



中华人民共和国国家标准

GB/T 43423—2023

空间数据与信息传输系统 深空光通信编码与同步

Space data and information transfer systems—
Deep space optical communications coding and synchronization

2023-11-27 发布

2024-03-01 实施

国家市场监督管理总局
国家标准化管理委员会 发布

目 次

前言	III
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	1
5 总则	2
5.1 体系结构	2
5.2 功能	3
5.3 下行数据编码与同步子层内部组织	3
5.4 上行信标和可选伴随 AOS/USLP 传送帧的编码与同步子层内部组织	5
5.5 位序号的约定	6
6 下行数据的编码与同步	6
6.1 信号说明	6
6.2 传送帧	7
6.3 附加同步标识	7
6.4 分块器	8
6.5 伪随机发生器	9
6.6 添加循环冗余校验(CRC)	9
6.7 添加终止二进制位	11
6.8 SCPPM 编码器	11
6.9 信道交织器	14
6.10 码同步标识	15
6.11 重复	16
6.12 时隙映射器	16
6.13 插入保护时隙	16
6.14 传送帧有效性确认	17
6.15 序列指示	17
7 上行信标和可选伴随数据传送帧的编码与同步	17
7.1 信号说明	17
7.2 信标信号	17
7.3 编码与同步子层输入	18
7.4 附加同步标识	18
7.5 分块器	18

7.6	LDPC 编码器	19
7.7	伪随机发生器	19
7.8	信道交织器	19
7.9	码字同步标识	19
7.10	PN 扩频器	20
7.11	2-PPM 时隙映射器	20
7.12	插入保护时隙	20
7.13	AOS/USLP 传送帧有效性确认和序列指示	21
8	管理参数	21
8.1	管理参数说明	21
8.2	下行数据管理参数	21
8.3	上行数据管理参数	21
附录 A (规范性)	服务定义	22
参考文献	24



前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第 1 部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由全国宇航技术及其应用标准化技术委员会(SAC/TC 425)提出并归口。

本文件起草单位：北京空间飞行器总体设计部、北京卫星信息工程研究所、西安空间无线电技术研究所、南京大学、山东瑞鑫地理信息工程有限公司、中国航天标准化研究所、天津航天中为数据系统科技有限公司、杭州润州光电技术有限公司。

本文件主要起草人：白帆、袁瑞佳、苏昱玮、董涛、何熊文、金世超、张建华、谢天娇、韩宇、李炯卉、殷杰、刘志慧、邸航、强晖萍、赵康健、周玉霞、李明、邓守峰、刘继光、徐灶良。



空间数据与信息传输系统

深空光通信编码与同步

1 范围

本文件描述了空间数据与信息传输领域深空任务背景下空间对地光通信中数据链路层编码与同步子层的功能、下行数据的编码与同步、上行信标和可选伴随数据传送的编码与同步以及管理参数。

本文件适用于深空激光通信领域通过大气信道传输的空对地和地对空场景的高光子效率链路,其他应用参考使用。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中,注日期的引用文件,仅该日期对应的版本适用于本文件;不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 42041—2022 航天术语 空间数据与信息传输

GB/T 43372—2023 空间数据与信息传输系统 深空光通信物理层

3 术语和定义

GB/T 42041—2022 界定的以及下列术语和定义适用于本文件。

3.1

编码码率 code rate

编码器输入信息符号的平均数与相应输出符号平均数之比。

3.2

码字 codeword

编码后输出、在译码时作为一个整体处理的数据序列。

[来源:GB/T 42041—2022,3.3.13]

3.3

卷积码 convolution code

在信息码序列中加入的多余码元(校验元)与当前的若干信息码元以及以前的若干信息码元存在某种函数关系的一种纠错码。

[来源:GB/T 42041—2022,3.3.16]

3.4

同步标识传送帧 synchronization-marked transfer frame;SMTF

由附加同步标识(ASM)和传送帧组成的数据单元。

4 缩略语

下列缩略语适用于本文件。

- AOS:高级在轨系统(Advanced Orbiting Systems)
- ASM:附加同步标识(Attached Synchronization Marker)
- CRC:循环冗余校验码(Cyclic Redundancy Check)
- CSM:码同步标识(Code word Synchronization Marker)
- FECF:帧差错控制域(Frame Error Control Field)
- LDPC:低密度奇偶校验码(Low-Density Parity-Check)
- MSB:最高有效位(Most Significant Bit)
- OSI:开放系统互连(Open System Interconnection)
- OTM:光遥测(Optical Telemetry)
- PN:伪随机噪声(Pseudo-random Noise)
- PPM:脉冲位置调制(Pulse Position Modulation)
- SCPPM:串行级联卷积码脉冲位置调制(Serially concatenated Convolutionally coded Pulse Position Modulation)
- SMCW:同步标识码字(Synchronization-Marked Code Word)
- SMTF:同步标识传送帧(Synchronization-Marked Transfer Frame)
- TM:遥测(Telemetry)
- USLP:统一空间数据链路协议(Unified Space data Link Protocol)

5 总则

5.1 体系结构

图 1 描述了本文件与 OSI 分层的关系。空间数据链路协议把数据链路层分为同步与信道编码子层和数据链路子层协议,其中,数据链路协议子层提供传送帧的生成功能。



图 1 本文件与 OSI 分层的关系

深空光通信编码与同步子层所支持的空间数据链路协议包括 TM 空间数据链路协议(见 GB/T 39351—2020)、AOS 空间数据链路协议(见 GB/T 39345—2020)和 USLP(见 GB/T 42040—2022)。光通信编码与同步对固定长度的传送帧提供以恒定的帧速率在空间链路的物理信道上的传输服务,服务以原语的形式提供,具体要求按附录 A 执行。

本文件规定的深空光通信编码与同步协议为传送帧在深空链路上传送提供同步与信道编码功

能,包括信道编码、同步和传送帧有效性确认。

GB/T 43372—2023 规定了传送帧(包括下行数据、上行信标与可选伴随数据)的物理特征。在实际应用中,地面站将信标和可选数据发射至航天器,航天器将下行数据信号发射到地面站。

5.2 功能

深空光通信系统架构如图 2 所示。在整个通信会话过程中,地面站终端 A 发送信标以及可选伴随 AOS 传送帧或 USLP 传送帧数据。星上终端 B 接收机锁定信标,并利用其辅助光通信终端发射机精准瞄准。同时,上行可选伴随 AOS 传送帧或 USLP 传送帧被星上终端 B 在轨译码。星上终端 B 发射 OTM 传送帧、AOS 传送帧或固定长度 USLP 传送帧,地面站终端 A 接收 OTM 传送帧、AOS 传送帧或固定长度 USLP 传送帧。

深空光通信编码与同步子层用于传输深空链路上传送帧,提供以下功能:

- a) 信道编码;
- b) 同步;
- c) 传送帧有效性确认。

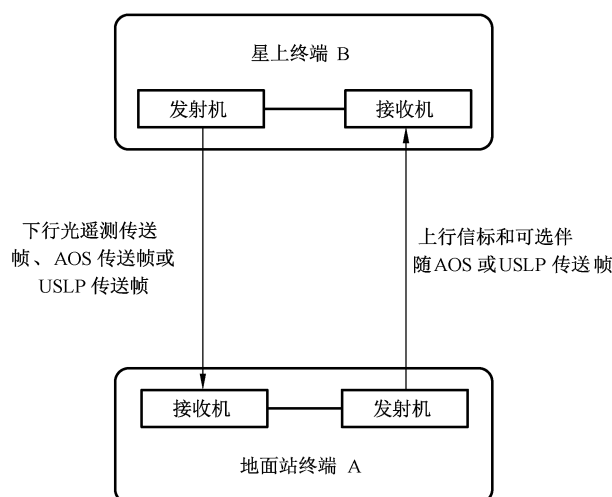


图 2 深空光通信系统架构

星上终端 B 向地面站终端 A 发射的下行数据(包括空间对地面的下行链路 OTM 传送帧、AOS 传送帧或固定长度 USLP 传送帧)的编码与同步特性见第 6 章,地面站终端 A 向星上终端 B 发送上行信标和可选伴随 AOS 传送帧或 USLP 传送帧的编码与同步特性见第 7 章。

注:本文件未规定链路的方向或几何形状。在每次通信会话期间,上述信号传送可同时在链路的两端持续发生。

5.3 下行数据编码与同步子层内部组织

5.3.1 发送端传送帧内部组织

图 3 为下行发送端编码与同步子层内部组织流程图,主要描述了各功能之间的逻辑关系。在发送端,编码与同步子层接收来自数据链路协议子层长度固定且速率恒定的传送帧,根据任务选择执行功能,向物理层提供二进制向量,以指定包含光脉冲的时隙。

