

一种数据流的解交织的方法、设备和系统

5 本申请要求于 2024 年 01 月 16 日提交的申请号为 202410063609.7、发明名称为“一种数据流的解交织的方法、设备和系统”的中国专利申请，以及 2023 年 6 月 27 日提交的申请号为 202310773216.0、发明名称为“一种用于解交织的同步方法”的中国专利申请，以及于 2023 年 7 月 3 日提交的申请号为 202310808219.3、发明名称为“一种数据流的解交织的方法、设备和系统”的中国专利申请的优先权，它们的全部内容通过引用结合在本申请中。

10 技术领域

本申请涉及通信领域，尤其涉及一种数据流的解交织的方法、设备和系统。

背景技术

在以太网数据传输过程中，由于环境干扰、系统误差等多种因素，数据接收端所接收到的数据与数据发送端所发送出的数据之间会存在不一致，即误码不可避免。现前向纠错(Forward Error Correction, FEC)以及交织器(interleaver)等手段可以降低误码率(bit error rate, BER)。交织器输出的数据经过通信信道传输至数据接收端，输入数据接收端的解交织器进行解交织，然后进入 FEC 解码器进行解码。

20 为了确定正确的解交织位置，数据发送端通常数据中周期性地插入专门用于确定解交织位置的固定数据段，数据接收端通过寻找这些固定数据段来判断出正确的解交织位置。然而，这种方式需要在原有需要传输的数据之外再额外传输一段固定数据段，从而带来更高的传输成本。

发明内容

提供了一种数据流的解交织的方法，用于解决插入固定数据段导致的解交织效率低以及网络传输效率低的技术问题。

25 第一方面，提供了一种数据流的解交织的方法，该方法包括，获取经过前向纠错 FEC 编码的第一交织数据流，基于第一交织数据流的交织粒度，获得第一交织数据流对应的 N 条子数据流，N 为第一交织数据流的交织深度，且 N 为大于 1 的正整数，基于 N 条子数据流中至少两条子数据流的 FEC 码字边界，确定解交织位置，基于该解交织位置，根据第一交织数据流获得第一解交织数据流。其中，解交织位置又可以称为正确的解交织位置，即用于正确的执行解交织过程，恢复出原始码字、及原始码字的相对顺序，以便于恢复出编码前的数据流的位置。通过该方法，无需插入额外的专门用于寻找解交织位置的固定数据段，即可完成解交织，正确恢复数据。从而避免增加额外数据，节省硬件资源，提高解交织效率，同时降低传输成本，提升网络效率。

30 在一种可能的实现方式中，N 条子数据流按照顺序排列，所述基于所述 N 条子数据流中至少两条子数据流的 FEC 码字边界，确定解交织位置，包括：基于第一子数据流的 FEC 码字边界在第二子数据流的 FEC 码字边界之后，确定所述解交织位置为所述第二子数据流的 FEC 码字边界；其中，所述第一子数据流和所述第二子数据流为所述 N 条子数据流中的两条相邻子数据流。其中，FEC 码字边界的先后顺序是指 FEC 码字边界位置上的数据在第一交织数据流中传输的先后顺序，或称为快慢顺序。当发现相邻的两条子数据流的 FEC 码字边界存在差异时，FEC 码字边界在先的子数据流即为传输更快的子数据流，该传输更快的子数据流的 FEC 码字边界即为正确的解交织位置。

35 在一种可能的实现方式中，所述第一子数据流的排列顺序在所述第二子数据流之前。

40 在一种可能的实现方式中，所述第一子数据流的 FEC 码字边界与所述第二子数据流的 FEC 码字边界相差 1 个符号。

45 在一种可能的实现方式中，所述 N 条子数据流按照顺序排列，所述基于所述 N 条子数据流中至少两条子数据流的 FEC 码字边界，确定解交织位置，包括：基于所述 N 条子数据流中每条子数据流的 FEC 码字边界对齐，确定所述解交织位置为第一子数据流的 FEC 码字边界；其中，所述第一子数据流在所述 N 条子数据流中排列顺序最先。其中，每条子数据流的 FEC 码字边界对齐，是指每条子数据流的 FEC 码字边界位置上的数据在一次解交织分发所对应的数据中，也即每条子数据流的 FEC 码字边界

位置上的数据在数据发送端的一组交织所对应的数据中。当发现各条子数据流的 FEC 码字边界对齐时，当前开始解交织分发的位置即为正确的解交织位置，也即 N 条子数据流中排列顺序最先的子数据流的 FEC 码字边界即为正确的解交织位置。

5 在一种可能的实现方式中，所述基于所述解交织位置，根据所述第一交织数据流获得第一解交织数据流，包括：基于所述解交织位置和所述 N 条子数据流的排列顺序，确定所述 N 条子数据流的解交织组合顺序；基于所述解交织组合顺序，组合所述第一交织数据流中的数据，以获得所述第一解交织数据流。

10 在一种可能的实现方式中，所述基于所述第一交织数据流的交织粒度，获得所述第一交织数据流对应的 N 条子数据流，包括：从所述第一交织数据流中选取数据流分发开始位置，基于所述第一交织数据流的交织粒度，将所述第一交织数据流分发为 N 条子数据流；其中，所述数据流分发开始位置为在所述第一交织数据流中随机选取的位置。

在一种可能的实现方式中，所述基于所述第一交织数据流的交织粒度，获得所述第一交织数据流对应的 N 条子数据流，包括：基于所述第一交织数据流的交织粒度，将所述第一交织数据流的数据轮询分发到所述 N 条子数据流中。

15 在一种可能的实现方式中，所述方法还包括：确定所述 N 条子数据流中每条子数据流的 FEC 码字边界。

在一种可能的实现方式中，所述确定所述 N 条子数据流中每条子数据流的 FEC 码字边界，包括：根据 FEC 码字特征，确定所述 N 条子数据流中每条子数据流的 FEC 码字边界。

20 在一种可能的实现方式中，所述根据 FEC 码字特征，确定所述 N 条子数据流中每条子数据流的 FEC 码字边界，包括：在每条子数据流中，分别基于假设的 FEC 码字边界进行特征值验证，根据验证结果确定每条子数据流的 FEC 码字边界。

25 在一种可能的实现方式中，所述 N 条子数据流的 FEC 码字之间为循环移位关系，所述根据 FEC 码字特征，确定所述 N 条子数据流中每条子数据流的 FEC 码字边界，包括：对获得的所述 N 条子数据流的 FEC 码字进行逆循环移位；根据 FEC 码字特征，确定逆循环移位后的所述 N 条子数据流中每条子数据流的 FEC 码字边界。

在一种可能的实现方式中，所述第一交织数据流中不包括周期性的对齐数据段，所述周期性的对齐数据段包括对齐字、帧对齐字或导引数据。

在一种可能的实现方式中，所述第一交织数据流的 FEC 编码方式为非级联码的 FEC 编码，或级联码的 FEC 编码。

30 在一种可能的实现方式中，所述 N 为 4 或 8。

在一种可能的实现方式中，所述第一交织数据流的交织类型为符号交织或卷积交织。

在一种可能的实现方式中，所述方法由物理层 PHY 芯片，转发设备，或可插拔模块执行。

在一种可能的实现方式中，所述第一交织数据流的通道速率大于等于 200 吉比特每秒 Gbps。

35 第二方面，提供了一种以太网设备，包括至少一个模块；该至少一个模块用于执行如上述第一方面或第一方面的任一可选方式所提供的方法。其中，该至少一个模块可以基于软件、硬件或软件和硬件的结合实现，该模块可以基于具体实现进行任意组合或分割。

第三方面，提供了一种以太网设备，包括存储器和处理器；该存储器用于存储计算机程序；该处理器用于执行该存储器中存储的计算机程序以使得该以太网设备执行如上述第一方面或第一方面的任一可选方式所提供的方法。

40 第四方面，提供了一种以太网设备，包括主控板和接口板，该主控板或该接口板用于实现如上述第一方面或第一方面的任一可选方式所提供的方法。

第五方面，提供了一种通信系统，该系统通信系统包括发送端设备和接收端设备，所述接收端设备用于执行如上述第一方面或第一方面的任一可选方式所提供的方法。

45 第六方面，提供了一种计算机可读存储介质，该计算机可读存储介质中存储有计算机程序，该计算机程序被执行时实现如上述第一方面或第一方面的任一可选方式所提供的方法。

第七方面，提供了一种计算机程序产品，该计算机程序产品包括程序或代码，该程序或代码被执行时实现如上述第一方面或第一方面的任一可选方式所提供的方法。

第八方面，提供了一种芯片，该芯片运行时实现如上述第一方面或第一方面的任一可选方式所提供的方法。该芯片可以是控制芯片或转发芯片，该芯片包括可编程逻辑电路和/或程序指令。

上述第二方面至第八方面的技术效果可以参考第一方面的技术效果，这里不做赘述。

5 附图说明

为了更清楚地说明本申请的技术方案，下面将对实施例中使用的附图作简单地介绍。显而易见地，下面附图只是本申请的一些实施例的附图，对于本领域普通技术人员来说，在不付出创造性劳动性的前提下，还可以根据这些附图获得同样能实现本申请的其他技术方案和附图。

