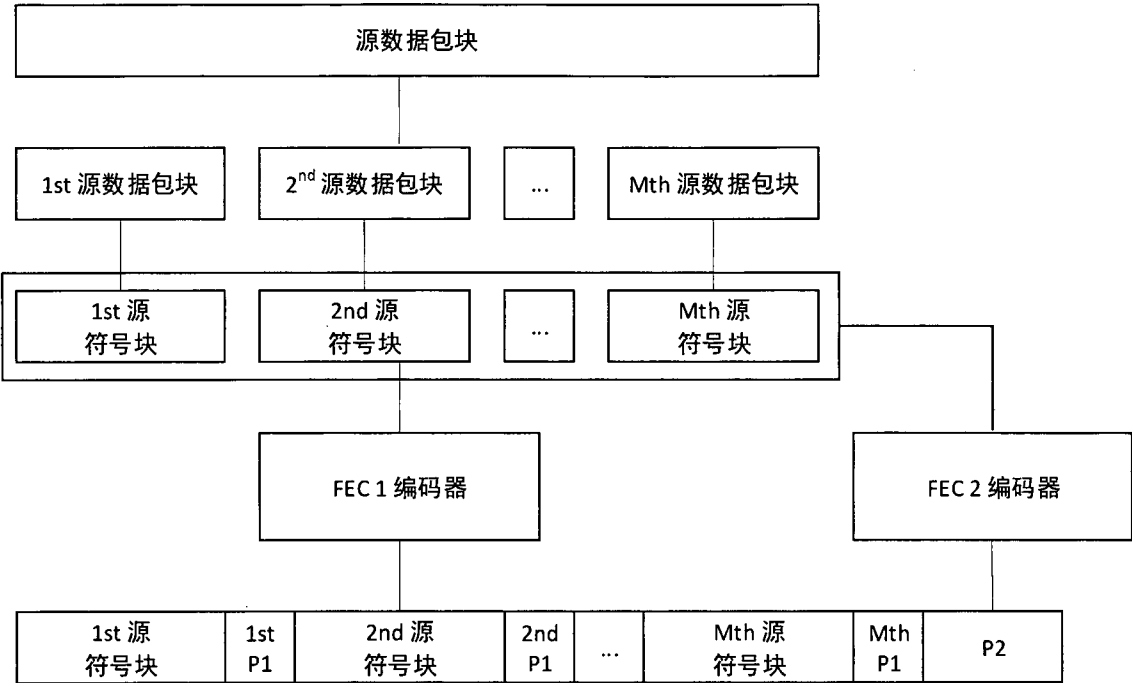


图 1

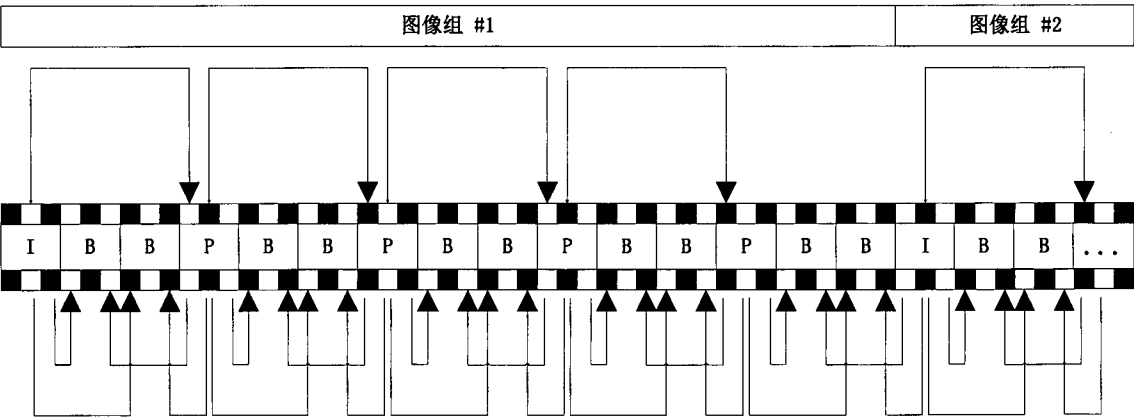


图 2

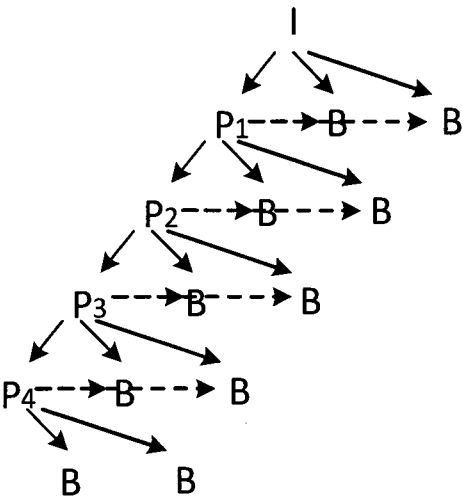


图 3

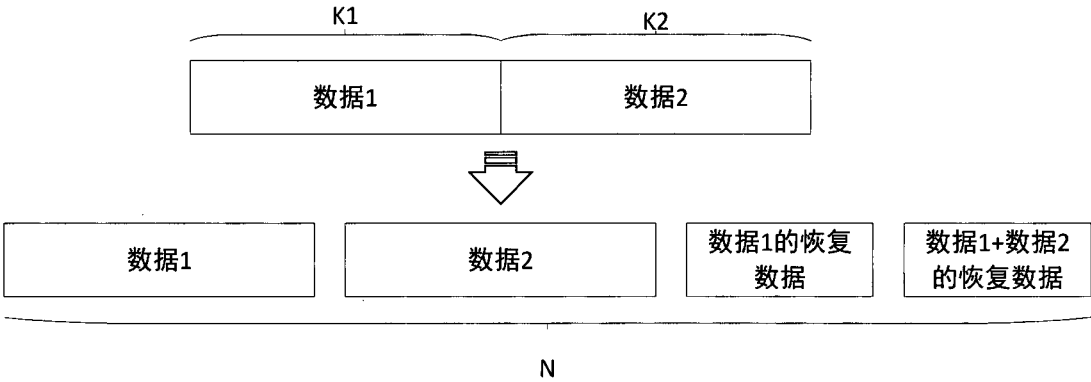