注:图 3 所示内容并不表示实际系统中硬件和软件的配置。

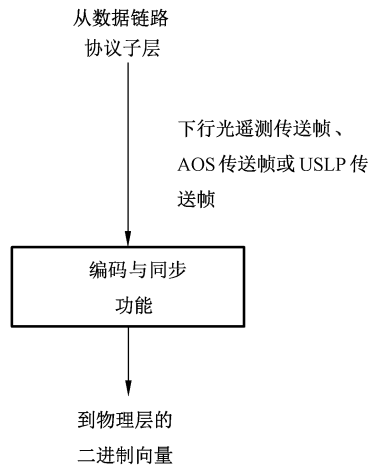


图 3 下行发送端编码与同步子层内部组织流程图

5.3.2 接收端传送帧内部组织

5.3.2.1 下行接收端流程说明

图 4 为下行接收端编码与同步子层内部组织流程图,主要描述了子层所执行的功能以及功能之间的逻辑关系。在接收端,编码与同步子层接收物理层输出,运行任务所选功能。终端接收机输出是指已接收传输每个时隙的时隙测量结果,包括终端接收机对光信号强度、观测到的光子数量或相关统计的评估。码字同步与 SCPPM 解码是其中的一项功能,此功能可恢复 SMTF。SMTF 中所出现的同步标识可以同步、恢复每个传送帧,传送帧与质量指示和序列指示一起发送至数据链路协议子层。

注:图 4 所示内容并不表示实际系统中硬件和软件的配置。深空光通信与射频通信的协议标准的分层有所不同,深空光通信将信号的解调制功能定义在编码与同步子层之中,而非在物理层。

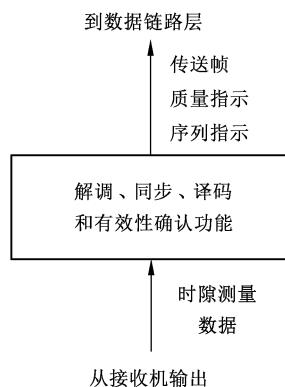


图 4 下行接收端编码与同步子层内部组织流程图

5.3.2.2 传送帧有效性确认

SCPPM 解码、传送帧恢复后,接收端的协议上层需确认恢复后的每个传送帧是否可作为有效的数据单元。进行传送帧有效性确认,同时生成质量指示。

SCPPM 解码器决定 SCPPM 每个码字是否解码正确。从正确解码的 SCPPM 码字中恢复的传送

帧标记为合格帧,从一个或多个解码不正确码字中恢复的传送帧标记为无效帧。

5.3.2.3 同步

使用 ASM 同步传送帧的方法,见 6.3。

5.4 上行信标和可选伴随 AOS/USLP 传送帧的编码与同步子层内部组织

5.4.1 发送端传送帧内部组织

图 5 为上行发送端的内部组织图,主要描述了编码与同步子层所进行的功能以及功能之间的逻辑关系,在发送端,编码与同步子层接收来自数据链路协议子层的固定长度 AOS/USLP 传送帧。然后,编码与同步子层执行任务所选功能,生成 LDPC 编码的 SMCW。SMCW 是 PN 扩频映射到二进制脉冲位置调制(2-PPM)的符号。随后将两个保护间隙插入 2-PPM 符号流,发送到物理层。

注:图 5 所示内容不具体指向真实系统中的任何硬件或软件配置。

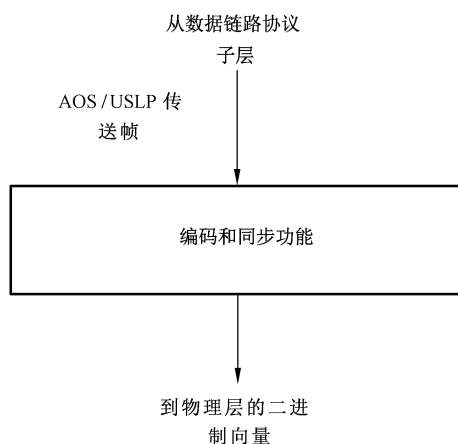


图 5 上行发送端内部组织图

5.4.2 接收端传送帧内部组织

5.4.2.1 上行接收端流程说明

图 6 为上行接收端内部组织图,该图展示了子层所进行的功能以及功能之间的逻辑关系,接收端编码与同步子层接收来自物理层接收器的输出,执行任务所选功能。这些功能包括 2-PPM、码字同步、LDPC 解码和 AOS 传送帧同步。AOS/USLP 传送帧恢复后,与质量指示和序列指示一起发送至数据链路协议子层。

注:图 6 所示内容不具体指向真实系统中的任何硬件或软件配置。

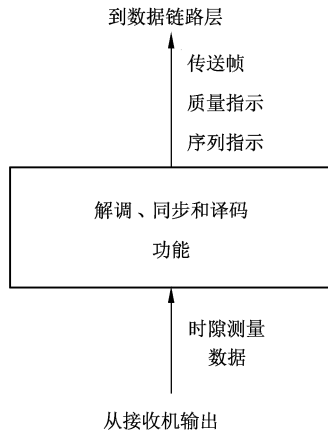


图 6 上行接收端内部组织图

5.4.2.2 同步

使用 ASM 同步传送帧的方法,见 6.3。

5.5 位序号的约定

本文件采用下述位序号的约定,以便识别 N bit 数据域中的各位。

对 N bit 数据的高低位遵守下述约定:第 1 个传送的 bit 称之为 bit 0,紧接的 bit 定义为 bit 1,这样依次到 bit $N-1$ 。当该数据域用作表示一个二进制值时(例如计数器),MSB 是该数据域最先传送位,也就是 bit 0,如图 7 所示。根据通信惯例,1 个字节为 8 bit。

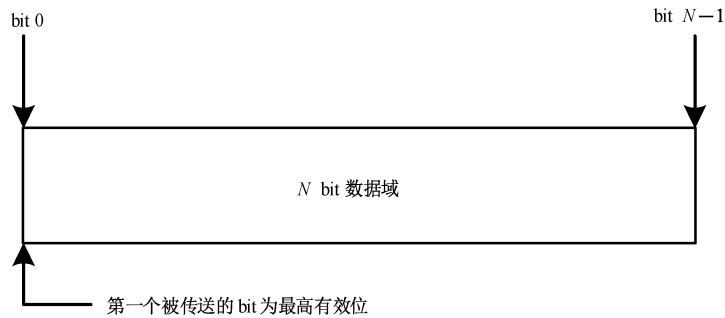


图 7 位序号的约定

6 下行数据的编码与同步

6.1 信号说明

下行以 TM、AOS/USLP 传送帧作为输入,产生一个指示脉冲时隙位置的二进制向量输出到物理层。

图 8 为下行数据传送功能流程图。该图标识出了该子层的功能及功能之间的逻辑关系。功能模块的选择取决于实际任务,并不是所有的功能都将出现在编码和同步子层中。

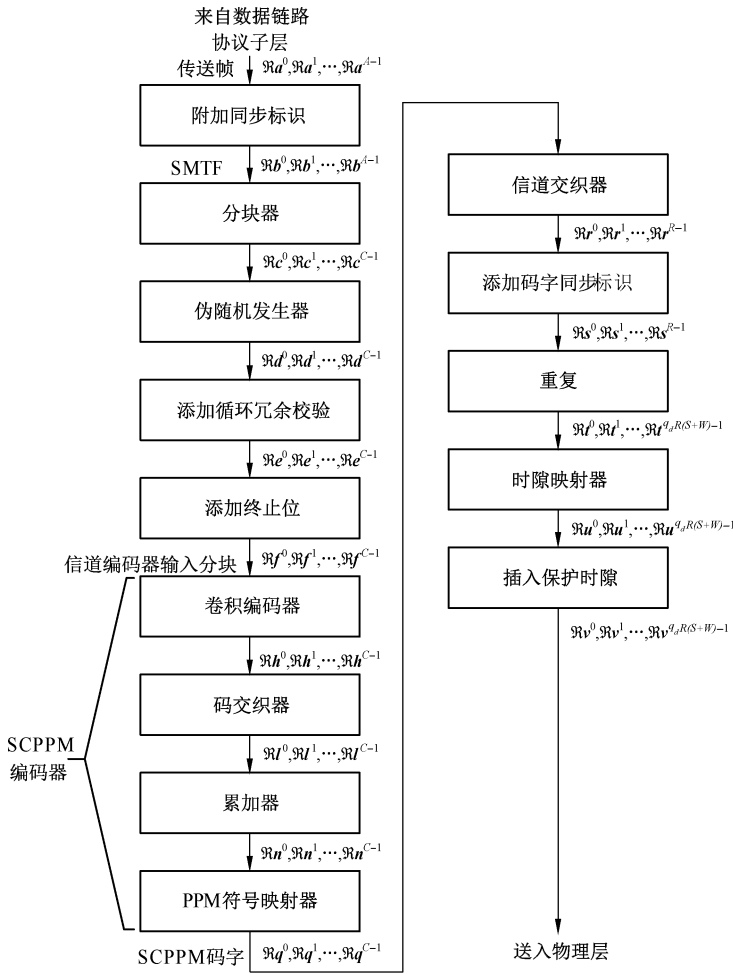


图 8 下行数据传送功能流程图

如图 8 所示，每个传送帧的前端都包含一个 ASM。6.3 中规定了 ASM 的生成与插入方式，由 ASM 和传送帧组成的数据单元称为 SMTF。SMTF 数据流被分割成信息块并做伪随机化。随后每个伪随机化后的信息块被添加上循环冗余校验和终止位，作为信道编码器的输入。由生成码字组成的 PPM 符号进行交织，在每个交织码字前附加一个 CSM。每个 PPM 符号被重复并映射成长度为 M 的二进制向量，并生成脉冲时隙向量。在每个 PPM 符号后插入保护时隙。