- 图 1(a)为本发明实施例提供一种随机误码示意图；
- 10 图 1(b)为本发明实施例提供一种随机误码示意图；
- 图 2 为本发明实施例提供一种交织器处理过程示意图；
- 图 3 为本发明实施例提供一种交织的数据处理过程的示意图；
- 图 4(a)为本发明实施例提供一种交织的数据处理过程的示意图；
- 图 4(b)为本发明实施例提供一种交织的数据处理过程的示意图；
- 15 图 5 为本发明实施例提供一种交织的数据处理过程的示意图；
- 图 6 为本发明实施例提供一种插入数据的处理过程的示意图；
- 图 7 为本发明实施例提供一种数据流的解交织的方法流程示意图；
- 图 8(a)为本申请实施例提供一种数据流解交织方法的适用场景；
- 图 8(b)为本申请实施例提供一种数据流解交织方法的适用场景；
- 20 图 8(c)为本申请实施例提供一种数据流解交织方法的适用场景；
- 图 9 为本申请实施例提供一种循环移位的处理过程的示意图；
- 图 10 为本申请实施例提供一种存在循环移位的交织的处理过程的示意图；
- 图 11 为本申请实施例提供一种数据流的解交织的处理过程的示意图；
- 图 12 为本申请实施例提供一种数据流的解交织的处理过程的示意图；
- 25 图 13 为本申请实施例提供一种数据流的解交织的处理过程的示意图；
- 图 14 为本申请实施例提供一种数据流的解交织的处理过程的示意图；
- 图 15 为本发明实施例提供一种以太网设备的结构示意图；
- 图 16 为本发明实施例提供一种以太网设备的结构示意图；
- 30 图 17 为本发明实施例提供一种通信系统的示意图。

具体实施方式

在通信传输过程中，由于环境干扰和系统误差等多种原因，数据接收端所收到的数据与数据发送端所发出的数据之间极有可能存在不一致，这种不一致又称为误码。在通信传输过程中的误码问题不可避免。误码会引起很多问题，例如，当网络中的各种设备之间交互的关键控制信号存在误码时，可能会导致系统崩溃、数据丢失等严重问题，另外，误码的存在会在很大程度上影响网络通信时延，进而影响消费者观看视频、网络游戏以及通话等活动的体验。因此，误码率 BER 一直是通信系统的重要性能指标。数据接收端的 BER 数值越小，代表网络传输可靠性越高。

为了确保通信系统的高可靠性，行业内通常会对通信系统的 BER 提出明确要求。例如，电气电子工程师协会（Institute of Electrical and Electronics Engineers, IEEE）标准 802.3bs 中要求数据接收端所接收的数据在进入介质访问控制（media access control, MAC）子层时的 BER 应当低于 1×10^{-13} 。然而，网络中的数据在完成在网络链路中的传输后进入数据接收端时，其 BER 通常可达 2×10^{-4} 。此时，数据接收端可以采用 FEC 对数据流中的误码进行纠正，以将其恢复为发送的数据。FEC 可以消除大部分的误码，使得处理后的数据 BER 大幅降低。

FEC 的效果与误码分布有关，当两条数据流的误码分布情况不同时，即便它们的纠前 BER 相同，它们的纠后 BER 也会存在差异。其中，纠前 BER 指进行 FEC 之前的 BER，纠后 BER 指经过 FEC 之后的 BER。误码的分布情况主要可以分为两类：随机误码（random error）和非随机误码（non-random error）。为了形象说明，图 1(a)和图 1(b)中对随机误码和非随机误码进行了示例，其中 b 表示数据正确

的比特位, x 表示出现误码的比特位。随机误码表现为误码分布随机, 如图 1(a), 非随机误码表现为多个误码集中出现在较短的数据序列上, 如图 1(b), 因而, 非随机误码又可称为突发误码 (burst error)。突发误码在实际通信网络中普遍存在, 例如, 判决反馈均衡 (Decision Feedback Equalization, DFE) 导致的连续误码, 以及无线信息的多径效应引发快衰落所导致的连续误码等等。如上所示, 图 1(a)和图 1(b)中数据当前的 BER 相同, 但不同的误码分布情况将导致这些数据在经过 FEC 后的 BER 产生差异。在大多数通信传输条件下, 基于相同的纠前 FEC, 突发误码会导致比随机误码更好的纠后 BER。

针对突发误码的情况, 为了进一步降低纠后 FEC, 通常会引入交织器, 又称为信道交织器。交织器的使用方式如图 2 所示。图 2 中, T_x 为数据发送端中待输入 FEC 编码器的数据, 这些数据在数据发送端经过 FEC 编码器编码后输入至交织器进行交织, 交织器输出的数据经过通信信道传输至数据接收端, 数据在数据接收端输入解交织器进行解交织, 然后进入 FEC 解码器进行解码, R_x 为数据接收端中 FEC 解码器输出的数据, 经过 FEC 和交织处理, R_x 中的 BER 将大幅降低。

在本申请实施例中, 交织器的交织方式可以具有多种属性。交织类型是交织方式的一种属性, 交织类型包括符号交织或卷积交织。其中, 符号交织, 也可以称为分组交织 (block interleaver)。符号交织和卷积交织的在纠错性能方面基本一致, 但达到同等纠错性能的情况下, 两种方式的时延和功耗可能不同。交织深度也是交织方式的一种属性, 交织深度指示参与交织的 FEC 码字数量, 参与交织的 FEC 码字数量会影响纠错性能。

图 3 示例性地给出了符号交织的数据处理方式的示意图。

如图 3 所示, 从待交织的数据流中获取多个 FEC 编码后的码字作为参与交织的码字, 如图中的 codewordA, codewordB, codewordC, codewordD 等。当参与交织的码字来自同一个数据流时, 即数据流内的交织; 当参与交织的码字来自不同数据流时, 即数据流之间的交织。其中每个码字包括 p 个 FEC 编码后的符号 (symbol), 每个符号包括 q 个比特, 也即每个码字包括 $p \times q$ 个比特。其中, p 和 q 为正整数。例如, 每个符号的长度可以为 10 比特, 即以 10 比特为粒度的符号交织。又例如, 在脉冲幅度调制 4 (pulse amplitude modulation-4, PAM4) 中每个符号的长度可以为 2 比特, 即 PAM4 可以采用 2 比特为粒度的符号交织。又例如, p 也可以为 1, 即每个符号的长度可以为 1 比特, 即以 1 比特为粒度的符号交织。如图 3 中从待交织的数据流中选取 4 个码字参与交织, 即交织深度为 4 个码字, 当该码字为 RS 码字时, 可称作交织深度为 $4 \times RS$ 。将这 4 个码字以符号为粒度进行交替并分发为 n 条 lane 上的 n 条数据流输出, 使得每条数据流相邻的符号来自不同的码字, 即来自上述 4 个码字的符号交替出现, 则形成交织后数据序列, 即图 3 中所示的 interleaved lane 1 至 interleaved lane n 。

符号交织可以针对一条数据流进行, 也可以针对多条数据流进行。图 4(a)和图 4(b)分别示例性地给出了针对一条或多条数据流进行符号交织的数据处理方式的示意图。

如图 4(a)所示, 待交织的数据流为一条数据流 T_x 。从待交织的数据流 T_x 中获取 4 个 FEC 码字参与交织, 如图 4(a)中的 cwA, cwB, cwC, cwD。每次选取 4 个 FEC 码字进行交织, 即交织深度为 4。每个码字包括的符号数为 8, 如码字 cwA 由 8 个符号组成, 每个符号为 a , 即码字 cwA 可展开为 aaaaaaaa, 与之类似, cwB, cwC 和 cwD 分别展开为 bbbbbbbb, cccccccc, 和 dddddddd。将这 4 个码字以符号为粒度进行交替组合, 形成交织后的数据流 abcdabcdabcd...abcd, 即完成一组交织, 由于交织深度为 4, 也可以称为一组 4:1 交织。此后, 再进一步选取后续的 4 个 FEC 码字继续进行下一组 4:1 交织, 以此类推, 不再赘述。

如图 4(b)所示, 待交织的数据流为 4 条数据流 T_{x1} , T_{x2} , T_{x3} 和 T_{x4} 。从待交织的 4 条数据流中分别取出一个 FEC 码字作为参与交织的 FEC 码字, 如图 4(b)中从 T_{x1} , T_{x2} , T_{x3} 和 T_{x4} 分别选取的 cwA, cwB, cwC, cwD。每次选取 4 个 FEC 码字进行交织, 即交织深度为 4。每个码字的组成与图 4(a)相同。即码字 cwA, cwB, cwC 和 cwD 分别展开为 aaaaaaaa, bbbbbbbb, cccccccc, 和 dddddddd。将这 4 个码字以符号为粒度进行交替组合, 形成交织后的数据流 abcdabcdabcd...abcd。此后, 再进一步从待交织的 4 条数据流中分别取出后续的 4 个 FEC 码字继续进行交织, 以此类推, 不再赘述。

