

# 用于 GPRS 无线传输的抗分组丢失算法

张灯灿, 林春深, 万晋, 李学来

(福州大学化学化工学院, 福州 350108)

**摘要:** 针对 GPRS 网络中 IP 信道的干扰问题, 提出一种抗分组丢失算法。该算法将传输数据分段进行 RS 纠删编码, 并采用交织技术对数据进行最优分组并传输, 在纠删能力不变的同时获得近似线性的编解码时间。仿真结果表明, 与原有 RS 编码方式相比, 该算法可以更好地满足大数据编解码对实时性的要求。

**关键词:** 抗分组丢失; 纠删码; 交织

## Anti-packeting Loss Algorithm Used in GPRS Wireless Transmission

ZHANG Deng-can, LIN Chun-shen, WAN Jin, LI Xue-lai

(College of Chemistry and Chemical Engineering, Fuzhou University, Fuzhou 350108)

**【Abstract】** Aiming at the disturbance of the IP channel in GPRS network, this paper proposes a packet loss resilience encoding algorithm. The algorithm makes data transmission cutted into several sections and encoded by section based on RS erasure code, uses interleaving technology to process optimal grouping and transmission for the data, and acquires approximate linear encoding and decoding time without weakening the ability of erasure correction. Simulation results show that the algorithm can better satisfy the need of real-time for mass data coding than existing RS coding style.

**【Key words】** anti-packeting loss; erasure code; interleaving

### 1 概述

通用分组无线业务(GPRS)是在现有 GSM 系统上发展出来的一种新承载业务, 目的是为用户提供分组形式的电信业务。作为承载网络, GPRS 系统本身采用 IP 网络结构, 实现与 Internet 的互联。基于 GPRS 的无线数据传输技术, 具有数据传输不受地域限制、建设成本低、数据传输速率高以及永远在线等特点, 它在远程数据采集如远程抄表、远程监控等工业领域得到广泛应用<sup>[1-2]</sup>。

#### 1.1 用于 GPRS 的差错控制技术

在实际生产环境中, 基于无线 GPRS 网络的 IP 信道存在很多干扰, 在网络传输的过程中, 分组丢失是不可避免的。因此, 必须采用一定的差错控制技术以防止分组数据的丢失以及提高数据传输的效率。在现有基于 TCP/IP 协议的互联网中, 一般采用自动重传请求(Auto Repeat Request, ARQ)技术进行差错控制, 重传丢失的数据。由于 ARQ 技术是以吞吐率为代价提高可靠性的, 因此将 ARQ 技术用于 GPRS 无线传输容易发生网络过载和阻塞现象, 严重时会造成网络瘫痪, 而且 ARQ 技术并不适用于实时性要求较高或信道丢包率较高的应用场合<sup>[3]</sup>。

在 TCP/IP 体系结构中, 数据在 IP 层传输时, 一旦发生数据错误, 该数据分组即被丢弃, 因此, IP 信道实际上是纠删信道。采用基于纠删码的前向纠错(Forward Error Correction, FEC)技术是目前解决 IP 网络分组丢失问题的一种比较可行的方法<sup>[4]</sup>。在 GPRS 网络中, 设计一种纠删能力和实时性都较好的纠删编码方法是很必要的。

#### 1.2 纠删码现状

在基于 IP 的应用中, 纠删码以分组为单位进行前向纠

错, 其原理可简单描述为: 发送端对  $k$  个源数据分组进行编码, 得到  $n-k$  个校验分组, 将这  $n$  个分组在 IP 网络上传输, 接收方只要接收到一定数量的编码分组就可以将原始信息恢复。目前, 广泛研究和应用的纠删码有 2 种: Reed-Solomon 码和 Tornado 码<sup>[5]</sup>。

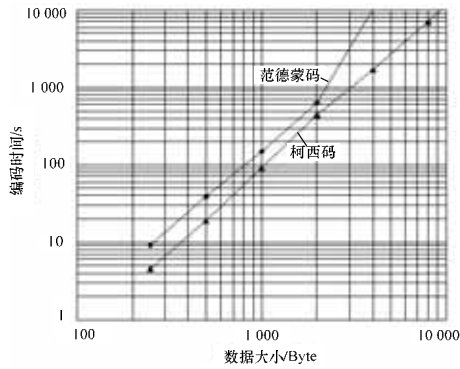
Reed-Solomon 码简称 RS 码, 它是现有技术中应用最为广泛的一类纠删码。RS 码继承了 BCH 码抗随机误码的能力, 同时又具有抗突发误码的能力。RS 码根据其生成矩阵的不同, 分为范德蒙码和柯西码<sup>[5-6]</sup>。RS 码是一种极大最小距离可分码(Maximum Distance Code, MDS), 在理想情况下, RS 纠删码的接收端只要接收到  $n$  个分组中的  $k$  个就可以恢复全部  $k$  个源数据分组<sup>[6]</sup>。因此, 在理想条件下, RS 码的纠删性能和冗余量都达到最佳。