地面接收端需要两级同步，即码字同步（由 CSM 识别）和传送帧同步（由 ASM 识别）。通过识别符号流中特定的同步码来实现 CSM 同步，再通过进一步核验完成同步验证。

6.2 传送帧

编码和同步子层的输入为传送帧序列，其结构可用公式(1)表示：

$$\mathfrak{R}a = \mathfrak{R}a^0, \mathfrak{R}a^1, \dots, \mathfrak{R}a^i, \dots, \mathfrak{R}a^{A-1} \dots \dots \dots (1)$$

对于 $i \in \{0, 1, \dots, A-1\}$ ，第 i 个传送帧 $\mathfrak{R}a^i$ 用公式(2)表示：

$$\mathfrak{R}a^i = \mathfrak{R}a^i(0), \mathfrak{R}a^i(1), \dots, \mathfrak{R}a^i(j), \dots, \mathfrak{R}a^i(T_i - 1) \dots \dots \dots (2)$$

式中：
 $\mathfrak{R}a^i(j) \in \{0, 1\}$ —— 第 i 个帧的第 j bit；
 T_i —— 第 i 个帧的位数。

6.3 附加同步标识

在每个传送帧的前面加入一个 32 bit 的 ASM，形成一个 SMTF，其中 ASM 是固定的 32 bit 的同步

标识,序列为: $s = s_0, s_1, \dots, s_{31}$,用十六进制符号表示 1ACFFC1D。

对于 $i \in \{0, 1, \dots, A-1\}$,第 i 个 SMTF 的 \mathfrak{Rb}^i 用公式(3)和公式(4)表示:

$$\mathfrak{Rb}^i = \mathfrak{Rb}^i(0), \mathfrak{Rb}^i(1), \dots, \mathfrak{Rb}^i(j), \dots, \mathfrak{Rb}^i(B_i - 1) \dots\dots\dots (3)$$

其中 $B_i = T_i + 32$,且:

$$\mathfrak{Rb}^i(j) = \begin{cases} s_j, & 0 \leq j < 32 \\ \mathfrak{Ra}^i(j - 32), & 32 \leq j < B_i \end{cases} \dots\dots\dots (4)$$

附加 ASM 序列的 SMTF 结构如图 9 所示。

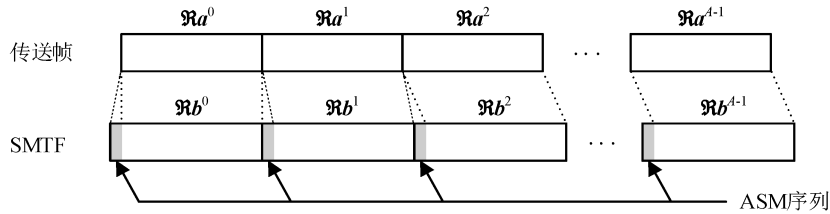


图 9 附加 ASM 序列的 SMTF 结构图

6.4 分块器

6.4.1 分块长度

SMTF 序列被分割成长度为 k 的信息块,其中 k 由编码码率 $r \in \{1/3, 1/2, 2/3\}$ 决定,见表 1, r 为管理参数。最后一个信息块的长度可能小于 k 。

表 1 信息块定义

编码码率 r	信息块大小 k	添加 CRC-32 和 2 bit 终止符后的信息块长度 \hat{k}
1/3	5 006	5 040
1/2	7 526	7 560
2/3	10 046	10 080

6.4.2 补零

分块器的输出应以最少数量的“0”进行补零,使其长度为 k 的倍数。通过分块和补零后分块器输出如图 10 所示。

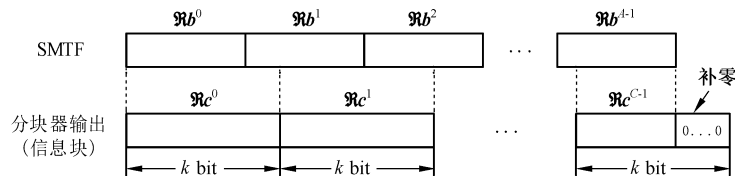


图 10 分块器输出示意图

SMTF 序列的 $\mathfrak{Rb}^0, \mathfrak{Rb}^1, \dots, \mathfrak{Rb}^i, \dots, \mathfrak{Rb}^{A-1}$ 可看作一个具有相同顺序的二进制一维向量,可用公式(5)表示:

$$\hat{\mathfrak{Rb}} = \hat{\mathfrak{Rb}}(0), \hat{\mathfrak{Rb}}(1), \dots, \hat{\mathfrak{Rb}}(j), \dots, \hat{\mathfrak{Rb}}(B - 1) \dots\dots\dots (5)$$

式中：

B —— \mathfrak{Rb}^i 包含的 bit 数的和。

序列 \mathfrak{Rb}^i 的末尾用最小数量的“0”填充,使其长度为 k 的整数倍。填充“0”后,序列 $\tilde{\mathfrak{Rb}}^i$ 可用公式(6)~公式(8)表示:

$$\begin{aligned} \tilde{\mathfrak{Rb}}^i &= \mathfrak{Rb}^i(0), \mathfrak{Rb}^i(1), \dots, \mathfrak{Rb}^i(j), \dots, \mathfrak{Rb}^i(B-1), \underbrace{0, 0, \dots, 0}_P \\ &= \tilde{\mathfrak{Rb}}^i(0), \tilde{\mathfrak{Rb}}^i(1), \dots, \tilde{\mathfrak{Rb}}^i(j), \dots, \tilde{\mathfrak{Rb}}^i(B+P-1) \end{aligned} \quad \dots\dots\dots(6)$$

$$\tilde{\mathfrak{Rb}}^i(j) = \begin{cases} \mathfrak{Rb}^i(j), & 0 \leq j < B \\ 0, & B \leq j < B+P \end{cases} \quad \dots\dots\dots(7)$$

$$P = \min\{p:k \mid B+p\} \quad \dots\dots\dots(8)$$

分块器将 $\tilde{\mathfrak{Rb}}^i$ 分为 $C=(B+P)/k$ 个分块并重新编号,每个分块长度为 k 的 \mathfrak{Rc}^i 可用公式(9)~公式(11)表示:

$$\mathfrak{Rc}^i = \mathfrak{Rc}^i(0), \mathfrak{Rc}^i(1), \dots, \mathfrak{Rc}^i(j), \dots, \mathfrak{Rc}^i(C-1) \quad \dots\dots\dots(9)$$

对于 $i \in \{0, 1, \dots, C-1\}$,第 i 个分块表示为:

$$\mathfrak{Rc}^i = \mathfrak{Rc}^i(0), \mathfrak{Rc}^i(1), \dots, \mathfrak{Rc}^i(j), \dots, \mathfrak{Rc}^i(k-1) \quad \dots\dots\dots(10)$$

对于 $j \in \{0, 1, \dots, k-1\}$,第 i 个分块第 j 个符号表示为:

$$\mathfrak{Rc}^i(j) = \tilde{\mathfrak{Rb}}^i(ki+j) \quad \dots\dots\dots(11)$$

6.5 伪随机发生器

实现伪随机化的方式是将分块器输出的每一个大小为 k bit 的信息块与标准的伪随机序列逐位进行异或运算,经过一个伪随机序列周期(255 bit)后继续重复循环,直到信息块结束为止,伪随机发生器数据流程图如图 11 所示。

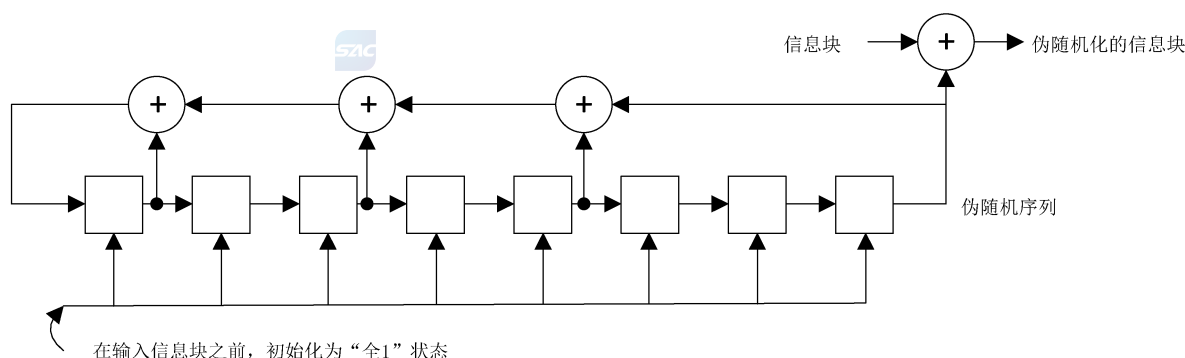


图 11 伪随机发生器数据流程图

对于 $i \in \{0, 1, \dots, C-1\}$,第 i 个伪随机信息块用公式(12)表示:

$$\mathfrak{Rd}^i = \mathfrak{Rd}^i(0), \mathfrak{Rd}^i(1), \dots, \mathfrak{Rd}^i(j), \dots, \mathfrak{Rd}^i(k-1) \quad \dots\dots\dots(12)$$