图 5 示例性地给出了卷积交织的一种数据处理方式的示意图。卷积交织涉及多种设计参数, 包括延时行数 (delay line number)、延时块长度 (delay block length)、延时块数量 (delay block number) 等。如图 5 所示, 从待交织的数据流中获取多个 FEC 编码后的码字参与交织, 包括 cwA, cwB, cwC, cwD 等。交织深度为 4 个码字, 即从待交织的数据流中选取 4 个码字, 即上述的 cwA, cwB, cwC, cwD, 延

时行数为4,即从上到下的4行数据line 0, 1, 2, 3。图5中D代表延时块,每个延时块由移位寄存器(LFSR)组成,可暂时存储一定数据,并且每个延时行上的延时块数量不同。交织器的输入、输出侧各有一个可切换连接的开关,用于连接某条延时行。FEC码字的数据以延时块长度为粒度按列进入卷积交织器,如图5中a,b,c,d所示。其中,当FEC码字长度与延时块长度相同时,进入卷积交织器的粒度也即FEC码字的长度,当FEC码字长度与延时块长度不相同,在进入卷积交织器时,需要FEC码字数据调整为以延时块长度为粒度进入卷积交织器。当输入侧每输入一个延时块长度的数据时,输出侧会同时输出一个延时块长度的数据,之后两侧的可切换连接开关会切换到下一条延时行,按照{line 0, line 1, line 2, line 3, line 0, line 1, line 2, line 3, line 0, line 1, ...}的顺序轮询。此外,图5中输出侧x表示延时块内存存储的4个码字之前的数据。并且,虽然图5中以一条数据流的卷积交织为例,但多条数据流之间的卷积交织与图5中所示的过程相似,区别在于参与交织的码字来源于多条数据流,在此不再赘述。

如前所述,交织器输出的数据经过通信信道传输至数据接收端,数据在数据接收端输入解交织器进行解交织,然后进入FEC解码器进行解码。其中,解交织器又可称为信道解交织器。在数据接收端的解交织过程中,首先要找到解交织位置,该解交织位置例如可以是一组交织中的第一个FEC码字边界,如图4(a)所示,其中圆圈所在的位置即为解交织位置,也即一组交织中的第一个FEC码字边界位置。从该位置开始,可以将交织数据流进行解交织,组合获得解交织数据流。容易理解的是,该解交织位置在交织数据流中周期性出现,例如每组交织中都存在该组内的第一个FEC码字边界位置。确定上述解交织位置后,可以从该解交织位置开始,对收到的交织数据流基于交织粒度和交织深度进行分发,然后将分发后的码字按顺序重新组合,获得解交织的数据流。例如,对于如图4(a)所示的4:1交织形成的交织数据流,基于交织粒度分发得到4个FEC码字,将这4个FEC码字按照A,B,C,D的顺序重新组合,得到一条解交织后的数据流。对于如图4(b)所示的交织数据流,基于交织粒度分发得到4个FEC码字,将这4个FEC码字按照A,B,C,D的顺序分别分配至4条解交织后的数据流。

由此可见,确定正确的解交织位置是解交织的前提条件。为了确定解交织位置,数据发送端可以在解交织位置之前或之后周期性地插入数据接收端已知的固定数据段,这些固定数据段专门用于寻找解交织位置,从而数据接收端通过寻找这些固定数据段来判断出解交织位置。上述固定数据段可以包括对齐字(alignment marker, AM)、帧对齐字(frame alignment word)或导引数据(pilot)等。图6为其中一种插入额外数据的方式,pppp为插入的固定数据段。可以看出,这种方式需要在原有需要传输的数据之外再额外传输一段固定数据段,从而带来更高的传输成本,此外,在带宽或锁相环频点受限的传输条件下,这种方式的适用性大大降低。

本申请提出了一种数据流解交织的方法,也使得交织数据流恢复正确数据顺序的方法。该方法可以由以太网中的物理层PHY芯片、转发设备或可插拔模块执行。其中,物理层PHY芯片及可插拔模块可以位于转发设备或服务器中,转发设备包括但不限于以太网交换机或路由器,服务器包括但不限于计算服务器或存储服务器,可插拔模块包括但不限于光模块。该方法可在接收到的数据流的任意位置开始处理,通过识别FEC码字特征等方式找到正确的解交织位置,恢复出数据发送端的交织前数据流。通过上述方法,无需插入额外的专门用于寻找解交织位置的固定数据段,即可完成解交织,正确恢复数据。从而避免增加额外数据,节省硬件资源,提高解交织效率,同时降低传输成本,提升网络效率。

图7为本申请实施例一种数据传输方法700的流程示意图。该方法可以包括:

步骤S701、获取第一交织数据流。

数据接收端获取数据流。其中,所述第一交织数据流为经过前向纠错FEC编码的数据流。在一种可能的实现方式中,所述第一交织数据流的FEC编码方式可以是非级联码的FEC编码,也可以是IEEE802.3dj标准中的级联码的FEC编码。

图8(a),图8(b)和图8(c)示出了本申请实施例中的数据流解交织方法的适用场景,其中,图8(a)为非级联码的FEC编码场景,图8(b)和图8(c)为IEEE802.3dj标准中级联码的FEC编码场景,其中虚线框为可以去除的功能。图8(b)和图8(c)中各个功能的具体内容可参见IEEE802.3dj标准,在此不再赘述。图8(b)和图8(c)中均引入了循环移位功能,其中,图8(b)中的循环移位在第二级FEC之前进行,图8(c)中的循环移位在第二级FEC之后进行。

循环移位是在级联码 FEC 编码场景下引入的功能，通过循环移位，能够打乱级联码的内码 FEC 码字的内部结构，进一步分散突发误码，降低突发误码对级联码的外码 FEC 的影响，增强纠错性能。循环移位的具体功能实现如图 9 所示，图 9 中以信道交织深度为 4 进行举例，针对每组待交织的相邻 4 个内码 FEC 码字，对每个 FEC 码字内的比特内容进行一定长度的循环移位。其中每行为 1 个内码 FEC 码字，包括 A~H 共 8 个符号，每个符号的长度例如可以为 10bit，A、B、C、D 等符号块表示来自不同级联外码 FEC 码字的符号。可见循环移位会打乱原有的级联码的内码 FEC 码字结构，即打乱包括信息位加末尾的校验位的 FEC 码字结构，从而会引起 FEC 码字特征的变化。图 10 中示出了加入了循环移位后的信道交织数据序列，其中 A.1, A.2 等每个符号例如可以为 10bit 长度的符号。

在一种可能的实现方式中，所述第一交织数据流的交织类型为符号交织或卷积交织。所述符号交织或卷积交织的具体内容参见上述介绍，在此不再赘述。

在一种可能的实现方式中，所述第一交织数据流中不包括周期性的对齐数据段，所述周期性的对齐数据段包括对齐字（alignment marker）、帧对齐字（frame alignment word）或导引数据（pilot）。如前所述，通过本申请实施例中的数据流解交织方法，无需在数据发送端为数据流中插入专门用于确定解交织位置的额外的固定数据段，即可在数据接收端完成解交织，正确恢复数据。

在一种可能的实现方式中，所述第一交织数据流的通道速率大于等于 200 吉比特每秒 Gbps。本申请实施例中的数据流解交织方法，可以适用于高速以太网传输场景，例如 IEEE 标准定义的 800GbE 或 1.6TbE 的场景。例如，所述第一交织数据流的通道速率可以为 200Gbps, 400 Gbps, 800 Gbps 或 1.6Tbps 等等。

步骤 S702、基于所述第一交织数据流的交织粒度，获得所述第一交织数据流对应的 N 条子数据流。其中，所述 N 为所述第一交织数据流的交织深度，所述 N 为大于 1 的正整数。

数据接收端在获取所述第一交织数据流之后，对该第一交织数据流进行解交织。在一种可能的实现方式中，数据接收端的处理过程可以从所述第一交织数据流中的任意位置开始。例如，数据接收端在所述第一交织数据流中随机选取起始位置，开始进行处理。

在本申请实施例中，数据接收端能够获知数据发送端在处理所述第一交织数据流过程中的参数，例如，FEC 码字长度、交织粒度、或交织深度等，在此基础上，数据接收端对所述第一交织数据流进行解交织。

在一种可能的实现方式中，确定起始位置后，数据接收端从所述起始位置开始，基于所述第一交织数据流的交织粒度，将所述第一交织数据流分发为 N 条子数据流。该起始位置又可以称为数据流分发开始位置。所述子数据流又可称为子通道数据流。

在一种可能的实现方式中，数据接收端基于所述第一交织数据流的交织粒度，将所述第一交织数据流的数据轮询分发到所述 N 条子数据流中。

在一种可能的实现方式中，所述 N 为 4 或 8。

在一种可能的实现方式中，所述 N 条子数据流不是基于分发一次性获得的，而是每次从第一交织数据流中取出一条子数据流，进行处理之后再取出下一条子数据流，进行 N 次来得到 N 条子数据流。

