

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04L 12/56 (2006.01)

H04L 29/08 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200510098563.X

[43] 公开日 2007年3月7日

[11] 公开号 CN 1925459A

[22] 申请日 2005.9.2

[21] 申请号 200510098563.X

[71] 申请人 中兴通讯股份有限公司

地址 518057 广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦法律部

[72] 发明人 吴中文

[74] 专利代理机构 北京安信方达知识产权代理有限公司

代理人 王漪 王继长

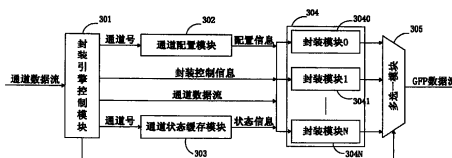
权利要求书 2 页 说明书 7 页 附图 3 页

[54] 发明名称

封装多通道数据包的装置及方法

[57] 摘要

本发明公开了一种封装多通道数据包的装置及方法，所述方法为：封装引擎控制模块首先判断是否有封装模块正在对该通道的数据进行封装处理，如果有，则继续使用该封装模块进行该通道的数据封装；否则，从多个封装模块中调出一个空闲的封装模块，同时根据通道号分别从配置模块和通道状态缓存模块中获取该通道的配置信息和封装状态信息，并将这些信息送往选中的封装模块；封装模块根据通道的配置信息和状态信息进行重装载，并对输入的数据包数据进行封装处理。当封装处理完毕后，输出 GFP 帧数据，并将该通道当前的封装状态信息保存到通道状态缓存模块中。采用本发明所述的装置及方法，可以大大减少所需的硬件资源。



1、一种封装多通道数据包的装置，其特征在于，包括：

封装引擎控制模块，进行封装模块的调度，确定对当前通道进行封装的封装模块，同时根据通道号对通道缓存和通道配置模块进行读取操作，将读取的封装状态信息和配置信息送往封装模块；

封装模块，用于生成通用成帧规程帧的核心头、净荷头部、净荷信息字段和净荷校验序列，并与输入的数据包数据进行成帧处理，形成完整的通用成帧规程帧并予以输出；

通道状态缓存模块，用于缓存各通道的封装状态信息以及封装的中间数据，当需要进行某通道的数据包封装时，则从该模块读取出相应的状态信息，当对某通道的当前数据处理完毕时，则将该通道的封装状态信息缓存到该模块；

通道配置模块，用于缓存各通道的封装配置信息以及相关统计信息。

2、如权利要求1所述的封装多通道数据包的装置，其特征在于：

所述通道数可以为64、128、256。

3、如权利要求1或2所述的封装多通道数据包的装置，其特征在于：

所述封装模块还包括，

封装控制模块，对封装模块内部的处理进行控制；

帧头生成模块，根据通道数据包的长度生成通用成帧规程帧的核心头，根据数据包的类型生成净荷头部，以及根据需要生成空闲帧；

成帧模块，用于将帧头生成模块产生的数据字节、接收到数据包数据字节以及生成的帧校验序列校验进行整合，形成一个完整的通用成帧规程帧予以输出；

帧校验序列生成模块，根据通道配置，对通用成帧规程帧的净荷部分进行CRC-32运算，在净荷部分结束时，将生成的32位帧校验序列送给成帧模块；

加扰模块, 根据通道配置, 对通用成帧规程帧的净荷部分和帧校验序列校验部分进行 $X43+1$ 的加扰运算。

4、如权利要求 3 所述的封装多通道数据包的装置, 其特征在于:

所述封装模块的个数由该模块完成一次数据处理所需要的时钟周期数决定。

5、一种封装多通道数据包的方法, 其特征在于, 包括:

步骤 1、获取需要进行封装处理的通道号, 并根据通道号从对应的物理端口提取需要进行封装的数据包数据;

步骤 2、查询是否有封装模块正在进行该通道的数据封装处理, 若有, 则继续使用该封装模块进行数据封装处理, 转入步骤 5, 若无, 则从封装处理模块中调度出一个空闲的, 对当前通道的数据进行处理, 转入步骤 3;

步骤 3、根据通道号, 查询该通道的封装状态信息和配置信息, 并送往对应的封装处理模块;