对于 $j \in \{0, 1, \dots, k-1\}$,第 i 个信息块第 j bit 用公式(13)表示:

$$\mathfrak{Rd}^i(j) = \mathfrak{Rc}^i(j) \oplus p(j) \quad \dots\dots\dots(13)$$

伪随机序列 $p(j)$ 的定义见 GB/T 39351—2020 中 11.4。

6.6 添加循环冗余校验(CRC)

每个伪随机化信息块的末尾需要添加 32 bit 的循环冗余校验位(32-CRC),如图 12 所示。

在输入信息块之前，初始化到“全1”状态

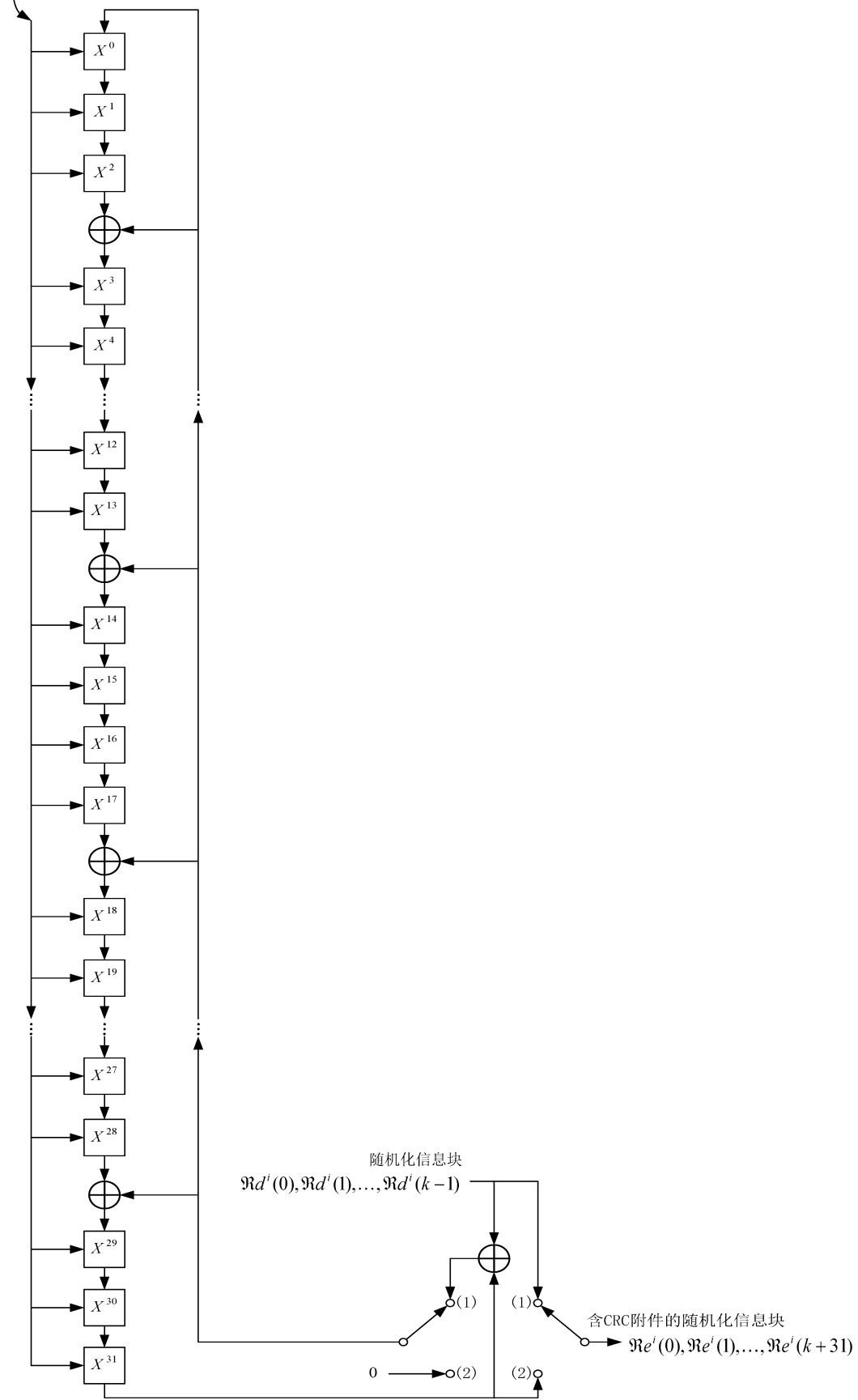


图 12 32-CRC 生成示意图

32-CRC 二进制位的计算为:

第 i 个添加了 32-CRC 的伪随机化信息块用公式(14)和公式(15)表示:

$$\mathfrak{Re}^i = \mathfrak{Re}^i(0), \mathfrak{Re}^i(1), \dots, \mathfrak{Re}^i(j), \dots, \mathfrak{Re}^i(k+31) \dots\dots\dots(14)$$

$$\mathfrak{Re}^i(j) = \begin{cases} \mathfrak{Rd}^i(j), & 0 \leq j < k \\ z_{j-k}^i, & k \leq j < k+32 \end{cases} \dots\dots\dots(15)$$

对于 $i \in \{0, 1, \dots, C-1\}$, 第 i 个伪随机化信息块 \mathfrak{Rd}^i 用 32 个“0”填充, 多项式用公式(16)表示:

$$\mathfrak{Rd}^i[X] = \mathfrak{Rd}^i(0)X^{k+31} + \mathfrak{Rd}^i(1)X^{k+30} + \dots + \mathfrak{Rd}^i(k-1)X^{32} \dots\dots\dots(16)$$

32-CRC 二进制位多项式用公式(17)表示:

$$z^i(X) = z_0^i X^{31} + z_1^i X^{30} + \dots + z_{30}^i X + z_{31}^i \dots\dots\dots(17)$$

并且由公式(18)和公式(19)计算得出:

$$z^i(X) = [\mathfrak{Rd}^i[X] + \sum_{j=0}^{31} X^{k+j}] \bmod h(X) \dots\dots\dots(18)$$

其中:

$$h(X) = X^{32} + X^{29} + X^{18} + X^{14} + X^3 + 1 \dots\dots\dots(19)$$

$\sum_{j=0}^{31} X^{k+j}$ 项的移位寄存器初始化为“全 1”状态。

6.7 添加终止二进制位

在每个带有 32-CRC 的伪随机化信息块后端添加 2 个“0”, 产生一组长度为 $\hat{k} = k + 34$ bit 的信息块, 对于 $i \in \{0, 1, \dots, C-1\}$, 输入 SCPPM 编码器的第 i 个信息块用公式(20)和公式(21)表示:

$$\mathfrak{Rf}^i = \mathfrak{Rf}^i(0), \mathfrak{Rf}^i(1), \dots, \mathfrak{Rf}^i(j), \dots, \mathfrak{Rf}^i(\hat{k}-1) \dots\dots\dots(20)$$

其中:

$$\mathfrak{Rf}^i(j) = \begin{cases} \mathfrak{Re}^i(j), & 0 \leq j < \hat{k} - 2 \\ 0, & \hat{k} - 2 \leq j < \hat{k} \end{cases} \dots\dots\dots(21)$$

这些信息块被称为 SCPPM 编码器输入信息块。带有 32-CRC 和二进制终止位的伪随机化信息块如图 13 所示。

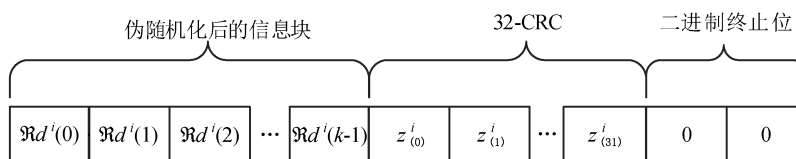


图 13 SCPPM 编码器输入信息块

6.8 SCPPM 编码器

6.8.1 SCPPM 编码器的结构

SCPPM 编码器的结构如图 14 所示。每个 SCPPM 编码器输入信息块的长度为 $\hat{k} = 15\ 120r$ (见表 1), 外码编码器产生 15 120 个二进制卷积编码符号, 编码符号经交织、累加后映射为 PPM 符号。



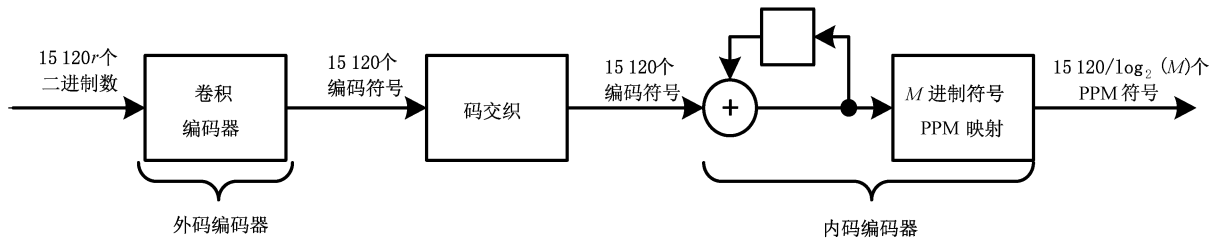


图 14 SCPPM 编码器结构示意图

6.8.2 卷积编码器

6.8.2.1 生成多项式

SCPPM 外码是一个约束长度为 3 的卷积码，其生成多项式可用公式(22)表示：

$$\begin{aligned}
 g^{(1)}(D) &= 1 + D^2 \\
 g^{(2)}(D) &= 1 + D + D^2 \quad \dots\dots\dots (22) \\
 g^{(3)}(D) &= 1 + D + D^2
 \end{aligned}$$