图 11 示例性地给出了将所述第一交织数据流分发为 N 条子数据流的过程。示例性的，其中交织深度 N 为 4。数据接收端收到交织数据流，该交织数据流中包括多个符号，如图中的 D.3, A.4, B.4, C.4...A.0, B.0, C.0。数据接收端从中选择任意位置，例如图 11 中的 C.5，从该位置开始，基于交织粒度对交织数据流进行分发，以得到 4 条子数据流。并且，将该交织数据流中的数据轮询分发到 4 条子数据流中。即 lane0 最先得到第一个符号 C.5，lane1 得到第二个符号 D.5，lane2 得到第三个符号 A.0，lane3 得到第四个符号 B.0，lane0 得到第五个符号 C.0，以此类推。分发后的每条子数据流内的数据均为一连串 FEC 码字组成，以备后续步骤中确定每条子数据流的 FEC 码字边界。

在一种可能的实现方式中，如果在数据发送端有图 8(c)中所示的循环移位存在时，即当数据发送端的循环移位在第一级 FEC 之后进行时，则在数据接收端的获得的 N 条子数据流中，每条子数据流内的每个 FEC 码字内的数据结构均已被打乱，即 N 条子数据流的 FEC 码字之间为循环移位关系。

步骤 S703、基于所述 N 条子数据流中至少两条子数据流的 FEC 码字边界，确定解交织位置。

在一种可能的实现方式中，数据接收端在获得所述第一交织数据流对应的 N 条子数据流之后，将进一步确定所述 N 条子数据流中的每条子数据流的 FEC 码字边界。在本申请实施例中，对于确定 FEC

码字边界的具体方式不做限定。在本申请实施例中，解交织位置用于正确的对数据进行解交织，恢复出原始码字、及原始码字的相对顺序，以便于恢复出编码前的数据流。

在一种可能的实现方式中，数据接收端可以根据 FEC 码字特征确定所述 N 条子数据流中每条子数据流的 FEC 码字边界。例如，在子数据流中基于假设的 FEC 码字边界进行特征值验证，根据验证结果确定 FEC 码字边界。其中，特征值例如可以是校验序列，syndrome。如图 11 所示，在一种示例中，4 条子数据流中每条子数据流的 FEC 码字边界均为符号 0 所在的位置，即 lane0 中的 FEC 码字边界为符号 C.0 所示的位置，lane1 中的 FEC 码字边界为符号 D.0 所示的位置，lane2 中的 FEC 码字边界为符号 A.0 所示的位置，lane3 中的 FEC 码字边界为符号 B.0 所示的位置。需要说明的是，每条子数据流中的 FEC 码字边界在该数据流中周期性地出现，各个 FEC 码字边界之间间隔的长度为 FEC 码字长度。

在本申请实施例中，根据 FEC 码字特征确定数据流中的 FEC 码字边界的具体内容，可以在申请日为 2020 年 5 月 19 日申请号为 202010424884.9 的专利文件，申请日为 2021 年 8 月 6 日申请号为 202110904102.6 的专利文件，以及申请日为 2022 年 5 月 12 日申请号为 202210520888.6 的专利文件中找到，它们的全部内容通过引证结合于此，犹如全部陈述的一样。

在一种可能的实现方式中，数据接收端也可以根据除 FEC 码字特征以外的其他信息确定所述 N 条子数据流中每条子数据流的 FEC 码字边界。其他信息包括但不限于：扰码信息、N 条子数据流中的 AM 等。其中，N 条子数据流中的 AM 例如可以是物理编码码子层 PCS 插入的 AM，该 AM 在数据发送端的数据交织后打乱，也即数据接收端在收到交织数据流时，无法找到该 AM，但在获得的 N 条子数据流中可能恢复出该 AM，并且利用该 AM 来确定子数据流内的 FEC 码字边界。

在一种可能的实现方式中，数据接收端可以根据 FEC 码字特征或其他信息先确定所述 N 条子数据流的其中一条子数据流的 FEC 码字边界，然后基于这一条子数据流的 FEC 码字边界进一步确定其他子数据流的 FEC 码字边界，例如先根据 FEC 码字特征或其他信息先确定所述 N 条子数据流的其中任意一条子数据流的 FEC 码字边界，然后基于这一条子数据流的 FEC 码字边界来估计其他子数据流的 FEC 码字边界的可能位置，从而大幅减少找到所有子数据流码字边界所需的运算量或功耗。如图 11 所示，在一种示例中，数据接收端先找到了 4 条子数据流中 lane0 的 FEC 码字边界，即符号 C.0 所示的位置，也可以称为位置 0。则基于交织的特征可以得知，分发后所有子数据流的 FEC 码字边界分布在同一列中，或者分布在相邻的两列中。因此在确定 lane0 的 FEC 码字边界在上述位置 0 以后，其他子数据流的 FEC 码字边界必然也在上述位置 0 或者位置 0 相邻的位置，例如位置 5。以 lane1 为例，lane1 的 FEC 码字边界必然在图中 D.0 所示的位置或 D.5 所示的位置。从而对于 lane1，只需要验证 D.0 或 D.5 这两个位置则可找到 FEC 码字边界，而不需要验证其他位置如 D.1~D.4，大幅减少了运算量。同样的，对于 lane2，只需要验证 A.1 或 A.0 则可找到 FEC 码字边界，而不需要验证其他位置如 A.2~A.5。对于 lane3，只需要验证 B.1 或 B.0 则可找到 FEC 码字边界，而不需要验证其他位置如 B.2~B.5。需要说明的是，数据接收端可以先找到 4 条子数据流中任意一条 lane 的 FEC 码字边界位置，例如 lane2 的 FEC 码字边界，再基于此确定其他 3 条子数据流的 FEC 码字边界的可能位置，进而减少找到 3 条子数据流的 FEC 码字边界所需的运算量或功耗，原理相同，在此不再赘述。

在一种可能的实现方式中，如前所述，如果在数据发送端有图 8(c)中所示的循环移位存在时，即当数据发送端的循环移位在第一级 FEC 之后进行时，则在数据接收端的获得的 N 条子数据流中，N 条子数据流的 FEC 码字之间为循环移位关系，循环移位改变了子数据流内的 FEC 码字特征。此时，需要在根据 FEC 码字特征确定所述 N 条子数据流中每条子数据流的 FEC 码字边界之前，先对每条子数据流中的 FEC 码字进行逆循环移位，以对打乱的 FEC 码字结构进行恢复。进而可根据 FEC 码字特征确定逆循环移位后的所述 N 条子数据流中每条子数据流的 FEC 码字边界。

数据接收端在确定所述 N 条子数据流中每条子数据流的 FEC 码字边界之后，基于所述 N 条子数据流中至少两条子数据流的 FEC 码字边界，确定解交织位置。如前所述，在数据接收端获得所述第一交织数据流对应的 N 条子数据流后，所述 N 条子数据流按照顺序排列。在基于所述 N 条子数据流中至少两条子数据流的 FEC 码字边界，确定解交织位置时，可以分为两种情况。

在第一种可能的情况下，其中至少有两条相邻的子数据流的 FEC 码字边界存在差异，例如，这两条相邻的子数据流可以为第一子数据流和第二子数据流。数据接收端基于第一子数据流的 FEC 码字边界在第二子数据流的 FEC 码字边界之后，确定所述解交织位置为所述第二子数据流的 FEC 码字边界。

在本申请实施例中, FEC 码字边界的先后顺序是指 FEC 码字边界位置上的数据在第一交织数据流中传输的先后顺序, 或称为快慢顺序。当发现相邻的两条子数据流的 FEC 码字边界存在差异时, FEC 码字边界在先的子数据流即为传输更快的子数据流, 该传输更快的子数据流的 FEC 码字边界即为正确的解交织位置。

5 通常来说, 上述第一子数据流的排列顺序在第二子数据流之前, 并且两个子数据流的 FEC 码字边界相差 1 个符号。

在第二种可能的情况下, 每条子数据流的 FEC 码字边界对齐。数据接收端确定 N 条子数据流中排列顺序最先的子数据流的 FEC 码字边界即为正确的解交织位置。此时, 如果当前开始解交织分发的位置为 FEC 码字边界, 则当前开始解交织分发的位置即为正确的解交织位置。

10 在本申请实施例中, 每条子数据流的 FEC 码字边界对齐, 是指每条子数据流的 FEC 码字边界位置上的数据在一次解交织分发所对应的数据中, 也即每条子数据流的 FEC 码字边界位置上的数据在数据发送端的一组交织所对应的数据中。

图 11 和图 12 分别示出了上述基于所述 N 条子数据流中至少两条子数据流的 FEC 码字边界, 确定解交织位置的两种可能的情况。

15 如图 11 所示, 如前所述, 在一种示例中, 4 条子数据流中每条子数据流的 FEC 码字边界均为符号 0 所在的位置, 即 lane0 中的 FEC 码字边界为符号 C.0 所示的位置, lane1 中的 FEC 码字边界为符号 D.0 所示的位置, lane2 中的 FEC 码字边界为符号 A.0 所示的位置, lane3 中的 FEC 码字边界为符号 B.0 所示的位置。比较 4 条子数据流中 FEC 码字边界, 可以看出 4 条子数据流中至少两条子数据流的 FEC 码字边界不在同一列中, 即不对齐。如图 11 所示, lane1 和 lane2 的 FEC 码字边界存在差异, lane1 的 FEC 码字边界分别为 D.0, lane2 的 FEC 码字边界分别为 A.0, 且 lane1 的 FEC 码字边界在 lane2 的 FEC 码字边界之后, 因此 lane2 的 FEC 码字边界 A.0 为正确的解交织位置所在。