图 4

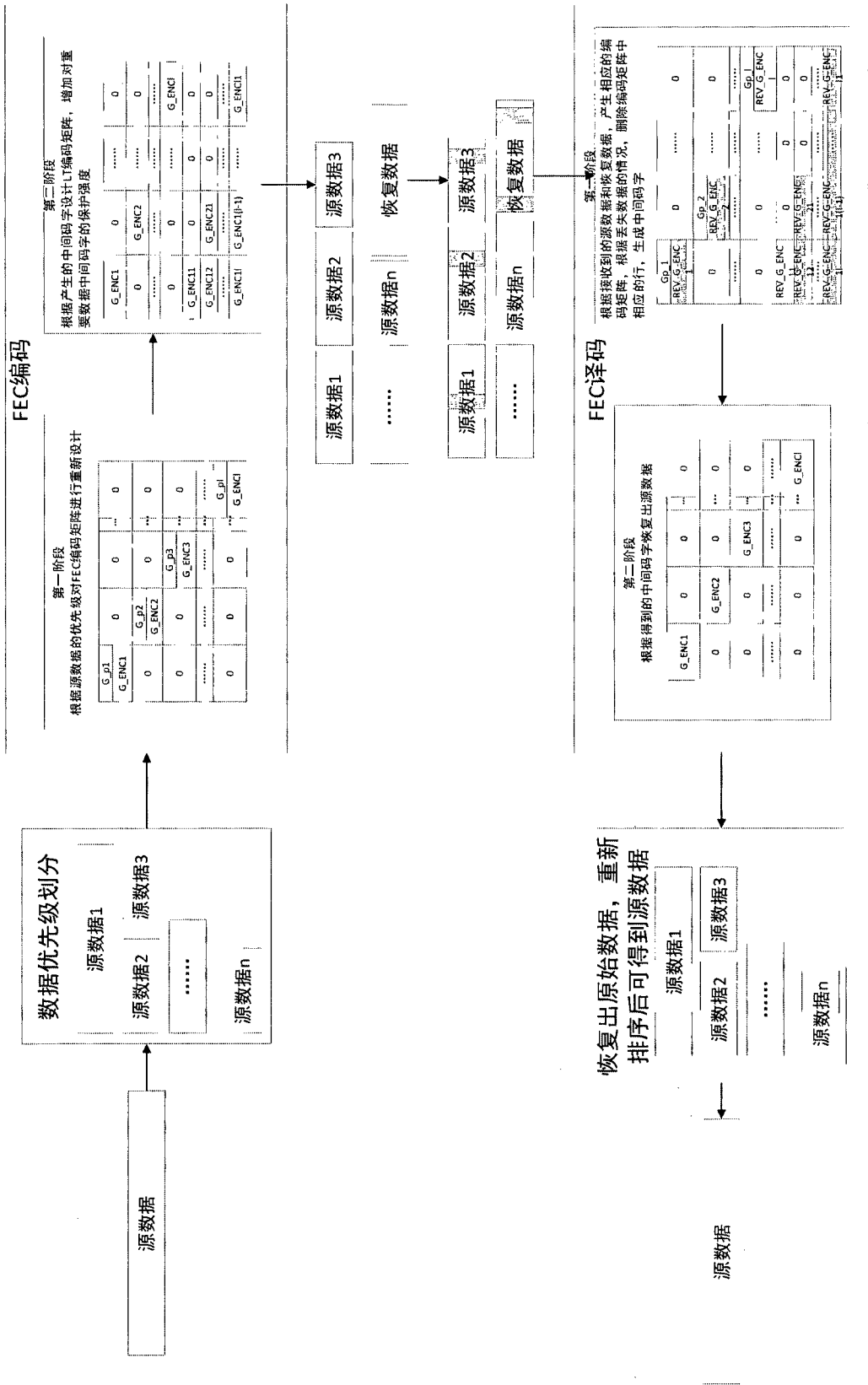


图 5

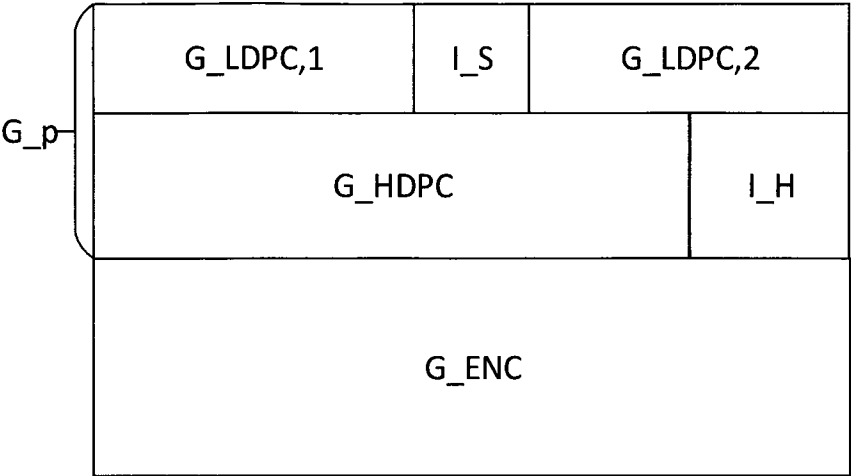


图 6

G_p	0
G_ENC1	
0	G_p
	G_ENC2

图 7

G_ENC1	0
0	G_ENC2
G_ENC1'	0
G_ENC1''	G_ENC2'