步骤 4、依据步骤 3 中查询到的状态和配置信息, 对选中的封装模块进行装载, 让封装模块恢复到该通道的最近封装状态, 以保证通道数据封装处理的连续性;

步骤 5、对通道的数据包数据进行封装处理, 对选定通道的数据处理完毕后, 保存该通道当前的封装处理状态; 同时将当前的封装处理模块标记为空闲状态。

6、如权利要求 5 所述的封装多通道数据的方法, 其特征在于:

根据一个封装模块完成一次数据封装处理所需要的最大时钟周期数, 确定一个封装模块是否处于空闲状态;

或当一个封装处理模块完成了指定的数据进行封装处理, 则自动转入空闲状态。

7、如权利要求 5 所述的封装多通道数据的方法, 其特征在于:

所述封装模块的个数由该模块完成一次数据处理所需要的时钟周期数决定。

封装多通道数据包的装置及方法

技术领域

本发明涉及数字传输领域，具体地说，涉及一种多通道数据包的封装装置及方法。

背景技术

通用成帧规程 (GFP, Generic Framing Procedure) 最近获得了极大进展，GFP 可解决许多在 SONET/SDH 上或直接在光纤上传输数据的相关问题。

首先，它采用的是一种强大的帧构造机制，提高了链接的可靠性。此外，它可以按原始方式传输以太网数据分组，进行以太网数据帧的本地传输。

GFP 还提供了一种通用机制，用于传输统计复用数据分组 (如以太网/IP)、基于块的数据结构 (如 SCSI 等存储协议) 及对延迟敏感的音频/视频信号传输。

GFP 具有多层结构，与协议无关的层支持用于各种有效载荷的强大的帧描述。它采用分组长度及相关的 CRC (循环冗余码校验) 值进行描述。由于长度是已知的，就很容易确定一个数据帧的帧尾和下一数据帧的帧头，从而可构造强大的帧。与协议有关的层分为两种，基于帧的 GFP 适用于以太网/IP 有效载荷和各种长度的分组。它提供与基础信息相配合的灵活结构，以及与用户特殊应用信息相配合的用户定义的分组头。

如图 1 所示，GFP 的帧结构包括 GFP 帧核心头、净荷头部、净荷信息字段和净荷校验序列。核心头包括 16 位的净荷长度指示和净荷长度的错

误校验值。净荷头部包括净荷类型、净荷类型错误校验、GFP 的扩展头、扩展帧头错误校验四部分。

净荷帧校验序列标识 (PFI)、扩展帧头标识 (EXI) 和客户净荷标识 (UPI), PTI 为 3bit, 指示帧的内容类型; PFI 为 1bit, 指示帧尾是否携带帧校验序列 (FCS); EXI 为 4bit, 指示扩展帧头的类型; UPI 为 8bit, 指示用户净荷数据类型。

在通信网络中, 为了将数据包在光传输网上进行传输, 需要先将数据包进行 GFP 封装, 再将封装而成的 GFP 帧映射到光传输网的通道上进行传输。

在现有的技术中, 一个封装处理模块只能封装一个通道的数据, 譬如专利《一种封装数据流的方法》, 专利号公开号为 CN1571415A, 公开日期为 2005 年 1 月 26 日, 或将来自多个输入端口的数据封装到一个传输通道上去。

但上述解决方案共同的问题是, 如果有多个传输通道, 则需要对应数量的封装模块进行封装处理, 这对小规模芯片系统还可行。如果系统的传输通道数为 64、128、256 或以上时, 此时会造成芯片系统的面积和规模成倍的增加, 这对芯片的工艺会带来更高的要求, 对降低芯片的规模和功耗带来更大的压力。

发明内容

为了克服现有技术中存在的上述问题, 本发明提供了一种封装模块复用的装置及方法, 可以实现利用一个封装模块组完成对多通道的数据包进行封装。

本发明所构造的封装多通道数据包的装置, 包括:

封装引擎控制模块, 进行封装模块的调度, 确定对当前通道进行封装的封装模块, 同时根据通道号对通道缓存和通道配置模块进行读取操作, 将读取的封装状态信息和配置信息送往封装模块;