图 12 示例性地给出了另一个将所述第一交织数据流分发为 N 条子数据流的过程。示例性的, 其中交织深度 N 为 4。与图 11 不同的是, 数据接收端从接收到的交织数据流中选择任意位置, 例如图 12 中的 A.5, 从该位置开始, 基于交织粒度对交织数据流进行分发, 以得到 4 条子数据流。并且, 将该交织数据流中的数据轮流分发到 4 条子数据流中。即 lane0 最先得到第一个符号 A.5, lane1 得到第二个符号 B.5, lane2 得到第三个符号 C.5, lane3 得到第四个符号 D.5, lane0 得到第五个符号 A.0, 以此类推。4 条子数据流中每条子数据流的 FEC 码字边界均为符号 0 所在的位置, 即 lane0 中的 FEC 码字边界为符号 A.0 所示的位置, lane1 中的 FEC 码字边界为符号 B.0 所示的位置, lane2 中的 FEC 码字边界为符号 C.0 所示的位置, lane3 中的 FEC 码字边界为符号 D.0 所示的位置。比较 4 条子数据流中 FEC 码字边界, 可以看出 4 条子数据流中 FEC 码字边界在同一列中, 即对齐。此时, 考虑 lane0 为 4 条子数据流中排列顺序最先, 则 lane0 的 FEC 码字边界 A.0 为解交织位置所在。

30 比较图 11 和图 12 可以看出, 数据发送端是对多个 FEC 码字边界对齐的 FEC 码字进行交织, 当数据接收端将随机选取的任意位置作为起始位置进行解交织时, 获得的多条子数据流中, FEC 码字边界可以是对齐的, 也可以是不对齐的。但即便是多条子数据流中的 FEC 码字边界不对齐时, 这些码字边界依旧相邻, 仅相差 1 个符号。此外, 需要说明的是, 如果确定出的多条子数据流的 FEC 码字边界是不相邻的, 相差 2 个以上的符号, 则说明前述步骤中出现了错误, 有可能是确定子数据流中 FEC 码字边界的过程中出错, 也可能是其他过程中出错。此时, 需要退回到步骤 S702, 重新确定一个起始位置, 从所述第一交织数据流中获取 N 条子数据流。

步骤 S704、基于所述解交织位置, 根据所述第一交织数据流获得第一解交织数据流。

40 在确定解交织位置后, 数据接收端可以组合第一交织数据流中的 FEC 码字, 获得解交织数据流, 恢复出交织前数据。

在一种可能的实现方式中, 除了确定所述 N 条子数据流中每条子数据流的 FEC 码字边界之外, 数据接收端还需要确定所述 N 条子数据流的解交织组合顺序, 进而, 基于所述解交织组合顺序, 组合所述 N 条子数据流中的 FEC 码字, 以获得所述第一解交织数据流。

45 在一种可能的实现方式中, 在确定所述 N 条子数据流的解交织组合顺序时, 数据接收端基于将所述第一交织数据流分发为所述 N 条子数据流的分发顺序, 以及所述 N 条子数据流中每条子数据流的 FEC 码字边界位置关系, 确定所述 N 条子数据流的解交织组合顺序。其中, 所述位置关系是根据所述

N 条子数据流中每条子数据流的 FEC 码字边界位置上的数据在所述第一交织数据流中的先后顺序确定。容易理解的是,数据在所述第一交织数据流中的先后顺序是指数据被接收的先后顺序,即先接收的数据在所述第一交织数据流中的顺序在先,后接收的数据在所述第一交织数据流中的顺序在后。

在一种可能的实现方式中,数据接收端在根据所述第一交织数据流获得第一解交织数据流时,并不要求所述第一交织数据流中的所有数据都在所述第一解交织数据流中,在一种示例中,所述第一交织数据流中的部分数据在所述第一解交织数据流中。

在一种可能的实现方式中,用于组合出第一解交织数据流的数据可以是所述第一交织数据流中的部分数据,或者说,所述第一交织数据流中的部分数据并未被组合到所述解交织数据流中。例如,用于组合出解交织数据流的数据与上述分发到所述 N 条子数据流并用于确定所述解交织位置的数据可以是第一交织数据流中的不同的数据。例如,上述分发到所述 N 条子数据流并用于确定所述解交织位置的数据可以在找到所述解交织位置后被丢弃,而并未被引入到所述解交织数据流中。例如,用于组合出解交织数据流的数据可以是在第一交织数据流中位于分发到所述 N 条子数据流的数据之后的数据。

在本申请实施例中,对于在找到所述解交织位置后获得所述解交织数据流的具体方式并不限定。在一种可能的实现方式中,在找到所述解交织位置后,可以不再将所述交织数据流分发为 N 条子数据流,而基于所述解交织位置,根据所述交织数据流获得所述解交织数据流。

图 13 给出了本申请实施例中的数据流解交织方法的一个示例。如图 13 所示,数据发送端获得待交织的单条数据流,该待交织的数据流由多个 FEC 码字组成,每个码字包含 12bit,每相邻 2bit 为一个 PAM4 symbol,如图 13 中的 A.0, A.1 等。数据发送端对该数据流进行交织,分发成 4 条子数据流,分别为图 13 中的 T0, T1, T2, T3,然后基于 PAM4 symbol 进行交织得到交织后的数据流。数据接收端获得该交织后的数据流,从中选择任意位置,例如图 13 中的 C.5,从该位置开始,基于交织粒度对交织数据流进行 PAM4 symbol 解交织分发,以得到 4 条子数据流,分别为 lane0, lane1, lane2, lane3。根据 FEC 码字特征,确定 4 条子数据流中每条子数据流的 FEC 码字边界均为符号 0 所在的位置。进而比较 4 条子数据流中 FEC 码字边界,并考虑 4 条子数据流中每条子数据流的 FEC 码字边界位置关系,以及 4 条子数据流的分发顺序,确定 4 条子数据流的排列顺序为 lane2, lane3, lane0, lane1, lane2 的 FEC 码字边界 A.0 为解交织位置所在。按照该顺序组合 4 条子数据流中的 FEC 码字,例如,从 lane2 中取出一个 FEC 码字 A.0 至 A.5,从 lane3 中取出一个 FEC 码字 B.0 至 B.5,从 lane0 中取出一个 FEC 码字 C.0 至 C.5,从 lane1 中取出一个 FEC 码字 D.0 至 D.5,按顺序组合即可获得一组解交织后的数据,以此类推,获得更多组解交织后的数据。可以看出,解交织后的数据,与数据发送端的交织前数据相同。

图 14 给出了本申请实施例中的数据流解交织方法的另一个示例。示例性的,其中子数据流的条数为 8,每个小方格表示一个 PAM4 symbol,灰色小方格为一组交织中的第一个 FEC 码字,也即正确的解交织位置。如图 14 所示,数据发送端根据 8 条子数据流进行交织,获得交织后的数据流,发送到数据接收端。数据接收端接收交织后的数据流,从任意位置开始处理,例如图 14 所示的斜线填充的小方格。根据 FEC 码字特征,确定每条子数据流的 FEC 码字边界。当相邻两条子数据流之间的 FEC 码字边界存在 1 个 PAM4 symbol 的差异时,如图 14 所示的第 2 条子数据流和第 3 条子数据流之间的 FEC 码字边界存在 1 个 PAM4 symbol 的差异,第 3 条子数据流的 FEC 码字边界更靠前,也即第 3 条子数据流为在交织数据流中传输更快的子数据流,则第 3 条子数据流的 FEC 码字边界为正确的解交织位置(如图 14 所示的第 3 条子数据流中的灰色小方格位置)。容易理解的是,如果 8 条子数据流中各条子数据流的 FEC 码字边界不存在差异,则上述任意位置即为正确的解交织位置。

通过上述方法,无需插入额外的固定数据段即可完成解交织,正确恢复数据。从而避免增加额外数据,降低传输成本,提升网络效率。

图 15 是本申请实施例提供的一种以太网设备的结构示意图,该以太网设备可用于接收数据,该以太网设备可以是图 7 所示的数据流的解交织方法中的以太网设备。基于图 15 所示的结构,该以太网设备 1500 能够执行图 7 所示的方法中的全部或部分操作。应理解到,该以太网设备可以包括比所示结构更多的附加结构或者省略其中所示的一部分结构,本申请实施例对此并不进行限制。如图 15 所示,该以太网设备包括:

解交织电路 1501,用于获取第一交织数据流,所述第一交织数据流为经过前向纠错 FEC 编码的数据流;基于所述第一交织数据流的交织粒度,获得所述第一交织数据流对应的 N 条子数据流,所述