图 8

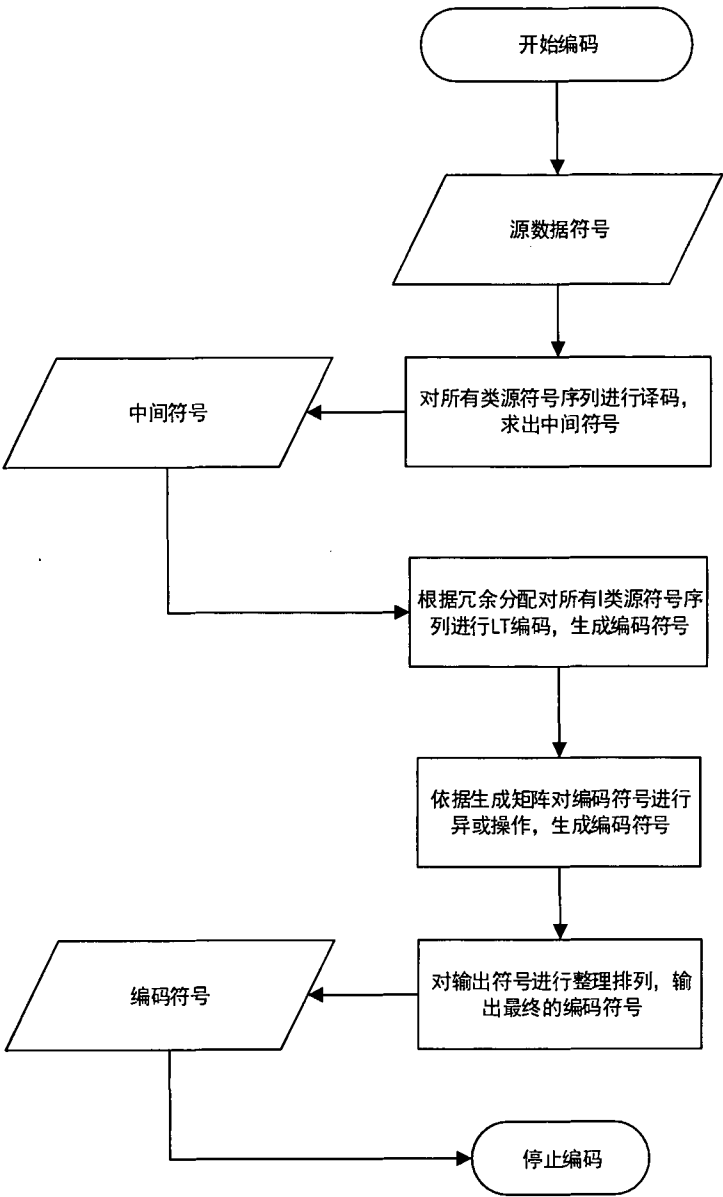


图 9

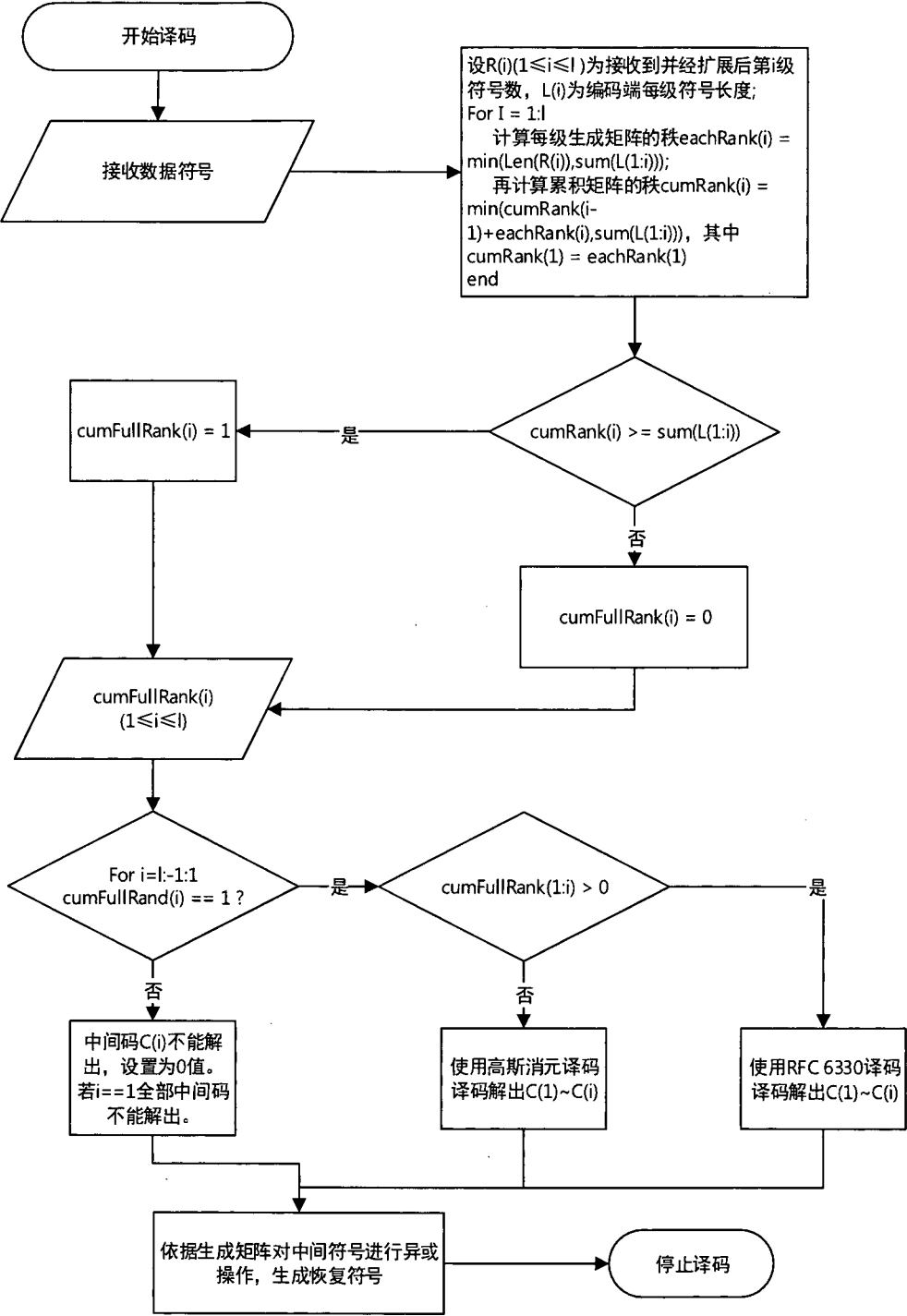


图 10

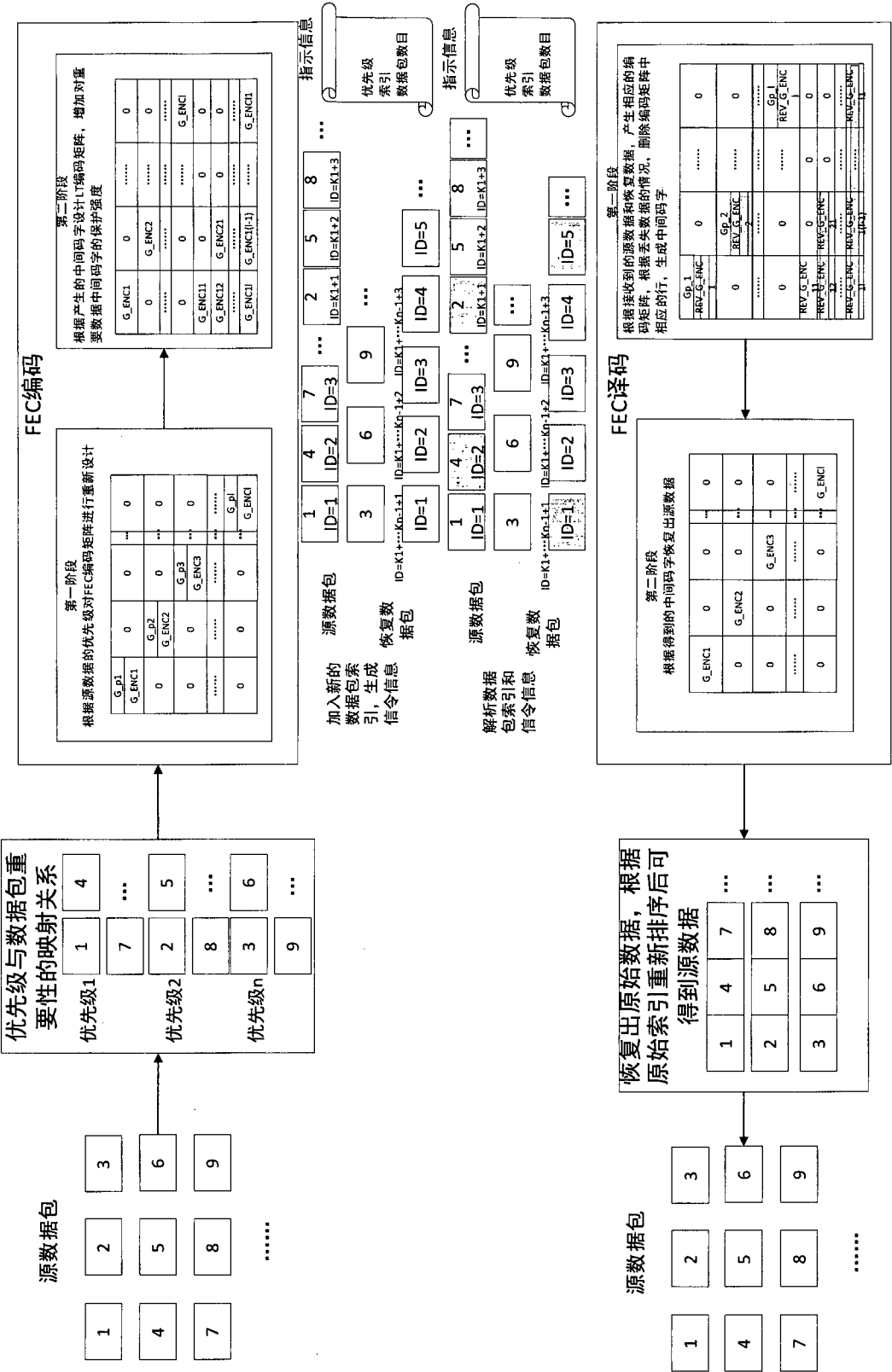


图 11

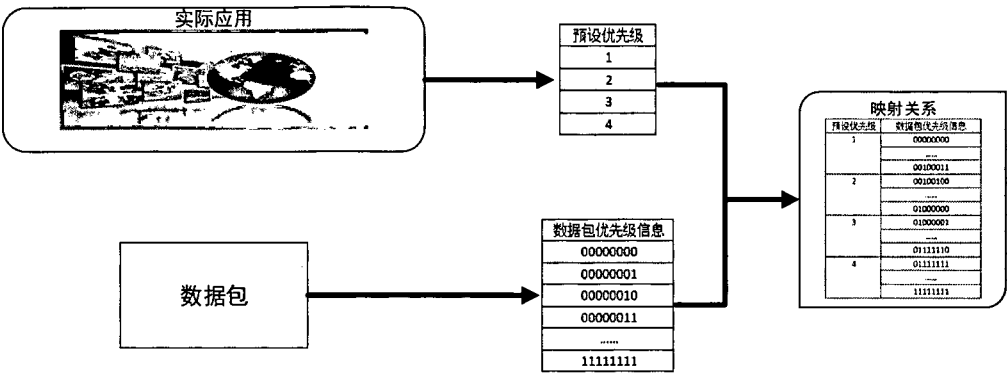


图 12

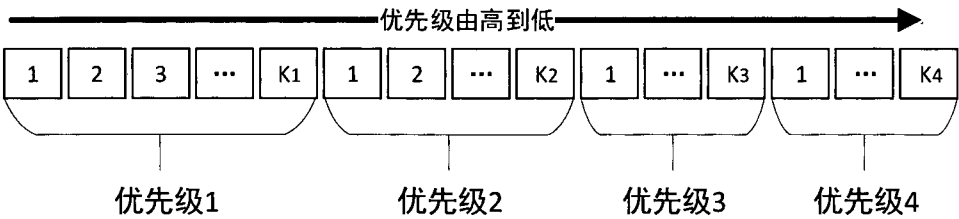


图 13

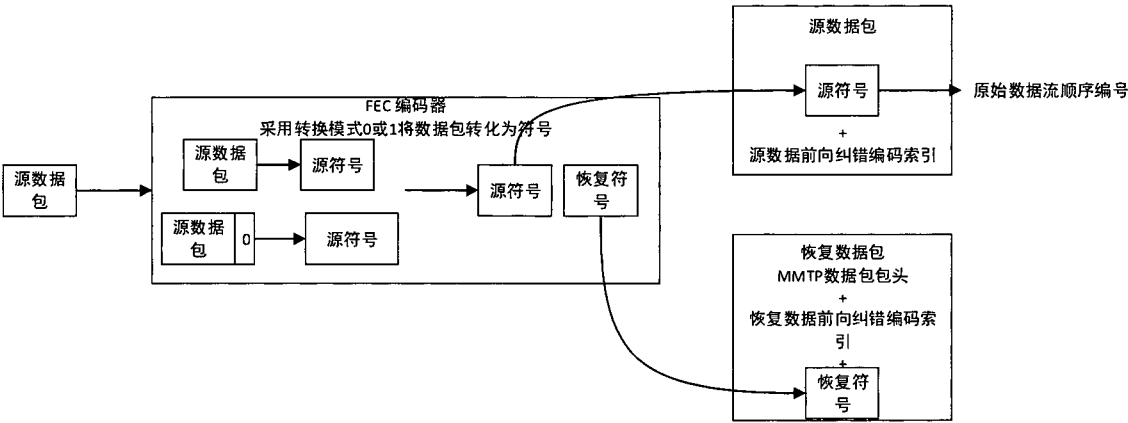


图 14

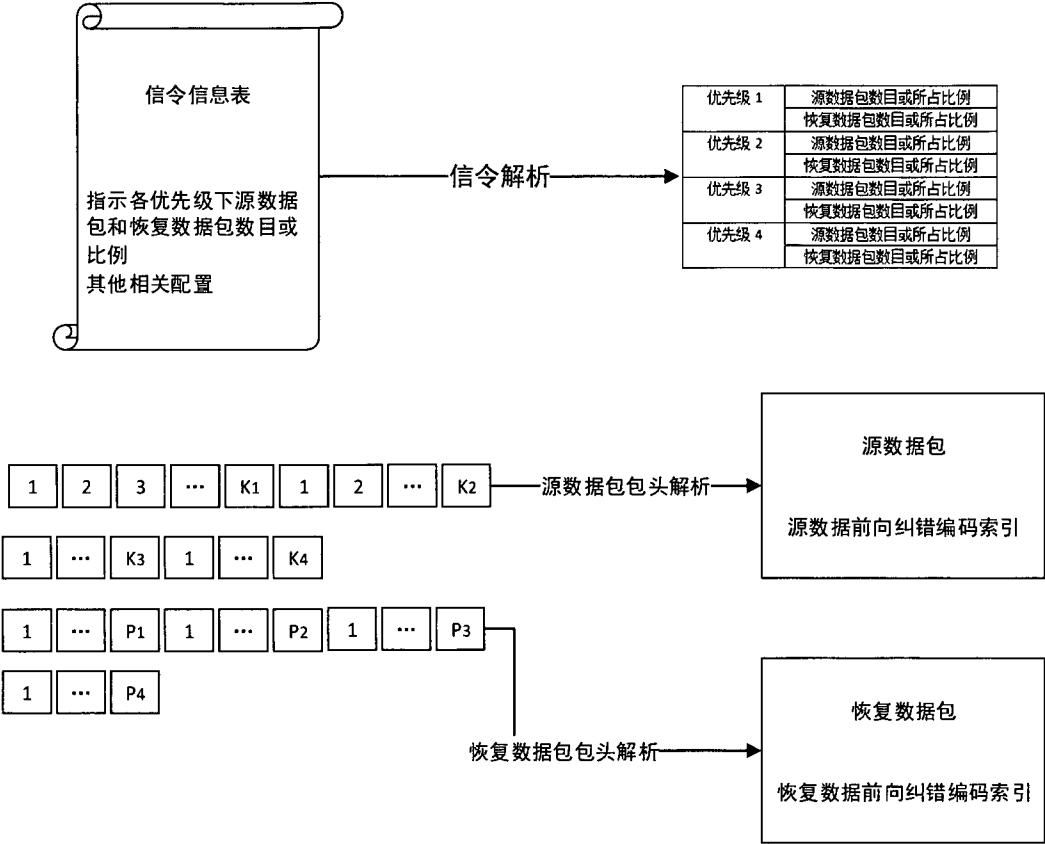


图 15

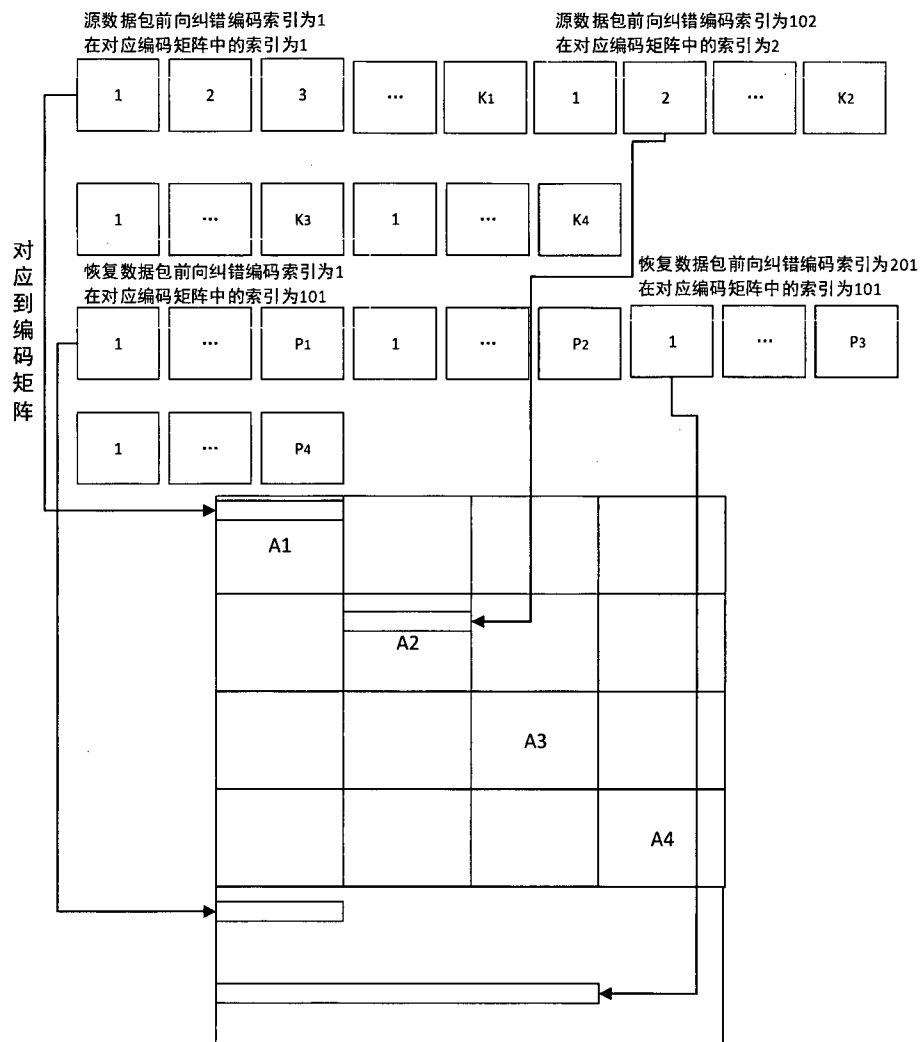


图 16

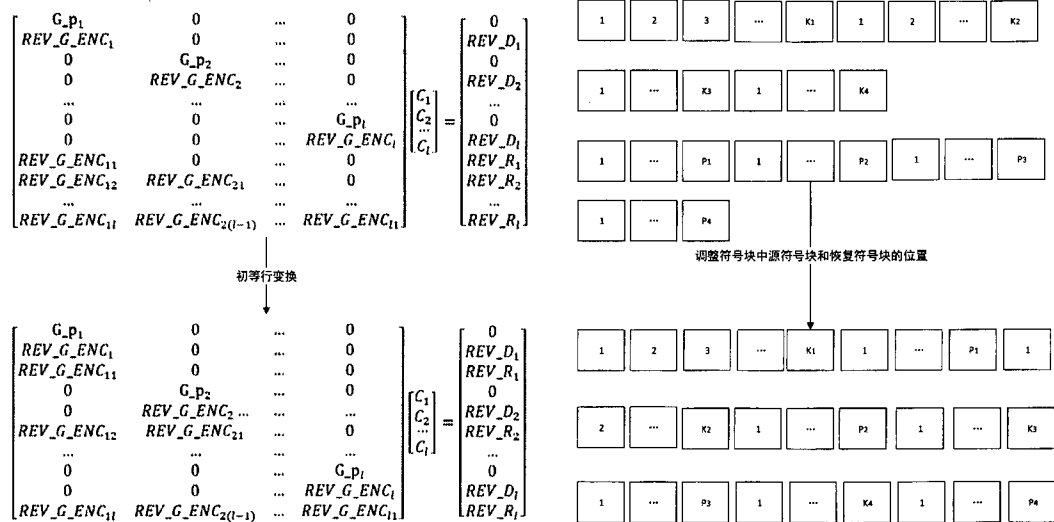


图 17

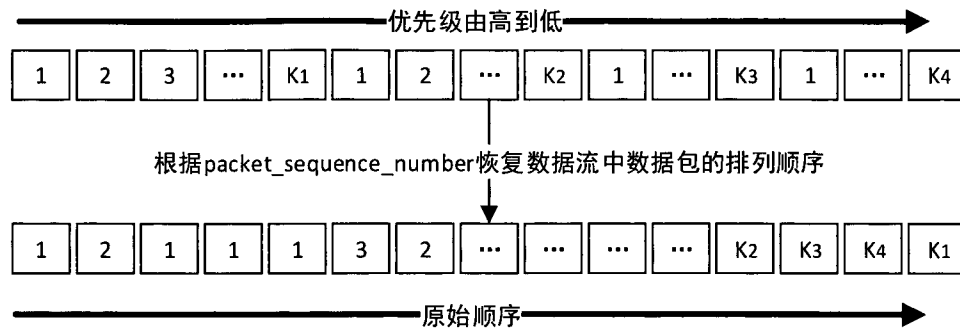


图 18

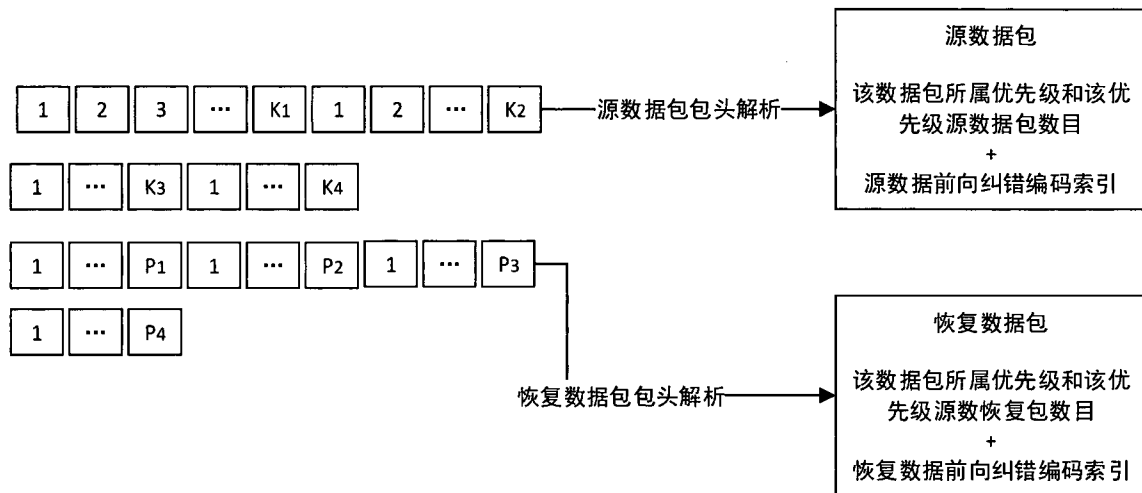


图 19

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2018/095051

A. CLASSIFICATION OF SUBJECT MATTER