RS 码在实际应用中存在一些问题, 如不易构造、编解码算法复杂度高。文献[7]测试了范德蒙码和柯西码的编解码时间。通过对实验数据的分析, RS 码的编解码时间随数据长度的增加呈近似幂函数增长。因此, 当需要进行大数据量传输时, 会造成严重的时延。

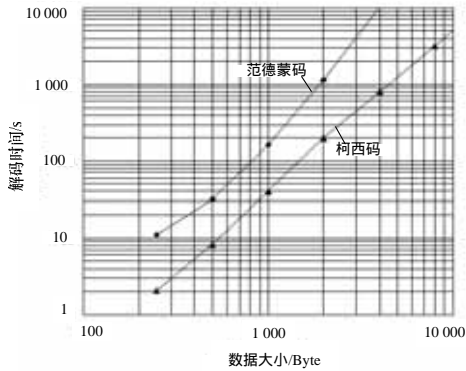
由于 IP 信道是纠删信道, 当分组中数据出现错误时, 整个数据分组即被丢弃, 因此当要传输大量数据时, 如果仅通过改变编码数据的长度来提高抗干扰性, 那么会使编解码时间急剧上升而导致大的时延, 此时, 就不能充分发挥 RS 码突出的纠删性能。RS 码的编码时间和解码时间如图 1 所示。

**作者简介:** 张灯灿(1983 -), 男, 硕士研究生, 主研方向: 无线数据采集技术; 林春深, 讲师、博士后; 万晋, 副教授; 李学来, 教授、博士生导师

**收稿日期:** 2009-05-17 **E-mail:** zdc@fzu.edu.cn



(a) 编码时间



(b) 解码时间

图 1 RS 码的编码时间和解码时间

Tornado 码是一类基于非正则二部图构造的级联型低密度纠错码，由于具有线性的编解码时间和较好的纠错性能，能大大缩短数据的编解码时间，因此得到广泛关注。Tornado 码的局限性在于：为了得到较快的编解码速度必须接收到略大于数据分组数  $k$  的分组数，且不能实现 100% 的译码成功。目前对 Tornado 码编码理论的研究尚不成熟，还有许多技术问题要解决，应用主要处于实验阶段。

已有学者对纠错码与交织技术相结合以提高抗连续丢包的能力进行研究<sup>[8]</sup>。但是单纯地将两者进行结合，当传输数据量变大时，编解码时间会急剧上升，造成严重时延。

本文针对上述问题，提出一种结合 RS 纠错码与交织技术，保证纠错性能的同时缩短大容量数据编解码时间。

## 2 抗分组丢失算法

本算法的基本思想是：在编码效率  $\eta$  不变的前提下，将长数据分成若干短数据，分别进行 RS 编码，以获得随数据长度呈线性增长的编解码时间，但这样会减弱其抗突发干扰能力，因此，采用交织技术将行码序列变换为并码传输，使信道传输过程中产生的突发错误离散化，以达到原有 RS 纠错码的纠错性能。算法由编码算法和解码算法构成，数据的发送与接收过程如图 2、图 3 所示。

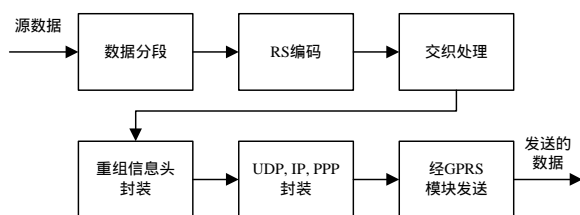


图 2 数据发送过程

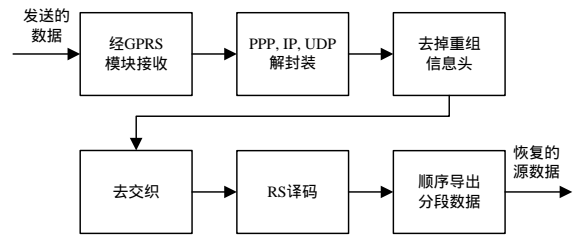


图 3 数据接收过程

### 2.1 RS——交织编码

(1) 对于一段  $x$  字节的数据流，将其分为  $m$  段，每段  $k$  个字节，建立数组，记为  $A$ ：

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mk} \end{bmatrix}$$

(2) 对各段数据  $A1=[a_{11}, a_{12}, \dots, a_{1k}]$ ,  $A2=[a_{21}, a_{22}, \dots, a_{2k}]$ , ...,  $A_m=[a_{m1}, a_{m2}, \dots, a_{mk}]$  分别进行 RS 编码<sup>[9-10]</sup>，生成  $(n-k)$  Byte 的校验码，设每段的编码时间为  $t_1$ ，得到编码后的数据：