该生成多项式用八进制记数法记为[5,7,7]。

母码编码码率为 1/3 的卷积编码器如图 15 所示。

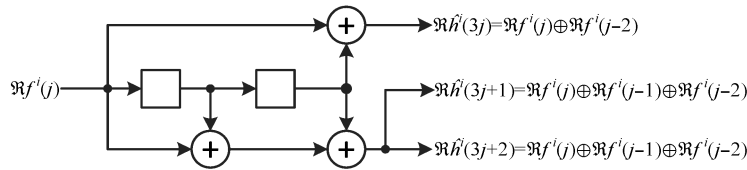


图 15 母码编码码率为 1/3 的卷积编码器

SCPPM 编码器的第 i 个二进制输入信息块的第 j 个符号 $Rf^i(j)$ 进入卷积编码器后，产生 3 个编码符号 $R\hat{h}^i(3j)$, $R\hat{h}^i(3j+1)$, $R\hat{h}^i(3j+2)$ 分别对应多项式 $g^{(1)}(D)$ 、 $g^{(2)}(D)$ 、 $g^{(3)}(D)$ ， Rf^i 的所有 \hat{k} 个二进制位输入编码器后，编码器产生卷积码字可用公式(23)表示：

$$R\hat{h}^i = R\hat{h}^i(0), R\hat{h}^i(1), \dots, R\hat{h}^i(3\hat{k}-1) \quad \dots\dots\dots (23)$$

6.8.2.2 初始化

在对每个输入信息块进行编码之前，应将编码器初始化为“全 0”状态。由于上一步骤的添加终止二进制位，第 1 个输入信息块以外的每个输入信息块，自然会被初始化为“全 0”状态。

6.8.2.3 删余

使用表 2 中给出的删余模式，可以对编码码率为 1/3 的码进行删余，得到编码码率为 1/2 码或 2/3 码。删余操作流程见图 16。

表 2 卷积编码器删余模式

编码码率	P_0	P_1	P_2	P_3	P_4	P_5
1/3	1	1	1	1	1	1
1/2	1	1	0	1	1	0
2/3	1	1	0	0	1	0

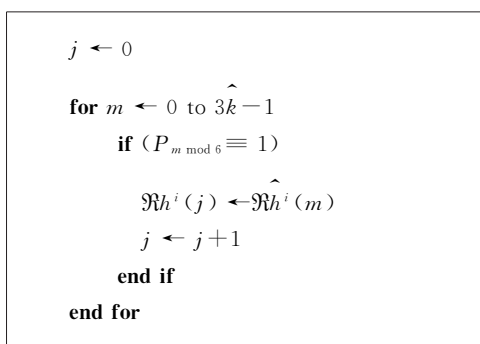


图 16 删余操作流程图

删余得到的第 i 个卷积码字用公式(24)表示：

$$\mathfrak{R}h^i = \mathfrak{R}h^i(0), \mathfrak{R}h^i(1), \dots, \mathfrak{R}h^i(15\ 119) \dots\dots\dots(24)$$

无论何种编码码率对应的 \hat{k} 取值,都会使卷积码的码字长度为 15 120。

6.8.3 编码交织器

每个长度为 15 120 bit 的卷积码字需要通过一个深度为 15 120 bit 的块交织器进行置换,对于 $i \in \{0, 1, \dots, C-1\}$,第 i 个交织码字用公式(25)表示：

$$\mathfrak{R}h^i = \mathfrak{R}h^i(0), \mathfrak{R}h^i(1), \dots, \mathfrak{R}h^i(15\ 119) \dots\dots\dots(25)$$

对于 $j \in \{0, 1, \dots, 15\ 119\}$,第 i 个交织码字的第 j 个码符号用公式(26)和公式(27)表示：

$$\mathfrak{R}h^i(j) = \mathfrak{R}h^i(\pi[j]) \dots\dots\dots(26)$$

其中：

$$\pi[j] = (11j + 210j^2) \bmod 15\ 120 \dots\dots\dots(27)$$

交织器可以通过将码符号从位置“0”到位置“15 119”顺序写入,并以交织顺序从位置 $\pi(0)$ 到 $\pi(15\ 119)$ 读出来实现。

编码交织器是 SCPPM 编码器的一部分,与信道交织器无关。

6.8.4 累加器

交织卷积码字进入一个编码码率为 1 且传递函数为 $1/(1+D)$ 的累加器,如图 17 所示。

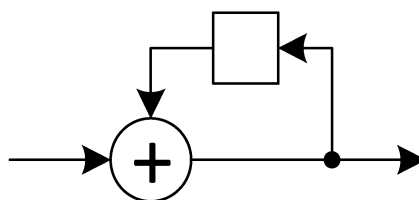


图 17 累加器

在对每个交织卷积码字进行编码之前,累加器应初始化为“全 0”状态,具体如下:对于第 i 个码字,累加器的第 j 个输出,由公式(28)给出：

$$\mathfrak{R}h^i(j) = \begin{cases} \mathfrak{R}h^i(j), & j = 0 \\ \mathfrak{R}h^i(j-1) \oplus \mathfrak{R}h^i(j), & 1 \leq j < 15\ 120 \end{cases} \dots\dots\dots(28)$$

6.8.5 PPM 符号映射器

累加器输出的码符号被映射为 M 阶 PPM 符号,其中 $M \in \{4, 8, 16, 32, 64, 128, 256\}$ 是一个管理参

数,具体说明如下。

- a) 每 $m = \log_2(M)$ 个二进制符号被组合在一起映射为一个 PPM 符号,该符号为 $\{0, 1, \dots, M-1\}$ 中的一个整数值。输出 SCPPM 码字序列 $\mathfrak{R}q^0, \mathfrak{R}q^1, \mathfrak{R}q^{C-1}$, 每个码字包含 $S = \frac{15\ 120}{m}$ 个 PPM 符号。
- b) 对于 $j \in \{0, 1, \dots, S-1\}$, 第 i 个 SCPPM 码字的第 j 个 PPM 符号为公式(29):

$$\mathfrak{R}q^i(j) = \sum_{a=0}^{m-1} 2^{m-a-1} \cdot \mathfrak{R}n^i(mj + a) \quad \dots\dots\dots(29)$$

这样,一个 PPM 符号可以简单地用一个整数来表示,整数值对应一个 m bit 的二进制码组。

示例: 当 $M = 16$, 累加器输出为:

$$\mathfrak{R}n^i(0), \mathfrak{R}n^i(1), \mathfrak{R}n^i(2), \mathfrak{R}n^i(3), \mathfrak{R}n^i(4), \mathfrak{R}n^i(5), \mathfrak{R}n^i(6), \mathfrak{R}n^i(7), \dots = \underbrace{1, 1, 0, 1, 0, 1, 0, 1, \dots}_{13} \dots \underbrace{1, 0, 1, 0, 1, \dots}_{5}, \dots$$

对应 PPM 符号 $\mathfrak{R}q^i(0), \mathfrak{R}q^i(1), \dots = 13, 5, \dots$, 由于 $15\ 120$ 是 $m = \log_2(M)$ 的倍数, 因此分组中没有剩余的码符号。

6.9 信道交织器

6.9.1 信道交织器工作原理

PPM 符号 $\mathfrak{R}\hat{q}$ 序列通过卷积交织器进行信道交织, 如图 18 所示。

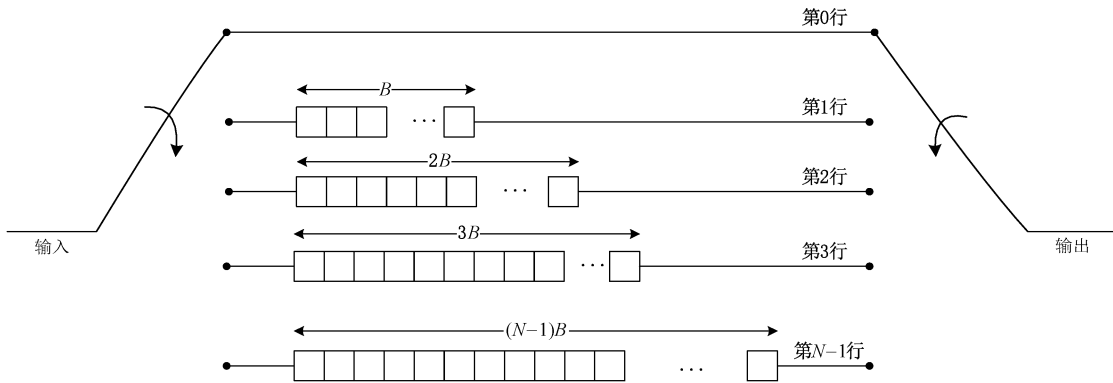


图 18 卷积信道交织器

图中信号交织器的输入符号可以用 SCPPM 码字序列 $\mathfrak{R}q = \mathfrak{R}q^0, \mathfrak{R}q^1, \dots, \mathfrak{R}q^{C-1}$ 表示, 它可以被看作由 PPM 符号组成的一维向量, 用公式(30)和公式(31)表示:

$$\mathfrak{R}\hat{q} = \mathfrak{R}\hat{q}(0), \mathfrak{R}\hat{q}(1), \dots, \mathfrak{R}\hat{q}(CS - 1) \quad \dots\dots\dots(30)$$