H04L 1/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L; H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, CNKI, EPODOC, WPI, 3GPP: 前向纠错编码, 编码, 译码, 解码, 自适应, 优先, 中间码, 媒体, 非等错保护, 不等差错保护, FEC, coding, encode, decode, adaptive, priority, intermediat code, media, UEP, unequal error protection

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 106603192 A (SHANGHAI JIAO TONG UNIVERSITY) 26 April 2017 (2017-04-26) description, paragraphs 41-72	1-20
A	CN 103248897 A (HARBIN INSTITUTE OF TECHNOLOGY SHENZHEN GRADUATE SCHOOL) 14 August 2013 (2013-08-14) entire document	1-20
A	EP 1198100 A2 (AGERE SYSTEMS GUARDIAN CORPORATION) 17 April 2002 (2002-04-17) entire document	1-20
A	EP 2091205 A1 (NTT DOCOMO, INC.) 19 August 2009 (2009-08-19) entire document	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

30 August 2018

Date of mailing of the international search report

27 September 2018

Name and mailing address of the ISA/CN

State Intellectual Property Office of the P. R. China
No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing
100088
China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2018/095051

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	106603192	A	26 April 2017	CN	105827361	A	03 August 2016
				KR	20170102524	A	11 September 2017
				WO	2016110275	A1	14 July 2016
				CA	2998900	A1	14 July 2016
				CN	105991226	A	05 October 2016
				CN	106603193	A	26 April 2017
				JP	2018505597	A	22 February 2018
				US	2018069654	A1	08 March 2018
CN	103248897	A	14 August 2013	None			
EP	1198100	A2	17 April 2002	JP	2002185429	A	28 June 2002
				US	6684367	B1	27 January 2004
EP	2091205	A1	19 August 2009	None			

国际检索报告

国际申请号

PCT/CN2018/095051

A. 主题的分类

H04L 1/00(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

H04L; H04N

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNPAT, CNKI, EPODOC, WPI, 3GPP: 前向纠错编码, 编码, 译码, 解码, 自适应, 优先, 中间码, 媒体, 非等错保护, 