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2k} & a_{2,k+1} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mk} & a_{m,k+1} & \cdots & a_{m,n} \end{bmatrix}$$

(3) 为了不失纠错能力，在接收端对丢失的分组进行恢复，具有  $(n-k)$  个校验字节的  $n$  Byte 数据应分成的分组数  $y$   $x \times n / (n-k) (x=1, 2, \dots)$ ，当  $x \times n / (n-k) < (x+1) \times n / (n-k)$  时，允许丢包数为  $r=x$ ，因此，分组数的最佳取值  $p$  为大于  $x \times n / (n-k)$  的最小整数，记为  $p=\lceil x \times n / (n-k) \rceil$ ，其中， $\lceil \cdot \rceil$  为向上取整，如  $\lceil 3 \times 31 / 4 \rceil = 24$ 。

将  $n$  Byte 数据分成  $p$  分组， $n$  Byte 数据在每分组所占最大字节数  $q=\lceil n/p \rceil$ ，不足补 0。经过交织处理后，可得到待发送数据如下：

$$\begin{bmatrix} a_{11} & a_{1,p+1} & \cdots & a_{1,p-3} & a_{21} & a_{2,p+1} & \cdots & a_{2,p-3} & \cdots & a_{m1} & a_{m,p+1} & \cdots & a_{m,p-3} \\ a_{12} & a_{1,p+2} & \cdots & a_{1,p-4} & a_{22} & a_{2,p+2} & \cdots & a_{2,p-4} & \cdots & a_{m2} & a_{m,p+2} & \cdots & a_{m,p-4} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{1,p-2} & a_{1,2p-2} & \cdots & a_{1n} & a_{2,p-2} & a_{2,2p-2} & \cdots & a_{2n} & \cdots & a_{m,p-2} & a_{m,2p-2} & \cdots & a_{mn} \\ a_{1,p-1} & a_{1,2p-1} & \cdots & 0 & a_{2,p-1} & a_{2,2p-1} & \cdots & 0 & \cdots & a_{m,p-1} & a_{m,2p-1} & \cdots & 0 \\ a_{1,p} & a_{1,2p} & \cdots & 0 & a_{2,p} & a_{2,2p} & \cdots & 0 & \cdots & a_{m,p} & a_{m,2p} & \cdots & 0 \end{bmatrix}$$

(4) 在每一分组的前面加 4 Byte 的重组信息头，首字节为数据编号  $gi$ ，第 2 个字节为分组总数，值即为  $p$ ，第 3 个字节为允许丢包数  $r$ ，第 4 个字节为分组序号，最终的待发送数据格式为

$$\begin{bmatrix} gi & p & r & 1 & a_{11} & a_{1,p+1} & \cdots & a_{1,p-3} & a_{21} & a_{2,p+1} & \cdots & a_{2,p-3} & \cdots & a_{m1} & a_{m,p+1} & \cdots & a_{m,p-3} \\ gi & p & r & 2 & a_{12} & a_{1,p+2} & \cdots & a_{1,p-4} & a_{22} & a_{2,p+2} & \cdots & a_{2,p-4} & \cdots & a_{m2} & a_{m,p+2} & \cdots & a_{m,p-4} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ gi & p & r & p-2 & a_{1,p-2} & a_{1,2p-2} & \cdots & a_{1n} & a_{2,p-2} & a_{2,2p-2} & \cdots & a_{2n} & \cdots & a_{m,p-2} & a_{m,2p-2} & \cdots & a_{mn} \\ gi & p & r & p-1 & a_{1,p-1} & a_{1,2p-1} & \cdots & 0 & a_{2,p-1} & a_{2,2p-1} & \cdots & 0 & \cdots & a_{m,p-1} & a_{m,2p-1} & \cdots & 0 \\ gi & p & r & p & a_{1,p} & a_{1,2p} & \cdots & 0 & a_{2,p} & a_{2,2p} & \cdots & 0 & \cdots & a_{m,p} & a_{m,2p} & \cdots & 0 \end{bmatrix}$$