封装模块, 用于生成通用成帧规程帧的核心头、净荷头部、净荷信息字

段和净荷校验序列，并与输入的数据包数据进行成帧处理，形成完整的通用成帧规程帧并予以输出；

通道状态缓存模块，用于缓存各通道的封装状态信息以及封装的中间数据，当需要进行某通道的数据包封装时，则从该模块读取出相应的状态信息，当对某通道的当前数据处理完毕时，则将该通道的封装状态信息缓存到该模块；

通道配置模块，用于缓存各通道的封装配置信息以及相关统计信息。

所述通道数可以为 64、128、256。

所述封装模块还可以包括，

封装控制模块，对封装模块内部的处理进行控制；

帧头生成模块，根据通道数据包的长度生成通用成帧规程帧的核心头，根据数据包的类型生成净荷头部，以及根据需要生成空闲帧；

成帧模块，用于将帧头生成模块产生的数据字节、接收到数据包数据字节以及生成的帧校验序列校验进行整合，形成一个完整的通用成帧规程帧予以输出；

帧校验序列生成模块，根据通道配置，对通用成帧规程帧的净荷部分进行 CRC-32 运算，在净荷部分结束时，将生成的 32 位帧校验序列送给成帧模块；

加扰模块，根据通道配置，对通用成帧规程帧的净荷部分和帧校验序列校验部分进行 $X43+1$ 的加扰运算。

所述封装模块的个数由该模块完成一次数据处理所需要的时钟周期数决定。

本发明所提出的一种封装多通道数据包的方法，包括以下步骤：

步骤 1、获取需要进行封装处理的通道号，并根据通道号从对应的物理端口提取需要进行封装的数据包数据；

步骤 2、查询是否有封装模块正在进行该通道的数据封装处理，若有，则继续使用该封装模块进行数据封装处理，转入步骤 5，若无，则从封装处理模块中调度出一个空闲的，对当前通道的数据进行处理，转入步骤 3；

步骤 3、根据通道号，查询该通道的封装状态信息和配置信息，并送往对应的封装处理模块；

步骤 4、依据步骤 3 中查询到的状态和配置信息，对选中的封装模块进行装载，让封装模块恢复到该通道的最近封装状态，以保证通道数据封装处理的连续性；

步骤 5、对通道的数据包数据进行封装处理，对选定通道的数据处理完毕后，保存该通道当前的封装处理状态；同时将当前的封装处理模块标记为空闲状态。

根据一个封装模块完成一次数据封装处理所需要的最大时钟周期数，确定一个封装模块是否处于空闲状态；

或当一个封装处理模块完成了指定的数据进行封装处理，则自动转入空闲状态。

所述封装模块的个数由该模块完成一次数据处理所需要的时钟周期数决定。

本发明设计所占的资源与逻辑通道的数量没有线性关系，即随着逻辑通道数目的增加，封装模块所占用的资源增加不多，因而当需要对更多的逻辑通道进行封装处理时，只需要增加相关的配置信息和状态缓存即可，并不需要增加任何处理逻辑，利用该方法和装置可以大大减少所需的硬件资源。

本发明适合于通道数为 64 或以上的封装系统，本发明的封装模块组需要的封装模块的个数决定于完成一次数据封装所需的最大时钟周期数，并不随逻辑通道数的增加而增加。

附图说明

图 1 是本发明 GFP 数据帧的帧格式示意图；

图 2 是本发明所述封装多通道数据包的方法流程图；

图 3 是本发明所述封装多通道数据包的装置结构图；

图 4 是本发明所述装置中封装模块的结构示意图。

具体实施方式

下面参照附图并举实施例，对本发明进一步详细说明。

在通信网络中，往往需要将来自一个或多个物理端口的数据封装到不同的 SDH 通道中去，对每个 SDH 通道，都需要一个封装模块。这对需要处理 64、128、256 或更多 SDH 传输通道的芯片系统来说，会造成芯片系统的面积和规模成倍的增加。因此本发明需要解决多个传输通道共用一个封装模块的问题。

参见图 2 所示，实现本发明的方法包括以下步骤：

步骤 201、获取需要进行封装处理的通道号，并根据通道号从对应的物理端口提取需要进行封装的数据包数据；

步骤 202、查询是否有封装模块正在进行该通道的数据封装处理，如果有，则继续使用该封装模块进行数据封装处理，转入步骤 206；如果没有，则从封装处理模块中调度出一个空闲的，对当前通道的数据进行封装处理，转入步骤 203；