N 为所述第一交织数据流的交织深度, 所述 N 为大于 1 的正整数; 基于所述 N 条子数据流中至少两条子数据流的 FEC 码字边界, 确定解交织位置; 基于所述解交织位置, 根据所述第一交织数据流获得第一解交织数据流。

5 在一种可能的实现方式中, 所述以太网设备还可以包括接口电路。该接口电路可以用于接收所述第一交织数据流。该接口电路还可以用于发送所述第一解交织数据流。

在一种可能的实现方式中, 所述 N 条子数据流按照顺序排列, 所述基于所述 N 条子数据流中至少两条子数据流的 FEC 码字边界, 确定解交织位置, 包括: 基于第一子数据流的 FEC 码字边界在第二子数据流的 FEC 码字边界之后, 确定所述解交织位置为所述第二子数据流的 FEC 码字边界; 其中, 所述第一子数据流和所述第二子数据流为所述 N 条子数据流中的两条相邻子数据流。

10 在一种可能的实现方式中, 所述第一子数据流的排列顺序在所述第二子数据流之前。

在一种可能的实现方式中, 所述第一子数据流的 FEC 码字边界与所述第二子数据流的 FEC 码字边界相差 1 个符号。

在一种可能的实现方式中, 所述 N 条子数据流按照顺序排列, 所述基于所述 N 条子数据流中至少两条子数据流的 FEC 码字边界, 确定解交织位置, 包括: 基于所述 N 条子数据流中每条子数据流的 FEC 码字边界对齐, 确定所述解交织位置为第一子数据流的 FEC 码字边界; 其中, 所述第一子数据流在所述 N 条子数据流中排列顺序最先。

15 在一种可能的实现方式中, 所述 N 条子数据流按照顺序排列, 所述基于所述解交织位置, 根据所述第一交织数据流获得第一解交织数据流, 包括: 基于所述解交织位置和所述 N 条子数据流的排列顺序, 确定所述 N 条子数据流的解交织组合顺序; 基于所述解交织组合顺序, 组合所述第一交织数据流中的数据, 以获得所述第一解交织数据流。

在一种可能的实现方式中, 所述基于所述第一交织数据流的交织粒度, 获得所述第一交织数据流对应的 N 条子数据流, 包括: 从所述第一交织数据流中选取数据流分发开始位置, 基于所述第一交织数据流的交织粒度, 将所述第一交织数据流分发为 N 条子数据流; 其中, 所述数据流分发开始位置为在所述第一交织数据流中随机选取的位置。

25 在一种可能的实现方式中, 所述基于所述第一交织数据流的交织粒度, 获得所述第一交织数据流对应的 N 条子数据流, 包括: 基于所述第一交织数据流的交织粒度, 将所述第一交织数据流的数据轮询分发到所述 N 条子数据流中。

在一种可能的实现方式中, 所述解交织电路还用于: 确定所述 N 条子数据流中每条子数据流的 FEC 码字边界。

30 在一种可能的实现方式中, 所述确定所述 N 条子数据流中每条子数据流的 FEC 码字边界, 包括: 根据 FEC 码字特征, 确定所述 N 条子数据流中每条子数据流的 FEC 码字边界。

在一种可能的实现方式中, 所述根据 FEC 码字特征, 确定所述 N 条子数据流中每条子数据流的 FEC 码字边界, 包括: 在每条子数据流中, 分别基于假设的 FEC 码字边界进行特征值验证, 根据验证结果确定每条子数据流的 FEC 码字边界。

35 在一种可能的实现方式中, 所述 N 条子数据流的 FEC 码字之间为循环移位关系, 所述根据 FEC 码字特征, 确定所述 N 条子数据流中每条子数据流的 FEC 码字边界, 包括: 对获得的所述 N 条子数据流的 FEC 码字进行逆循环移位; 根据 FEC 码字特征, 确定逆循环移位后的所述 N 条子数据流中每条子数据流的 FEC 码字边界。

40 在一种可能的实现方式中, 所述第一交织数据流中不包括周期性的对齐数据段, 所述周期性的对齐数据段包括对齐字、帧对齐字或导引数据。

在一种可能的实现方式中, 所述第一交织数据流的 FEC 编码方式为非级联码的 FEC 编码, 或级联码的 FEC 编码。

在一种可能的实现方式中, 所述 N 为 4 或 8。

在一种可能的实现方式中, 所述第一交织数据流的交织类型为符号交织或卷积交织。

45 在一种可能的实现方式中, 所述方法由物理层 PHY 芯片, 转发设备, 或可插拔模块执行。

在一种可能的实现方式中, 所述第一交织数据流的通道速率大于等于 200 吉比特每秒 Gbps。

其中, 关于该以太网设备所执行的操作的具体描述可参考上述图 7 所示方法实施例的具体描述,

在此不再赘述。

该以太网设备可以是物理层 PHY 芯片、转发设备或可插拔模块，其中转发设备可以是交换机、路由器等以太网转发设备，可插拔模块不限于可插拔的光模块或电模块等等。

- 5 当该以太网设备为交换机、路由器等以太网转发设备时，该以太网设备可以包括以太网芯片、或可插拔的光模块或电模块，其中的以太网芯片、或可插拔的光模块或电模块能够执行图 7 的方法中的全部或部分操作。

- 10 参见图 16，图 16 示出了本申请另一个示例性实施例提供的以太网设备 2100 的结构示意图，图 16 所示的以太网设备 2100 用于执行上述图 7 所示的数据流的解交织方法所涉及的全部或部分操作。该网络设备 2100 例如是物理层 PHY 芯片、转发设备或可插拔模块，该以太网设备 2100 可以由一般性的总线体系结构来实现。

如图 16 所示，以太网设备 2100 包括：主控板 2110 和接口板 2130。

主控板也称为主处理单元（main processing unit，MPU）或路由处理卡（route processor card），主控板 2110 用于对以太网设备 2100 中各个组件的控制和管理，包括路由计算、设备管理、设备维护、协议处理功能。主控板 2110 包括：中央处理器 2111 和存储器 2112。

- 15 接口板 2130 也称为线路接口单元（line processing unit，LPU）、线卡（line card）或业务板。接口板 2130 用于提供各种业务接口并实现数据包的转发。业务接口包括但不限于以太网接口、POS（Packet over SONET/SDH）接口等，以太网接口例如是灵活以太网业务接口（Flexible Ethernet Clients，FlexE Clients）。接口板 2130 包括：中央处理器 2131、网络处理器 2132、转发表项存储器 2134 和物理接口卡（physical interface card，PIC）2133。

- 20 接口板 2130 上的中央处理器 2131 用于对接口板 2130 进行控制管理并与主控板 2110 上的中央处理器 2111 进行通信。

- 网络处理器 2132 用于实现报文的转发处理。网络处理器 2132 的形态可以是转发芯片。转发芯片可以是网络处理器（network processor，NP）。在一些实施例中，转发芯片可以通过专用集成电路（application-specific integrated circuit，ASIC）或现场可编程门阵列（field programmable gate array，FPGA）实现。具体而言，网络处理器 2132 用于基于转发表项存储器 2134 保存的转发表转发接收到的报文，如果报文的地址为以太网设备 2100 的地址，则将该报文上送至 CPU（如中央处理器 2131）处理；如果报文的地址不是以太网设备 2100 的地址，则根据该地址从转发表中查找到该目的地址对应的下一跳和出接口，将该报文转发到该目的地址对应的出接口。其中，上行报文的处理可以包括：报文入接口的处理，转发表查找；下行报文的处理可以包括：转发表查找等等。在一些实施例中，中央处理器也可执行转发芯片的功能，比如基于通用 CPU 实现软件转发，从而接口板中不需要转发芯片。

- 30 物理接口卡 2133 用于实现物理层的对接功能，原始的流量由此进入接口板 2130，以及处理后的报文从该物理接口卡 2133 发出。物理接口卡 2133 也称为子卡，可安装在接口板 2130 上，负责将光电信号转换为报文并对报文进行合法性检查后转发给网络处理器 2132 处理。在一些实施例中，中央处理器 2131 也可执行网络处理器 2132 的功能，比如基于通用 CPU 实现软件转发，从而物理接口卡 2133 中不需要网络处理器 2132。

可选地，以太网设备 2100 包括多个接口板，例如以太网设备 2100 还包括接口板 2140，接口板 2140 包括：中央处理器 2141、网络处理器 2142、转发表项存储器 2144 和物理接口卡 2143。接口板 2140 中各部件的功能和实现方式与接口板 2130 相同或相似，在此不再赘述。

- 40 可选地，以太网设备 2100 还包括交换网板 2120。交换网板 2120 也可以称为交换网板单元（switch fabric unit，SFU）。在网络设备 2100 有多个接口板的情况下，交换网板 2120 用于完成各接口板之间的数据交换。例如，接口板 2130 和接口板 2140 之间可以通过交换网板 2120 通信。

- 45 主控板 2110 和接口板耦合。例如。主控板 2110、接口板 2130 和接口板 2140，以及交换网板 2120 之间通过系统总线与系统背板相连实现互通。在一种可能的实现方式中，主控板 2110 和接口板 2130 及接口板 2140 之间建立进程间通信协议（inter-process communication，IPC）通道，主控板 2110 和接口板 2130 及接口板 2140 之间通过 IPC 通道进行通信。