不等差错保护, FEC, coding, encode, decode, adaptive, priority, intermediat code, media, UEP, unequal error protection

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
A	CN 106603192 A (上海交通大学) 2017年 4月 26日 (2017 - 04 - 26) 说明书第41-72段	1-20
A	CN 103248897 A (哈尔滨工业大学深圳研究生院) 2013年 8月 14日 (2013 - 08 - 14) 全文	1-20
A	EP 1198100 A2 (AGERE SYSTEMS GUARDIAN CORPORATION) 2002年 4月 17日 (2002 - 04 - 17) 全文	1-20
A	EP 2091205 A1 (NTT DOCOMO, INC.) 2009年 8月 19日 (2009 - 08 - 19) 全文	1-20

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2018年 8月 30日

国际检索报告邮寄日期

2018年 9月 27日

ISA/CN的名称和邮寄地址

中华人民共和国国家知识产权局(ISA/CN)
中国北京市海淀区蓟门桥西土城路6号 100088

传真号 (86-10)62019451

受权官员

田涛

电话号码 86-(10)-53961637

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2018/095051

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	106603192	A	2017年 4月 26日	CN	105827361	A	2016年 8月 3日
				KR	20170102524	A	2017年 9月 11日
				WO	2016110275	A1	2016年 7月 14日
				CA	2998900	A1	2016年 7月 14日
				CN	105991226	A	2016年 10月 5日
				CN	106603193	A	2017年 4月 26日
				JP	2018505597	A	2018年 2月 22日
				US	2018069654	A1	2018年 3月 8日
CN	103248897	A	2013年 8月 14日	无			
EP	1198100	A2	2002年 4月 17日	JP	2002185429	A	2002年 6月 28日
				US	6684367	B1	2004年 1月 27日
EP	2091205	A1	2009年 8月 19日	无			

表 PCT/ISA/210 (同族专利附件) (2015年1月)