对于 $i \in \{0, 1, \dots, C-1\}$ 和 $j \in \{0, 1, \dots, S-1\}$,

$$\mathfrak{R}\hat{q}(iS + j) = \mathfrak{R}q^i(j) \quad \dots\dots\dots(31)$$

N 和 B 是管理参数, 选择参数时应使 BN 是 S 的倍数, 并且 S 是 N 的倍数。交织器有 N 行, 第 i 行包含长度为 iB 的移位寄存器, 这表示其缓存 iB 个 PPM 符号。

6.9.2 信道交织器初始化

在信道交织之前, 图 18 的移位寄存器可能处于任意状态。

6.9.3 信道交织器操作

6.9.3.1 信道交织的位置连接

输入的 PPM 符号 \mathfrak{R}_q , 从第 0 行开始, 以循环递增的顺序, 被多路分解到 N 行中。 N 行中的移位寄存器, 同样从第 0 行开始, 以循环递增的顺序, 被多路复用至输出端。在信道交织器的每一步操作中, 多路分解器和多路复用器都应连接在同一行中。

交织器的第 i 个输出可用公式(32)表示:

$$\mathfrak{R}_r(i) = \mathfrak{R}_q(\sigma[i]) \quad \dots\dots\dots(32)$$

其中, $\sigma[i]$ 以递归形式定义为公式(33):

$$\sigma[i] = \begin{cases} i, & i \bmod N = 0 \\ \sigma[i-1] - NB + 1, & \text{其他} \end{cases} \quad \dots\dots\dots(33)$$

$\sigma[i]$ 取负值时, 对应交织器寄存器的初始存储值; $\sigma[i]$ 取大于 $CS-1$ 的值时, 对应寄存器的结束存储值。在这些情况下, $\mathfrak{R}_r(i)$ 可能是任何值。

示例: 当 $N = 4$ 且 $B = 1$, 输入序列 $\mathfrak{R}_q(0), \mathfrak{R}_q(1), \mathfrak{R}_q(2), \dots$, 将产生交织器输出可表示为:

$$\begin{aligned} & \mathfrak{R}_r(0), \mathfrak{R}_r(1), \mathfrak{R}_r(2), \mathfrak{R}_r(3), \mathfrak{R}_r(4), \mathfrak{R}_r(5), \mathfrak{R}_r(6), \mathfrak{R}_r(7), \mathfrak{R}_r(8), \mathfrak{R}_r(9), \mathfrak{R}_r(10), \dots \\ & = \mathfrak{R}_q(0), \mathfrak{R}_q(-3), \mathfrak{R}_q(-6), \mathfrak{R}_q(-9), \mathfrak{R}_q(4), \mathfrak{R}_q(1), \mathfrak{R}_q(-2), \mathfrak{R}_q(-5), \mathfrak{R}_q(8), \mathfrak{R}_q(5), \mathfrak{R}_q(2), \dots \end{aligned}$$

6.9.3.2 信道交织器操作完成

当最后一个符号 $\mathfrak{R}_q(CS-1)$ 输入之后, 交织器应在 $\mathfrak{R}_q(CS-1)$ 被输出之前执行另外的 $BN(N-1)$ 步操作。

因此, 输出比输入多 $BN(N-1)$ 个符号。信道交织器的输出可用公式(34)表示:

$$\mathfrak{R}_r = \mathfrak{R}_r(0), \mathfrak{R}_r(1), \dots, \mathfrak{R}_r(SC + BN[N-1] - 1) \quad \dots\dots\dots(34)$$

对于 $i \in \{0, 1, \dots, SC + BN[N-1] - 1\}$, 第 i 个交织器输出定义见 6.9.3.1。

6.9.3.3 信道交织器输出重新编序形成的交织码字

可将序列 \mathfrak{R}_r 重新编序为 $R = C + BN(N-1)/S$ 个信息块, 每个信息块包含 S 个符号, 可用公式(35)表示:

$$\mathfrak{R}_r = \mathfrak{R}_r^0, \mathfrak{R}_r^1, \dots, \mathfrak{R}_r^{R-1} \quad \dots\dots\dots(35)$$

对于 $i \in \{0, 1, \dots, R-1\}$, 第 i 个信息块可用公式(36)和公式(37)表示:

$$\mathfrak{R}_r^i = \mathfrak{R}_r^i(0), \mathfrak{R}_r^i(1), \dots, \mathfrak{R}_r^i(S-1) \quad \dots\dots\dots(36)$$

对于 $j \in \{0, 1, \dots, S-1\}$,

$$\mathfrak{R}_r^i(j) = \mathfrak{R}_r(iS + j) \quad \dots\dots\dots(37)$$

\mathfrak{R}_r^i 被称为交织码字(尽管 \mathfrak{R}_r^i 包含来自不同 SCPPM 码字的符号), 包含 S 个 M 阶 PPM 符号。因为 $BN(N-1)$ 是 S 的倍数, 所以最后一个块中没有剩余的 PPM 符号。

6.10 码同步标识

在每个交织后的 SCPPM 码字前面, 加入一个由 W 个 PPM 符号组成的 CSM。在添加 CSM 之

后,第 i 个交织码字的第 j 个 PPM 符号可用公式(38)表示:

$$\mathfrak{R}_s^i(j) = \begin{cases} w(j), & 0 \leq j < W \\ \mathfrak{R}_s^i(j - W), & W \leq j < S + W \end{cases} \dots\dots\dots(38)$$

CSM 与交织码字组合在一起的序列为 $\mathfrak{R}_s = \mathfrak{R}_s^0, \mathfrak{R}_s^1, \dots, \mathfrak{R}_s^{R-1}$ 。

当 $M \geq 8$ 时, W 为 16; 当 $M = 4$ 时, W 为 24。CSM 取值可用公式(39)表示:

$$w = \begin{cases} (0, 3, 1, 2, 1, 3, 2, 0, 0, 3, 2, 1, 0, 2, 1, 3, 1, 0, 3, 2, 3, 2, 1, 0), & M = 4 \\ (0, 3, 1, 2, 5, 4, 7, 6, 6, 7, 4, 5, 2, 1, 3, 0), & M = 8 \dots\dots\dots(39) \\ (0, 2, 7, 14, 1, 2, 15, 5, 8, 4, 10, 2, 14, 3, 14, 11), & M \geq 16 \end{cases}$$

该序列进行 PPM 符号重新编序得到 $\mathfrak{R}_s = \mathfrak{R}_s(0), \mathfrak{R}_s(1), \dots, \mathfrak{R}_s(R[S+W]-1)$, 且 $\mathfrak{R}_s(iS+j) = \mathfrak{R}_s^i(j)$ 。

6.11 重复

每个 PPM 符号重复 q_d 次, 重复因子 $q_d \in \{1, 2, 3, 4, 8, 16, 32\}$ 是管理参数, 对于 $i \in \{0, 1, \dots, q_d R(S+W)-1\}$, 重复器输出的第 i 个符号可用公式(40)表示:

$$\mathfrak{R}_t(i) = \mathfrak{R}_s(\lfloor i/q_d \rfloor) \bmod M \dots\dots\dots(40)$$

式中, $\lfloor x \rfloor$ 指 x 的整数部分。

6.12 时隙映射器

对于 $i \in \{0, 1, \dots, q_d R(S+W)-1\}$, 第 i 个重复 PPM 符号 $\mathfrak{R}_t(i) \in \{0, 1, \dots, M-1\}$, 被映射为长度为 M 的二进制向量可用公式(41)和公式(42)表示:

$$\mathfrak{R}_u^i = \mathfrak{R}_u^i(0), \mathfrak{R}_u^i(1), \dots, \mathfrak{R}_u^i(M-1) \dots\dots\dots(41)$$

其中, 对于 $j \in \{0, 1, \dots, M-1\}$,

$$\mathfrak{R}_u^i(j) = \begin{cases} 1, & j = \mathfrak{R}_t(i) \\ 0, & \text{其他} \end{cases} \dots\dots\dots(42)$$

每个 \mathfrak{R}_u^i 含有 $M-1$ 个“0”和一个“1”。

6.13 插入保护时隙

在每个由 M 个时隙组成的二进制向量 \mathfrak{R}_u^i 后面插入 $M/4$ 个保护时隙, 如图 19 所示。得到时隙序列 $\mathfrak{R}_v = \mathfrak{R}_v^0, \mathfrak{R}_v^1, \dots, \mathfrak{R}_v^i, \dots, \mathfrak{R}_v^{q_d R(S+W)-1}$, 其中 \mathfrak{R}_v^i 是长度为 $5M/4$ 的向量, 且可用公式(43)表示:

$$\mathfrak{R}_v^i(j) = \begin{cases} \mathfrak{R}_u^i(j), & 0 \leq j < M \\ 0, & M \leq j < 5M/4 \end{cases} \dots\dots\dots(43)$$