步骤 203、系统有多个封装模块，封装模块的个数决定于完成一次数据封装处理所需的时钟周期数，在此对各封装模块的状态进行轮循，调度出一个处于空闲状态的封装模块，用于对当前通道的数据进行封装处理；

步骤 204、根据通道号，查询该通道的封装状态信息和配置信息，并送往对应的封装处理模块；

步骤 205、根据步骤 204 获得的通道封装状态信息和配置信息，对步骤 203 选中的封装模块进行装载，让封装模块恢复到该通道的最近封装状态，以保证通道数据封装处理的连续性；

步骤 206、对通道的数据进行封装处理，然后跳出本流程。

确定一个封装模块是否处于空闲状态可以包括有两种方式：一种是在步骤 203 中，根据一个封装模块完成一次数据封装处理所需的最大时钟周期数，当一个封装模块在最近连续该最大时钟周期数内没有进行数据封装处

理，则自动转入空闲状态；另一种是在步骤 206 中，当一个封装处理模块完成了当前指定的通道数据的封装处理，则自动转入空闲状态；

参见图 3 所示，基于上述方法，实现本发明的装置包括封装引擎控制模块、封装模块、通道状态缓存模块和通道配置模块四部分。

其中，封装引擎控制模块，进行封装模块的调度，确定对当前通道进行封装的封装模块，同时根据通道号对通道缓存和通道配置模块进行读取操作，将读取的封装状态信息和配置信息送往封装模块。

封装模块又包括封装控制模块、帧头生成模块、成帧模块、FCS 生成模块和加扰模块，用于生成 GFP 帧的核心头、净荷头部、净荷信息字段和净荷校验序列，并与输入的数据包数据进行成帧处理，形成完整的 GFP 帧并予以输出，本装置包含多个封装模块，封装模块的个数决定于该模块完成一次数据处理所需要的时钟周期数。

通道状态缓存模块，用于缓存各通道的封装状态信息以及封装的中间数据，当需要进行某通道的数据包封装时，则从该模块读取相应的状态信息；当对某通道的当前数据处理完毕时，则将该通道的封装状态信息缓存到该模块。

通道配置模块，用于缓存各通道的封装配置信息以及相关统计信息。

图 3 所示的本发明装置的工作过程如下：

封装引擎控制模块接收来自下游模块，如 SDH 传输通道的数据封装请求，此时，必须确保该传输通道的数据的连续性，否则会造数据包传输的中断。

封装引擎控制模块首先判断是否有封装模块正在对该通道的数据进行封装处理，如果有，则继续使用该封装模块进行该通道的数据封装；

否则，从多个封装模块中调出一个空闲的封装模块，同时根据通道号分别从配置模块和通道状态缓存模块中获取该通道的配置信息和封装状态信息，并将这些信息送往选中的封装模块；

封装模块根据通道的配置信息和状态信息进行重装载，并对输入的数据包数据进行封装处理。当封装处理完毕后，输出 GFP 帧数据，并将该通道当前的封装状态信息保存到通道状态缓存模块中。

下面详细说明封装模块的工作过程。

参见图 4 所示，所述封装模块进一步包括：封装控制模块、帧头生成模块、成帧模块、FCS 生成模块和加扰模块。

其中，封装控制模块接收通道的配置信息和封装状态信息，对该封装模块进行装载，恢复对应通道的最近封装状态和中间数据。

帧头生成模块接收数据包的长度和类型，生成 GFP 帧核心头、净荷头部、净荷信息字段和净荷校验序列，并将这些数据送往成帧模块进行处理。

成帧模块用于将帧头生成模块产生的数据字节、接收到的数据包数据字节以及生成的 FCS 校验进行整合，形成一个完整的 GFP 帧予以输出，并将需要进行 FCS 校验的帧数据字节送往 FCS 生成模块进行 FCS 校验码的运算。