(5) 对每一行数据  $[gi, p, r, 1, a_{11}, a_{1,p+1}, \dots, a_{m,n-p+3}]$ ,  $[gi, p, r, 2, a_{12}, a_{1,p+2}, \dots, a_{m,n-p+4}]$ , ...,  $[gi, p, r, p, a_{1,p}, a_{1,2p}, \dots, a_{m,0}]$  分别依次进行 UDP 封装和 IP 封装，最终封装成 PPP 数据帧，然后在 GPRS 无线数据链路层上进行传输。

### 2.2 解码算法研究

在接收端预先设定  $k, n$  的值，并在 TCP/IP 协议的应用层

设置一个计时器、一个计数器和一个接收数组并赋初值。当接收到数据编号为  $g_i$  的第 1 个分组数据时, 计时器开始计时, 每收到一个编号为  $g_i$  的分组数据时计数器加 1, 并根据分组序号将分组数据放入接收数组的相应行内。

(1) 当计数器的值等于  $p$  时, 计时器和计数器停止工作, 对接收数组中的数据去掉重组信息头, 然后进行交织译码。对于 IP 信道, 检错工作由 TCP/IP 体系结构的数据链路层、网际层和运输层完成, 发现错误即丢弃。因此, 接收到分组即认为是无错的, 无须经过 RS 译码校验, 直接去掉校验码, 还原成发送端发送的数据。

(2) 当计时器计时时间达到  $p$ , 计数器的值未达到  $p$  时, 校验计数器  $p-r$  是否成立, 若成立, 则对接收数组中的数据去掉重组信息头, 得接收数据:

$$\begin{bmatrix} a_{11} & a_{1,p+1} & \cdots & a_{1,n-p+3} & a_{21} & a_{2,p+1} & \cdots & a_{2,n-p+3} & \cdots & a_{m1} & a_{m,p+1} & \cdots & a_{m,n-p+3} \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots & & \vdots \\ a_{1,p-2} & a_{1,p-2} & \cdots & a_{1n} & a_{2,p-2} & a_{2,p-2} & \cdots & a_{2n} & \cdots & a_{m,p-2} & a_{m,p-2} & \cdots & a_{mn} \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ a_{1,p} & a_{1,p} & \cdots & 0 & a_{2,p} & a_{2,p} & \cdots & 0 & \cdots & a_{m,p} & a_{m,p} & \cdots & 0 \end{bmatrix}$$

对上述接收数据进行解交织处理后得到:

$$\begin{bmatrix} a_{11} & \cdots & a_{1,p-2} & 0 & a_{1,p} & \cdots & a_{1,2p-2} & 0 & a_{1,2p} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2,p-2} & 0 & a_{2,p} & \cdots & a_{2,2p-2} & 0 & a_{2,2p} & \cdots & a_{2n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & \cdots & a_{m,p-2} & 0 & a_{m,p} & \cdots & a_{m,2p-2} & 0 & a_{m,2p} & \cdots & a_{mn} \end{bmatrix}$$

由于丢失分组的分组序号是知道的, 因此在每 1 行  $n$  Byte 的数据中, 丢失数据所处的位置是可知的。且由于实际丢包数小于允许丢包数, 因此可调用 RS 纠错译码程序对各行分别进行纠错译码, 设每行的译码时间为  $t_2$ , 得到还原数组:

$$\hat{A} = \begin{bmatrix} a_{11} & \cdots & a_{1,p-2} & \hat{a}_{1,p-1} & a_{1,p} & \cdots & a_{1,2p-2} & \hat{a}_{1,2p-1} & a_{1,2p} & \cdots & a_{1k} \\ a_{21} & \cdots & a_{2,p-2} & \hat{a}_{2,p-1} & a_{2,p} & \cdots & a_{2,2p-2} & \hat{a}_{2,2p-1} & a_{2,2p} & \cdots & a_{2k} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & \cdots & a_{m,p-2} & \hat{a}_{m,p-1} & a_{m,p} & \cdots & a_{m,2p-2} & \hat{a}_{m,2p-1} & a_{m,2p} & \cdots & a_{mk} \end{bmatrix}$$

按顺序导出分段数据, 还原成发送端发送的数据。

(3) 如果计时时间到, 而计时器  $p-r$  不成立, 那么丢失情况超出纠错范围, 无法恢复丢失的数据。因此, 对接收到的数据依次进行去掉重组信息头、交织译码、去掉校验码操作, 并往上层应用程序传输, 或进行其他相应处理。