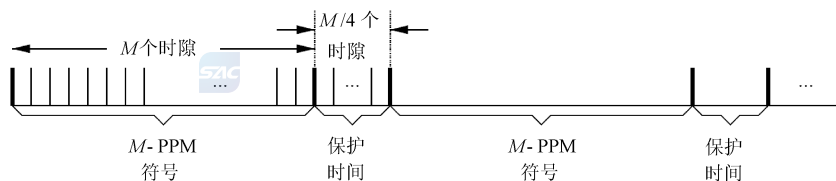


图 19 插入保护时隙

6.14 传送帧有效性确认

6.14.1 传送帧有效性确认步骤

在地面接收端,传送帧有效性的确认步骤为:

- a) 实现时隙获取和符号同步;
- b) 通过 CSM 进行码字同步;
- c) 对符号进行解交织;
- d) 对 SCPPM 码字进行译码和去随机化;
- e) 通过识别 ASM 完成 SMTF 帧同步;
- f) 移除 ASM 得到传送帧。

6.14.2 质量指示

如果传送帧是从一个或多个正确译码的 SCPPM 码字中恢复的,应标记为有效;如果传送帧是从一个或多个错误译码的码字中恢复的,则应标记为无效。

6.14.3 传送帧中的 FECF

FECF 以用来对数据链路协议子层中的附加帧进行有效性确认。

6.15 序列指示

当传送帧是前一帧的直接后继帧时,序列指示符应为“0”,当检测到间隙时,序列指示符则应为“1”。

7 上行信标和可选伴随数据传送帧的编码与同步

7.1 信号说明

第 7 章主要是针对上行信标和可选伴随低速率数据传送的编码与同步子层的相关要求。但通过将信号时隙分成以更高速率单独调制的窄时隙,也可兼容高速率数据传送。AOS/USLP 传送帧的操作流程与下行数据传送操作流程一致。

与传统的射频通信相比,在光通信中航天器通信终端接收机无须接收未调制的上行载波或捕获序列。

7.2 信标信号

下行数据与上行信标在链路两端可同时连续进行传输,上行信标可以包含传送数据,也可以不包含传送数据。当信标不携带数据时,信标为一个方波信号。当信标携带数据时,固定长度的 AOS/USLP 传送帧将作为编码和同步子层的输入,并在输出端产生指示脉冲时隙位置的二进制向量。图 20 给出了可选伴随 AOS/USLP 传送帧数据编码与同步功能实现流程。

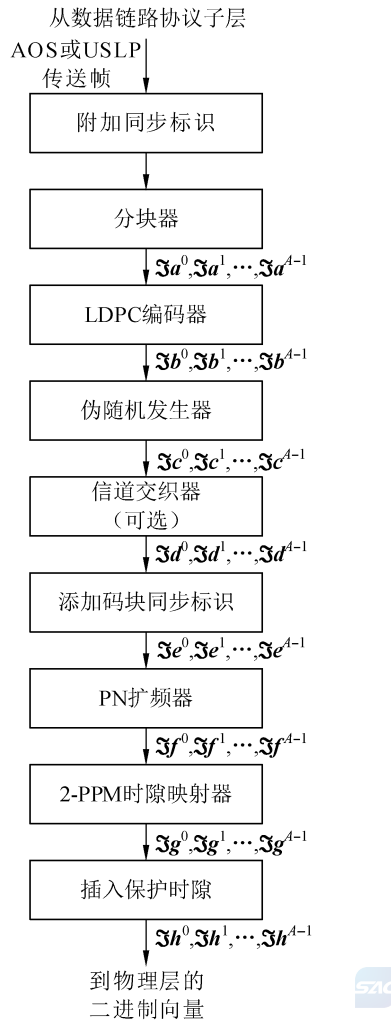


图 20 可选伴随 AOS/USLP 传送帧数据编码与同步功能实现流程图

7.3 编码与同步子层输入

7.3.1 AOS/USLP 传送帧

编码与同步子层的输入应为 AOS/USLP 传送帧。AOS 和 USLP 传送帧的帧结构规定分别见 GB/T 39345—2020 和 GB/T 42040—2022。

7.3.2 AOS/USLP 传送帧长度

AOS/USLP 传送帧的长度为固定长度，由管理参数控制。

7.4 附加同步标识

按照 6.3 规定的方式，在每个 AOS/USLP 传送帧前端插入 ASM，生成 SMTF。

7.5 分块器

7.5.1 分块长度

SMTF 分割成长度为 $k \in \{64, 256, 1024\}$ 的信息块，分块方法见 6.4。 k 为管理参数。

分块后的数据流形成一个连续的二进制向量序列 $\mathfrak{S}a = \mathfrak{S}a^0, \mathfrak{S}a^1, \dots, \mathfrak{S}a^{A-1}$ 。每个信息块长度为 k bit。对于 $i \in \{0, 1, \dots, A-1\}$, 第 i 个分块可用公式(44)表示为:

$$\mathfrak{S}a^i = \mathfrak{S}a^i(0), \mathfrak{S}a^i(1), \dots, \mathfrak{S}a^i(j), \dots, \mathfrak{S}a^i(k-1) \dots\dots\dots(44)$$

式中, $\mathfrak{S}a^i(j) \in \{0, 1\}$ 是第 i 个信息块的第 j bit。

7.5.2 补零

在传送结束时,分块器的输出应使用最少数量的“0”进行补零,使输出长度为 k 的整数倍。

7.6 LDPC 编码器

使用(128, 64)、(512, 256) 或 (2 048, 1 024)二进制 LDPC 码对每个信息块 $\mathfrak{S}a^i$ 进行编码。

对于 $i \in \{0, 1, \dots, A-1\}$, LDPC 编码后的第 i 个传送帧可用公式(45)表示:

$$\mathfrak{S}b^i = \mathfrak{S}b^i(0), \mathfrak{S}b^i(1), \dots, \mathfrak{S}b^i(j), \dots, \mathfrak{S}b^i(n-1), n = 2k \dots\dots\dots(45)$$

式中, $\mathfrak{S}b^i(j) \in \{0, 1\}$ 表示第 i 个码字的第 j bit。

7.7 伪随机发生器

7.7.1 伪随机化

每个码字 $\mathfrak{S}b^i$ 通过与二进制伪随机序列 $p(0), p(1), p(2), \dots$ 进行异或运算来实现伪随机化,对于 $i \in \{0, 1, \dots, A-1\}, j \in \{0, 1, \dots, n-1\}$, 伪随机发生器输出的第 i 个码字可用公式(46)和公式(47)表示:

$$\mathfrak{S}c^i = \mathfrak{S}c^i(0), \mathfrak{S}c^i(1), \dots, \mathfrak{S}c^i(j), \dots, \mathfrak{S}c^i(n-1) \dots\dots\dots(46)$$

其中:

$$\mathfrak{S}c^i(j) = \mathfrak{S}b^i(j) \oplus p(j) \dots\dots\dots(47)$$

式中 \oplus 表示异或运算。

7.7.2 序列规范

伪随机序列定义见 6.5。

7.7.3 序列初始化

移位寄存器初始化为全“1”状态。

7.8 信道交织器

信道交织器可用于 AOS 传送帧,由管理参数控制。

信道交织器对伪随机化码字进行操作,见 6.9。信道交织器的参数 S 为每个 AOS 传送帧伪随机化码字中的符号数, $S = n = 2k$ 。

信道交织器的行数为 N_u , 信道交织器中的移位寄存器步进增量为 B_u 。信道交织器输出表示为 $\mathfrak{S}d^0, \mathfrak{S}d^1, \dots, \mathfrak{S}d^i, \dots, \mathfrak{S}d^{A-1}$ 。

7.9 码字同步标识

使用(128, 64)或(512, 256)LDPC 码时,应在每个伪随机码字前添加 16 bit 的 CSM, CSM 的十六进制值为 EB90。使用(2 048, 2 014) LDPC 码时,应添加 64 bit 的 CSM, CSM 的十六进制值为 034776C7272895B0。

CSM 与伪随机化码字合并后称为 SMCW。当使用 16 bit 的 CSM 时,第 i 个 SMCW 可用公式(48)

表示：

$$\mathfrak{S}e^i = \mathfrak{S}e^i(0), \mathfrak{S}e^i(1), \dots, \mathfrak{S}e^i(j), \dots, \mathfrak{S}e^i(n+15) \dots\dots\dots(48)$$

7.10 PN 扩频器

对每个 SMCW 进行 PN 扩频,扩频因子为 $q_u, q_u \in \{1, 2, 3, 4, 8, 16, 32, 64\}$,每个 SMCW 的二进制符号重复出现 q_u 次,并将其按位与二进制 PN 序列 $p = p(0), p(1), \dots, p(q_u - 1)$ 进行相加。对于 $i \in \{0, 1, \dots, A-1\}$,第 i 个 PN 扩频 SMCW 表示为 $\mathfrak{S}f^i$,对于 $j \in \{0, 1, \dots, (n+16)q_u - 1\}$,第 j 个二进制符号可根据公式(49)计算得出：

$$\mathfrak{S}f^i(j) = \mathfrak{S}e^i(\lfloor j/q_u \rfloor) \oplus p(j \bmod q_u) \dots\dots\dots(49)$$