FCS 生成模块，根据通道配置，对 GFP 帧的净荷部分进行 CRC-32 运算，在净荷部分结束时，将生成的 32 位 FCS 送给成帧模块。

加扰模块，根据通道配置，对 GFP 帧的净荷部分和 FCS 校验部分进行 X43+1 的加扰运算。

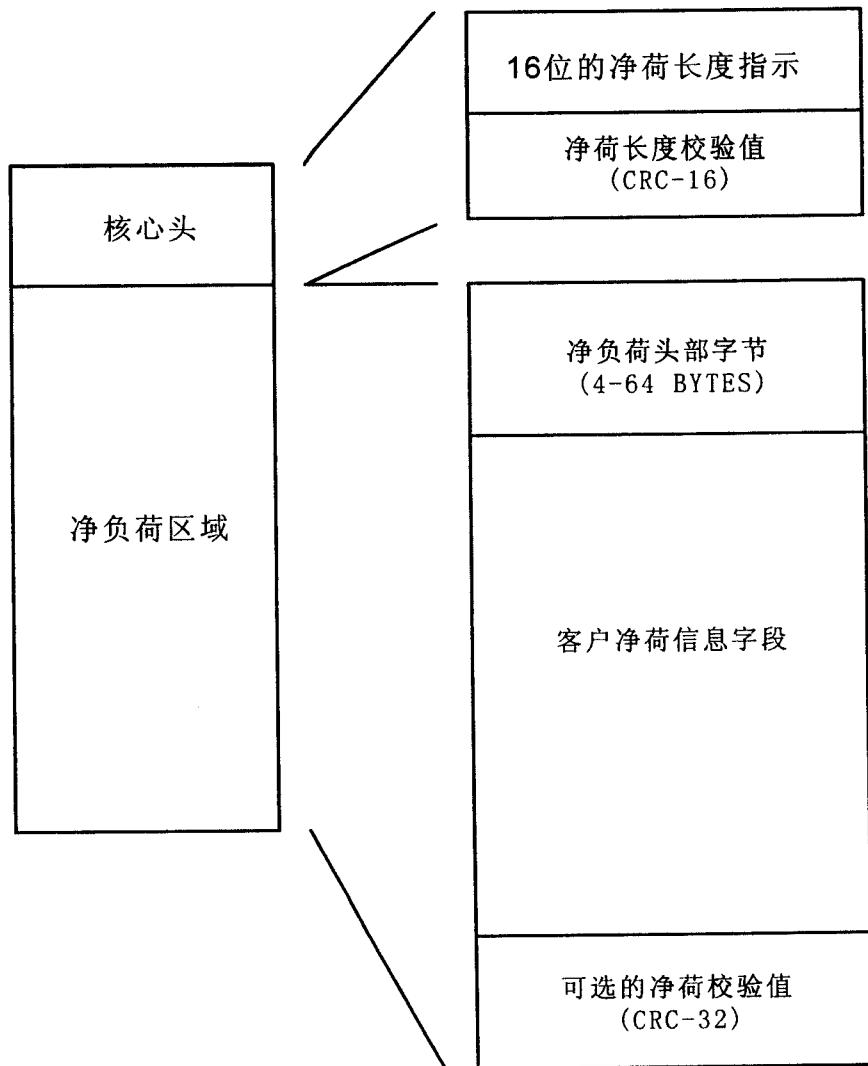