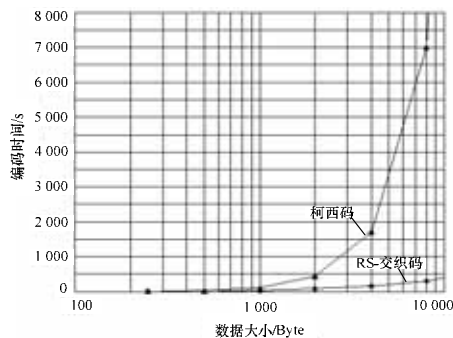
(4) 计时器、计数器清零, 接收数组重赋初值。

### 3 仿真实验分析

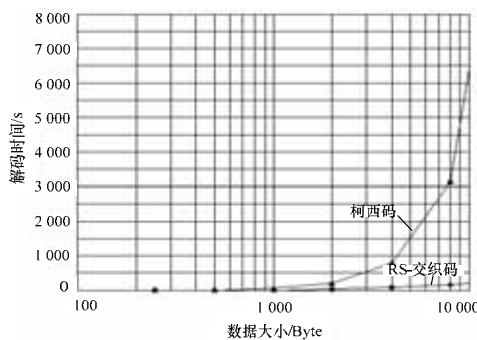
在基于 RS 纠错码的抗分组丢失算法中, 对于  $m \times k$  字长的原数据, 一般通过 RS 编码得到  $m \times (n-k)$  字长的校验码, 然后分成  $n$  个分组进行发送, 允许丢包数为  $(n-k)$ 。

在编码效率不变的情况下, 采用本算法编码, 将数据分成  $m$  段, 每段  $k$  Byte, 通过 RS 编码得到  $(n-k)$  字长的校验码, 交织处理时取  $x=(n-k)$ , 分组数  $p=[x \times n/(n-k)]=n$ , 允许丢包数为  $(n-k)$ 。2 种方法的允许丢包率均为  $=x/p=(n-k)/n$ , 且均允许在  $n$  个分组里连续丢失  $(n-k)$  个分组, 抗随机干扰和突发干扰能力不变。

本文通过仿真试验对本算法与原有基于柯西码的编解码时间进行测试, 结果如图 4 所示。测试结果显示, 当选定一定的 RS 编码格式  $(n, k)$ , 即  $n$  和  $k$  一定时, 随着数据量  $m$  的增大, 编解码时间呈近似线性增长, 而原有柯西编码则呈近似幂函数增长。可见, 本文提出的算法可大大缩短编解码的时间。



(a) 编码时间



(b) 解码时间

图 4 本算法的编码时间和解码时间

### 4 结束语

在不改变编码效率  $\eta$  的情况下, 通过分组数的合理选取, 本算法可达到原有 RS 纠错码的纠错性能。在突发性干扰严重时, 通过增大  $x$  的值可以增大交织编码的交织度, 进一步提高其抗连续丢包能力。结果证明本文算法在 RS 编码格式  $(n, k)$  选定时, 随着数据量  $m$  的增大, 编解码时间呈近似线性增长。

### 参考文献

- [1] 林梅金, 罗 飞, 李如雄. GPRS 网络的远程监控系统[J]. 计算机工程, 2006, 32(3): 240-242.
- [2] 王 琳. 基于 GPRS 的无线图像数据传输[J]. 计算机工程, 2008, 34(13): 232-233.
- [3] 李钟华, 李伟华. 大型数据的编码传输技术研究[J]. 计算机工程, 2006, 32(19): 120-122.
- [4] 艾 达, 常义林, 罗 忠, 等. 一种新的 IP 网络视频通信丢包错误纠正方案[J]. 电子与信息学报, 2006, 28(10): 1879-1882.
- [5] 慕建军, 路成业, 王新梅. 关于纠错码的研究与进展[J]. 电子与信息学报, 2002, 24(9): 1276-1280.
- [6] Rizzo L. Effective Erasure Codes for Reliable Computer Communication Protocols[J]. ACM Computer Communication Review, 1997, 27(2): 24-36.
- [7] Luby M. Benchmark Comparisons of Erasure Codes[Z]. [2008-12-03]. <http://www.icsi.berkeley.edu/~luby/erasure.html>.
- [8] 杨宗凯, 彭 杰, 余 江. 实时视频通信中的自适应前向纠错方案设计[J]. 计算机工程与科学, 2007, 29(8): 43-45.
- [9] 黄宏博, 肖峻岭, 佟俐鹃. 基于 Reed-Solomon 算法的 QR 码纠错编码[J]. 计算机工程, 2003, 29(1): 93-95.
- [10] 程念科, 胡瑞敏, 王中元. 基于 RS 编码的视频差错恢复方法[J]. 计算机应用研究, 2006, 23(4): 65-67.

编辑 陆燕菲