在逻辑上，以太网设备 2100 包括控制面和转发面，控制面包括主控板 2110 和中央处理器 2111，

转发面包括执行转发的各个组件，比如转发表项存储器 2134、物理接口卡 2133 和网络处理器 2132。控制面执行路由器、生成转发表、处理信令和协议报文、配置与维护网络设备的状态等功能，控制面将生成的转发表下发给转发面，在转发面，网络处理器 2132 基于控制面下发的转发表对物理接口卡 2133 收到的报文查表转发。控制面下发的转发表可以保存在转发表项存储器 2134 中。在有些实施例中，

- 5 控制面和转发面可以完全分离，不在同一网络设备上。
- 值得说明的是，主控板可能有一块或多块，有多块的时候可以包括主用主控板和备用主控板。接口板可能有一块或多块，网络设备的数据处理能力越强，提供的接口板越多。接口板上的物理接口卡也可以有一块或多块。交换网板可能没有，也可能有一块或多块，有多块的时候可以共同实现负载分担冗余备份。在集中式转发架构下，网络设备可以不需要交换网板，接口板承担整个系统的业务数据的处理功能。在分布式转发架构下，网络设备可以有至少一块交换网板，通过交换网板实现多块接口板之间的数据交换，提供大容量的数据交换和处理能力。所以，分布式架构的网络设备的数据接入和处理能力要大于集中式架构的网络设备。可选地，网络设备的形态也可以是只有一块板卡，即没有交换网板，接口板和主控板的功能集成在该一块板卡上，此时接口板上的中央处理器和主控板上的中央处理器在该一块板卡上可以合并为一个中央处理器，执行两者叠加后的功能，这种形态网络设备的数据交换和处理能力较低（例如，低端交换机或路由器等网络设备）。具体采用哪种架构，取决于具体的组网部署场景，此处不做任何限定。

在具体实施例中，以太网设备 2100 对应于上述图 15 所示的以太网设备。

- 例如，以太网设备 2100 可以为图 15 所示的以太网设备为交换机、路由器等以太网转发设备。此时，执行图 7 所示的方法的以太网芯片可以位于主控板 2110 中，或者也可位于接口板 2130 或接口板 2140 中，并且，该以太网芯片具体可位于物理接口卡 2133 或 2143 中。

如图 17 所示，本申请实施例还提供了一种通信系统 3000，该报文处理系统包括：发送端设备和 3001 和接收端设备 3002。可选地，接收端设备 3002 执行图 7 所示的数据流的解交织方法。

本申请实施例还提供了一种计算机可读存储介质，存储介质中存储有至少一条指令，指令由处理器加载并执行，以使计算机实现如上任一的数据发送方法或数据接收方法。

- 25 本申请实施例还提供了一种计算机程序（产品），当计算机程序被计算机执行时，可以使得处理器或计算机执行上述方法实施例中对应的各个步骤和/或流程。

本申请实施例还提供了一种芯片，包括处理器，用于从存储器中调用并运行存储器中存储的指令，使得安装有芯片的通信设备执行如上任一的数据发送方法或数据接收方法。

- 30 本申请实施例还提供另一种芯片，包括：输入接口、输出接口、处理器和存储器，输入接口、输出接口、处理器以及存储器之间通过内部连接通路相连，处理器用于执行存储器中的代码，当代码被执行时，处理器用于执行如上任一的数据发送方法或数据接收方法。

- 在上述实施例中，可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时，可以全部或部分地以计算机程序产品的形式实现。计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行计算机程序指令时，全部或部分地产生按照本申请的流程或功能。计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。计算机指令可以存储在计算机可读存储介质中，或者从一个计算机可读存储介质向另一个计算机可读存储介质传输，例如，计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线（例如同轴电缆、光纤、数字用户线）或无线（例如红外、无线、微波等）方式向另一个网站站点、计算机、服务器或数据中心进行传输。计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包含一个或多个可用介质集成的服务器、数据中心等数据存储设备。可用介质可以是磁性介质，（例如，软盘、硬盘、磁带）、光介质（例如，DVD）、或者半导体介质（例如，固态硬盘（solid state disk））等。

- 本领域普通技术人员可以意识到，结合本文中所公开的实施例中描述的各方法步骤和模块，能够以软件、硬件、固件或者其任意组合来实现，为了清楚地说明硬件和软件的可互换性，在上述说明中已经按照功能一般性地描述了各实施例的步骤及组成。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。本领域普通技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本申请的范围。

本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成，也可以通

过程序来指令相关的硬件完成，该程序可以存储于一种计算机可读存储介质中，上述提到的存储介质可以是只读存储器，磁盘或光盘等。

当使用软件实现时，可以全部或部分地以计算机程序产品的形式实现。该计算机程序产品包括一个或多个计算机程序指令。作为示例，本申请实施例的方法可以在机器可执行指令的上下文中被描述，
5 机器可执行指令诸如包括在目标的真实或者虚拟处理器上的器件中执行的程序模块中。一般而言，程序模块包括例程、程序、库、对象、类、组件、数据结构等，其执行特定的任务或者实现特定的抽象数据结构。在各实施例中，程序模块的功能可以在所描述的程序模块之间合并或者分割。用于程序模块的机器可执行指令可以在本地或者分布式设备内执行。在分布式设备中，程序模块可以位于本地和远程存储介质二者中。

10 用于实现本申请实施例的方法的计算机程序代码可以用一种或多种编程语言编写。这些计算机程序代码可以提供给通用计算机、专用计算机或其他可编程的数据处理装置的处理器，使得程序代码在被计算机或其他可编程的数据处理装置执行的时候，引起在流程图和/或框图中规定的功能/操作被实施。程序代码可以完全在计算机上、部分在计算机上、作为独立的软件包、部分在计算机上且部分在远程计算机上或完全在远程计算机或服务器上执行。

15 在本申请实施例的上下文中，计算机程序代码或者相关数据可以由任意适当载体承载，以使得设备、装置或者处理器能够执行上文描述的各种处理和操作。载体的示例包括信号、计算机可读介质等等。

信号的示例可以包括电、光、无线电、声音或其它形式的传播信号，诸如载波、红外信号等。

20 机器可读介质可以是包含或存储用于或有关于指令执行系统、装置或设备的程序的任何有形介质。机器可读介质可以是机器可读信号介质或机器可读存储介质。机器可读介质可以包括但不限于电子的、磁的、光学的、电磁的、红外的或半导体系统、装置或设备，或其任意合适的组合。机器可读存储介质的更详细示例包括带有一根或多根导线的电气连接、便携式计算机磁盘、硬盘、随机存取存储器（RAM）、只读存储器（ROM）、可擦除可编程只读存储器（EPROM 或闪存）、光存储设备、磁存储设备，或其任意合适的组合。

25 所属领域的技术人员可以清楚地了解到，为了描述的方便和简洁，上述描述的系统、设备和模块的具体工作过程，可以参见前述方法实施例中的对应过程，在此不再赘述。

在本申请所提供的几个实施例中，应该理解到，所揭露的系统、设备和方法，可以通过其它的方式实现。例如，以上所描述的设备实施例仅仅是示意性的，例如，该模块的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个模块或组件可以结合或者可以集成到另一个系统，
30 或一些特征可以忽略，或不执行。另外，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口、设备或模块的间接耦合或通信连接，也可以是电的，机械的或其它的形式连接。

该作为分离部件说明的模块可以是或者也可以不是物理上分开的，作为模块显示的部件可以是或者也可以不是物理模块，即可以位于一个地方，或者也可以分布到多个网络模块上。可以根据实际的需要选择其中的部分或者全部模块来实现本申请实施例方案的目的。

35 另外，在本申请各个实施例中的各功能模块可以集成在一个处理模块中，也可以是各个模块单独物理存在，也可以是两个或两个以上模块集成在一个模块中。上述集成的模块既可以采用硬件的形式实现，也可以采用软件功能模块的形式实现。

该集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分，
40 或者该技术方案的全部或部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）执行本申请各个实施例中方法的全部或部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（read-only memory，ROM）、随机存取存储器（random access memory，RAM）、磁碟或者光盘等各种可以存储程序代码的介质。

45 本申请中术语“第一”“第二”等字样用于对作用和功能基本相同的相同项或相似项进行区分，应理解，“第一”、“第二”、“第 n”之间不具有逻辑或时序上的依赖关系，也不对数量和执行顺序进行限定。还应理解，尽管以下描述使用术语第一、第二等来描述各种元素，但这些元素不应受术语的限