图 1

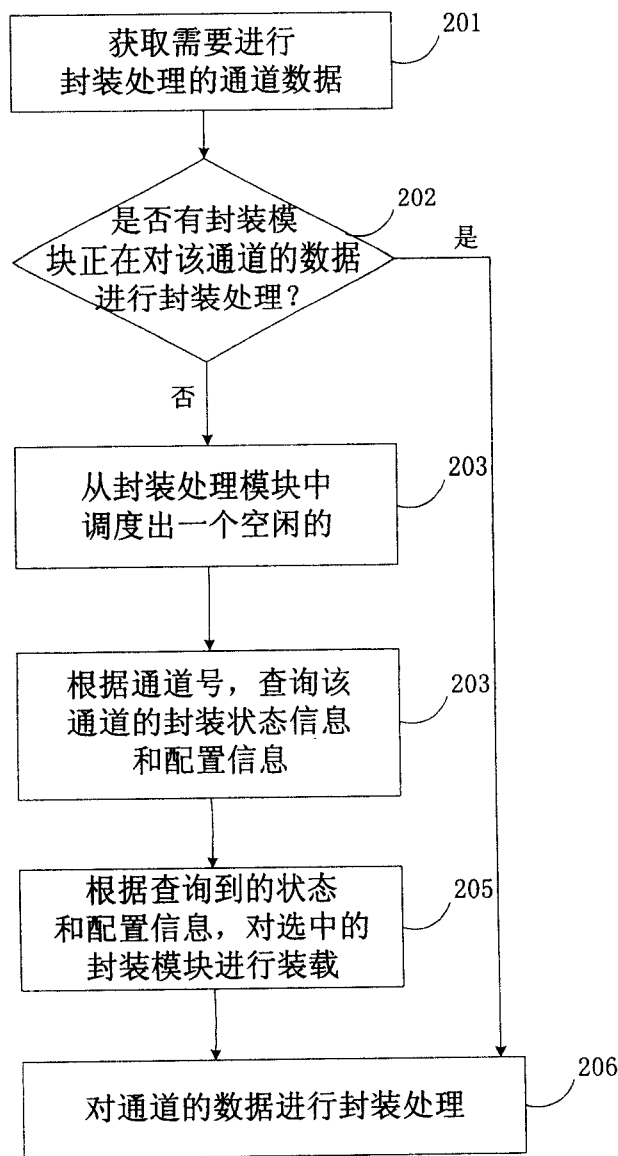


图 2

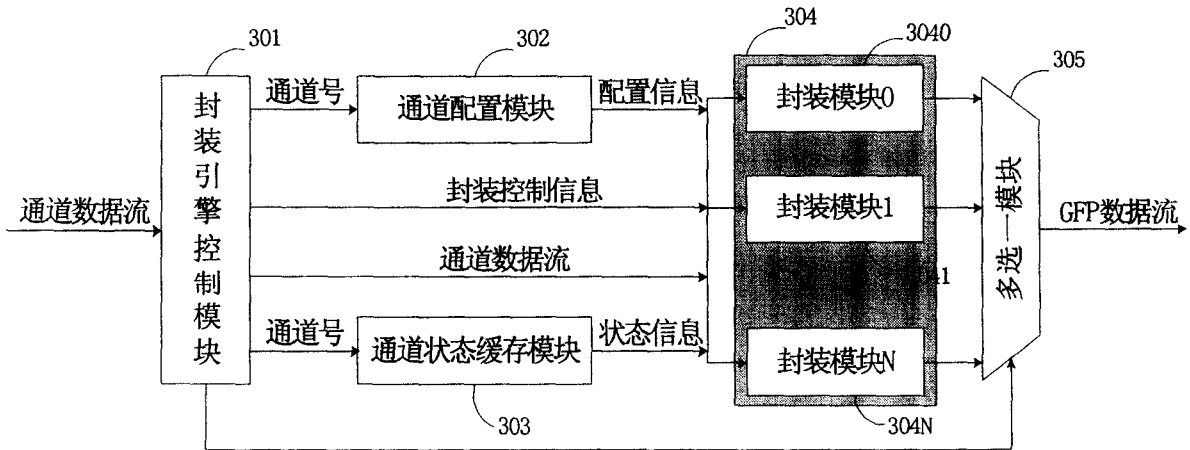


图 3

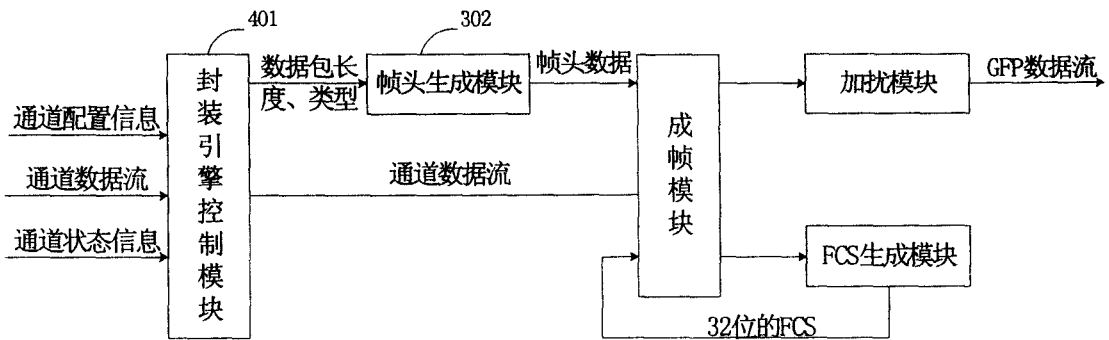


图 4