其中, $p(0), p(1), p(2), \dots$ 是表 3 中的二进制 PN 序列, $\lfloor x \rfloor$ 是 x 的整数部分。

表 3 PN 序列

q_u	p
1	0
2	01
3	011
4	0110
8	0111 0010
16	0111 0011 0110 1000
32	0111 0101 0001 1110 1101 1100 1001 0000
64	0100 1110 1101 0011 1010 0001 0101 1101 0110 0101 1100 1100 1100 1111 0000 1000

7.11 2-PPM 时隙映射器

对于 $i \in \{0, 1, \dots, A-1\}$,第 i 个 PN 扩频 SMCW 二进制符号序列可用公式(50)表示：

$$\mathfrak{S}f^i = \mathfrak{S}f^i(0), \mathfrak{S}f^i(1), \dots, \mathfrak{S}f^i(j), \dots, \mathfrak{S}f^i(\lfloor n+16 \rfloor q_u - 1) \dots\dots\dots(50)$$

按照公式(51)进行 2-PPM 调制,将每个二进制符号 $\mathfrak{S}f^i(j)$ 映射成长度为 2 的向量：

$$\mathfrak{S}g^i(j) = [\mathfrak{S}g_0^i(j), \mathfrak{S}g_1^i(j)] = \begin{cases} (1, 0) , & \mathfrak{S}f^i(j) = 0 \\ (0, 1) , & \text{其他} \end{cases} \dots\dots\dots(51)$$

第 i 个 SMCW 对应的长度为 $2(n+16)q_u$ 的时隙序列可用公式(52)表示：

$$\mathfrak{S}g^i = \mathfrak{S}g_0^i(0), \mathfrak{S}g_1^i(0), \dots, \mathfrak{S}g_0^i(j), \mathfrak{S}g_1^i(j), \dots, \mathfrak{S}g_0^i(\lfloor n+16 \rfloor q_u - 1), \mathfrak{S}g_1^i(\lfloor n+16 \rfloor q_u - 1) \dots\dots\dots(52)$$



7.12 插入保护时隙

每个 2-PPM 符号包含 2 个时隙,在其后面插入两个时隙的保护时间。对于 $i \in \{0, 1, \dots, A-1\}$,时隙序列中第 i 个 SMCW 的结果可用公式(53)表示：

$$\mathfrak{S}h^i = \mathfrak{S}g_0^i(0), \mathfrak{S}g_1^i(0), 0, 0, \mathfrak{S}g_0^i(1), \mathfrak{S}g_1^i(1), 0, 0, \dots, \mathfrak{S}g_0^i(\lfloor n+16 \rfloor q_u - 1), \mathfrak{S}g_1^i(\lfloor n+16 \rfloor q_u - 1), 0, 0 \dots\dots\dots(53)$$

7.13 AOS/USLP 传送帧有效性确认和序列指示

接收端使用 6.14 规定的流程对 AOS/USLP 传送帧进行有效性确认。使用 6.15 规定的流程对序列指示进行计算。

8 管理参数

8.1 管理参数说明

同步和信道编码的相关参数一般由管理者控制,而不是由线路上的通信协议直接处理。这些参数是一些长时间保持不变的参数,改变这些参数一般表示与特定任务相关的协议实体的主要配置发生了改变。通过使用管理系统,管理者可以将需要的信息传递给协议实体。

本章列出了信道编码和同步系统使用的管理参数。这些参数仅为抽象定义,并不代表管理系统的任何特定功能。

8.2 下行数据管理参数

表 4 规定了深空光通信下行数据的管理参数。

表 4 深空光通信下行数据管理参数

管理参数	允许值
TM/AOS/USLP 传送帧长度(8 位字节)	整数(最大值为 65 536)
PPM 阶数 M	4、8、16、32、64、128、256
编码码率 r	1/3、1/2、2/3
信道交织器的行数 N	BN 是 S 的倍数, S 是 N 的倍数
信道交织器中移位寄存器长度增量 B	
重复因子 q_d	1、2、3、4、8、16、32

8.3 上行数据管理参数

表 5 规定了深空光通信上行数据的管理参数。

表 5 深空光通信上行数据管理参数

管理参数	允许值
AOS/USLP 传送帧长度(8 位字节)	整数(最大值为 65 536)
输入信息块长度 k	64、256、1 024
PN 扩频因子 q_u	1、2、3、4、8、16、32、64
信道交织器	使用或不使用
信道交织器的行数 N_u	$B_u N_u$ 是 128 的倍数,128 是 N_u 的倍数
信道交织器中移位寄存器长度增量 B_u	

附 录 A
(规范性)
服 务 定 义

A.1 总则

本附录以原语的形式提供了服务定义,给出了服务提供者和用户之间对数据和控制信息进行逻辑交换的抽象模型。原语的定义与特定的实现方法无关。

原语参数以抽象的意义定义,规定了原语用户可利用的信息。本文件并不限定获取这些信息的特定实现方法。除了本附录规定的参数外,一个具体的实现也可能对服务用户提供其他参数,例如控制服务的参数、监视性能的参数、利用诊断的参数等。

A.2 服务综述

对于下行数据,以恒定数据速率在空间链路的物理信道上对 TM 传送帧、AOS 传送帧、USLP 传送帧序列进行单向的传输服务。

对于上行数据,以恒定数据速率在空间链路的物理信道上对 AOS 传送帧、USLP 传送帧序列进行单向的传输服务。

A.3 服务参数

A.3.1 帧

帧参数是本服务的服务数据单元。TM 传送帧在 GB/T 39351—2020 中给出了定义,AOS 传送帧在 GB/T 39345—2020 中给出了定义,USLP 传送帧在 GB/T 42040—2020 中给出了定义。一个物理信道上的任一个传送帧的长度应是相同的,并由管理建立。

A.3.2 质量指示

质量指示参数用于通知接收端的服务用户:接收的传送帧中存在一个无法纠正的错误。

A.3.3 序列指示

顺序指示参数用于通知接收端的服务用户:物理信道上的一个或多个传送帧由于帧同步失锁而丢失。

A.4 服务原语

A.4.1 OTM ChannelAccess.request 光遥测传送帧信道访问请求原语

A.4.1.1 功能:由发送端的服务用户向服务提供者传递的,要求将一个传送帧通过物理信道传送到接收端的用户。

A.4.1.2 语义:提供一个参数:OTM ChannelAccess.request(OTM 帧)。

A.4.1.3 产生时机:当需要处理和发送 OTM 传送帧时,将 OTM ChannelAccess.request 原语提交给服务提供者。

A.4.1.4 接受的影响:接收到 OTM ChannelAccess.request 原语,使得服务提供者执行第 6 章所描述的功能,并传送所产生的脉冲时隙序列。

A.4.2 OTM ChannelAccess.indication 光遥测传送帧信道访问指示原语

A.4.2.1 功能:OTM ChannelAccess.indication 是此服务的指示原语。

A.4.2.2 语义:提供一个参数:OTM ChannelAccess.indication(OTM 帧、质量指示、序列指示)。

A.4.2.3 产生时机:OTM ChannelAccess.indication 原语由服务提供者提交给接收端的服务用户,以交付一个 OTM 帧。

A.4.2.4 接受效果:服务用户接收到 OTM ChannelAccess.indication 时的响应没有定义。

A.4.3 ADU ChannelAccess.request AOS/USLP 传送帧信道访问请求原语

A.4.3.1 功能:由发送端的服务用户向服务提供者传递的,要求将一个传送帧通过物理信道传送到接收端的用户。

A.4.3.2 语义:提供一个参数:ADU ChannelAccess.request(AOS/USLP 帧)。

A.4.3.3 产生时机:当需要处理和发送 ADU 传送帧时,将 ADU ChannelAccess.request 原语提交给服务提供者。

A.4.3.4 接受的影响:接收到 ADU ChannelAccess.request 原语,使得服务提供者执行第 6 章和第 7 章所描述的功能,并传送所产生的脉冲时隙序列。

A.4.4 ADU ChannelAccess.indication AOS/USLP 传送帧信道访问指示原语

A.4.4.1 功能:ADU ChannelAccess.indication 是此服务的指示原语。

A.4.4.2 语义:提供一个参数:ADU ChannelAccess.indication(AOS/USLP 帧、质量指示、序列指示)。

A.4.4.3 产生时机:ADU ChannelAccess.indication 原语由服务提供者提交给接收端的服务用户,以交付一个 AOS/USLP 帧。

A.4.4.4 接受效果:服务用户接收到 ADU ChannelAccess.indication 时的响应没有定义。

参 考 文 献

- [1] GB/T 39345—2020 空间数据与信息传输系统 高级在轨系统空间数据链路协议
 - [2] GB/T 39351—2020 空间数据与信息传输系统 遥测空间数据链路协议
 - [3] GB/T 42040—2022 空间数据与信息传输系统 统一空间数据链路协议
-