- 制。这些术语只是用于将一元素与另一元素区别分开。例如，在不脱离各种示例的范围的情况下，第一图像可以被称为第二图像，并且类似地，第二图像可以被称为第一图像。第一图像和第二图像都可以是图像，并且在某些情况下，可以是单独且不同的图像。
- 5 还应理解，在本申请的各个实施例中，各个过程的序号的大小并不意味着执行顺序的先后，各过程的执行顺序应以其功能和内在逻辑确定，而不对本申请实施例的实施过程构成任何限定。
- 本申请中术语“至少一个”的含义是指一个或多个，本申请中术语“多个”的含义是指两个或两个以上，例如，多个第二报文是指两个或两个以上的第二报文。本文中术语“系统”和“网络”经常可互换使用。
- 10 应理解，在本文中对各种所述示例的描述中所使用的术语只是为了描述特定示例，而并非旨在进行限制。如在对各种所述示例的描述和所附权利要求书中所使用的那样，单数形式“一个(“a”, “an”)”和“该”旨在也包括复数形式，除非上下文另外明确地指示。
- 还应理解，本文中所使用的术语“和/或”是指并且涵盖相关联的所列出的项目中的一个或多个项目的任何和全部可能的组合。术语“和/或”，是一种描述关联对象的关联关系，表示可以存在三种关系，例如，A 和/或 B，可以表示：单独存在 A，同时存在 A 和 B，单独存在 B 这三种情况。另外，本
- 15 申请中的字符“/”，一般表示前后关联对象是一种“或”的关系。
- 还应理解，术语“包括”(也称“includes”、“including”、“comprises”和/或“comprising”)当在本说明书中使用指定存在所陈述的特征、整数、步骤、操作、元素、和/或部件，但是并不排除存在或添加一个或多个其他特征、整数、步骤、操作、元素、部件、和/或其分组。
- 20 还应理解，术语“若”和“如果”可被解释为意指“当...时”(“when”或“upon”)或“响应于确定”或“响应于检测到”。类似地，根据上下文，短语“若确定...”或“若检测到[所陈述的条件或事件]”可被解释为意指“在确定...时”或“响应于确定...”或“在检测到[所陈述的条件或事件]时”或“响应于检测到[所陈述的条件或事件]”。
- 应理解，根据 A 确定 B 并不意味着仅仅根据 A 确定 B，还可以根据 A 和/或其它信息确定 B。
- 25 还应理解，说明书通篇中提到的“一个实施例”、“一实施例”、“一种可能的实现方式”意味着与实施例或实现方式有关的特定特征、结构或特性包括在本申请的至少一个实施例中。因此，在整个说明书各处出现的“在一个实施例中”或“在一实施例中”、“一种可能的实现方式”未必一定指相同的实施例。此外，这些特定的特征、结构或特性可以任意适合的方式结合在一个或多个实施例中。
- 以上描述仅为本申请的可选实施例，并不用以限制本申请，凡在本申请的原则之内，所作的任何修改、等同替换、改进等，均应包含在本申请的保护范围之内。

权 利 要 求 书

1. 一种数据流的解交织方法，其特征在于，所述方法包括：
获取第一交织数据流，所述第一交织数据流为经过前向纠错 FEC 编码的数据流；
基于所述第一交织数据流的交织粒度，获得所述第一交织数据流对应的 N 条子数据流，所述 N 为
5 所述第一交织数据流的交织深度，所述 N 为大于 1 的正整数；
基于所述 N 条子数据流中至少两条子数据流的 FEC 码字边界，确定解交织位置；
基于所述解交织位置，根据所述第一交织数据流获得第一解交织数据流。
2. 根据权利要求 1 所述的方法，其特征在于，所述 N 条子数据流按照顺序排列，所述基于所述 N
条子数据流中至少两条子数据流的 FEC 码字边界，确定解交织位置，包括：
10 基于第一子数据流的 FEC 码字边界在第二子数据流的 FEC 码字边界之后，确定所述解交织位置为
所述第二子数据流的 FEC 码字边界；
其中，所述第一子数据流和所述第二子数据流为所述 N 条子数据流中的两条相邻子数据流。
3. 根据权利要求 2 所述的方法，其特征在于，所述第一子数据流的排列顺序在所述第二子数据流
之前。
- 15 4. 根据权利要求 2 或 3 所述的方法，其特征在于，所述第一子数据流的 FEC 码字边界与所述第
二子数据流的 FEC 码字边界相差 1 个符号。
5. 根据权利要求 1 所述的方法，其特征在于，所述 N 条子数据流按照顺序排列，所述基于所述 N
条子数据流中至少两条子数据流的 FEC 码字边界，确定解交织位置，包括：
20 基于所述 N 条子数据流中每条子数据流的 FEC 码字边界对齐，确定所述解交织位置为第一子数据
流的 FEC 码字边界；
其中，所述第一子数据流在所述 N 条子数据流中排列顺序最先。
6. 根据权利要求 1-5 任一项所述的方法，其特征在于，所述 N 条子数据流按照顺序排列，所述基
于所述解交织位置，根据所述第一交织数据流获得第一解交织数据流，包括：
25 基于所述解交织位置和所述 N 条子数据流的排列顺序，确定所述 N 条子数据流的解交织组合顺序；
基于所述解交织组合顺序，组合所述第一交织数据流中的数据，以获得所述第一解交织数据流。
7. 根据权利要求 1-6 任一项所述的方法，其特征在于，所述基于所述第一交织数据流的交织粒度，
获得所述第一交织数据流对应的 N 条子数据流，包括：
从所述第一交织数据流中选取数据流分发开始位置，基于所述第一交织数据流的交织粒度，将所
述第一交织数据流分发为 N 条子数据流；
30 其中，所述数据流分发开始位置为在所述第一交织数据流中随机选取的位置。
8. 根据权利要求 1-7 任一项所述的方法，其特征在于，所述基于所述第一交织数据流的交织粒度，
获得所述第一交织数据流对应的 N 条子数据流，包括：
基于所述第一交织数据流的交织粒度，将所述第一交织数据流的数据轮询分发到所述 N 条子数据
流中。
- 35 9. 根据权利要求 1-8 任一项所述的方法，其特征在于，所述方法还包括：
确定所述 N 条子数据流中每条子数据流的 FEC 码字边界。
10. 根据权利要求 9 所述的方法，其特征在于，所述确定所述 N 条子数据流中每条子数据流的 FEC
码字边界，包括：
根据 FEC 码字特征，确定所述 N 条子数据流中每条子数据流的 FEC 码字边界。
- 40 11. 根据权利要求 10 所述的方法，其特征在于，所述根据 FEC 码字特征，确定所述 N 条子数据
流中每条子数据流的 FEC 码字边界，包括：
在每条子数据流中，分别基于假设的 FEC 码字边界进行特征值验证，根据验证结果确定每条子数
据流的 FEC 码字边界。
12. 根据权利要求 10 或 11 所述的方法，其特征在于，所述 N 条子数据流的 FEC 码字之间为循环
移位关系，所述根据 FEC 码字特征，确定所述 N 条子数据流中每条子数据流的 FEC 码字边界，包括：
45 对获得的所述 N 条子数据流的 FEC 码字进行逆循环移位；

- 根据 FEC 码字特征，确定逆循环移位后的所述 N 条子数据流中每条子数据流的 FEC 码字边界。
13. 根据权利要求 9 所述的方法，其特征在于，所述确定所述 N 条子数据流中每条子数据流的 FEC 码字边界，包括：
- 5 确定所述 N 条子数据流中第三子数据流的 FEC 码字边界；
- 基于所述第三子数据流的 FEC 码字边界，确定所述 N 条子数据流中其他子数据流的 FEC 码字边界。
14. 根据权利要求 13 所述的方法，其特征在于，所述第三子数据流的 FEC 码字边界位于所述第三子数据流的第一列中，
- 所述基于所述第三子数据流的 FEC 码字边界，确定所述 N 条子数据流中其他子数据流的 FEC 码
- 10 字边界，包括：
- 所述其他子数据流的 FEC 码字边界分别位于所述其他子数据流的所述第一列中或所述第一列的相邻列中。
15. 根据权利要求 1-14 任一项所述的方法，其特征在于，所述第一交织数据流中不包括周期性的对齐数据段，所述周期性的对齐数据段包括对齐字、帧对齐字或导引数据。
- 15 16. 根据权利要求 1-15 任一项所述的方法，其特征在于，所述第一交织数据流的 FEC 编码方式为非级联码的 FEC 编码，或级联码的 FEC 编码。
17. 根据权利要求 1-16 任一项所述的方法，其特征在于，所述 N 为 4 或 8。
18. 根据权利要求 1-17 任一项所述的方法，其特征在于，所述第一交织数据流的交织类型为符号交织或卷积交织。
- 20 19. 根据权利要求 1-18 任一项所述的方法，其特征在于，所述方法由物理层 PHY 芯片，转发设备，或可插拔模块执行。
20. 根据权利要求 1-19 任一项所述的方法，其特征在于，所述第一交织数据流的通道速率大于等于 200 吉比特每秒 Gbps。
21. 一种以太网设备，其特征在于，所述以太网设备用于执行如权利要求 1-20 任一项所述的方法。
- 25 22. 一种芯片，其特征在于，所述芯片用于执行如权利要求 1-20 任一项所述的方法。
23. 一种通信系统，其特征在于，所述通信系统包括发送端设备和接收端设备，
- 其中，所述接收端设备用于执行如权利要求 1-20 任一项所述的方法。

30