(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105340208 A
(43)申请公布日 2016.02.17

(21)申请号 201480015656.9

(22)申请日 2014.01.17

(30)优先权数据
61/753,884 2013.01.17 US
61/818,106 2013.05.01 US
14/157,290 2014.01.16 US

(85)PCT国际申请进入国家阶段日
2015.09.15

(86)PCT国际申请的申请数据
PCT/US2014/011988 2014.01.17

(87)PCT国际申请的公布数据
WO2014/113636 EN 2014.07.24

(71)申请人 高通股份有限公司
地址 美国加利福尼亚

(72)发明人 M·G·卢比 L·C·明德

(74)专利代理机构 永新专利商标代理有限公司 72002
代理人 张立达 王英

(51)Int.Cl.
H04L 1/06(2006.01)
H04L 1/18(2006.01)

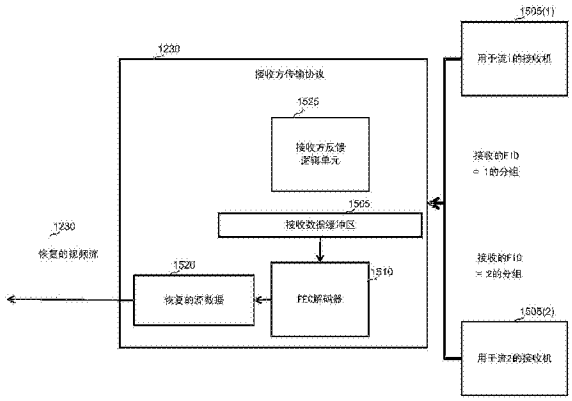
权利要求书4页 说明书30页 附图15页

(54)发明名称

用于多径流式传输的基于FEC的可靠传输控制协议

(57)摘要

客户端设备包括一个或多个处理器，所述一个或多个处理器被配置为：经由多个并行网络路径从服务器设备接收经前向纠错的数据；确定这些网络路径中的每个网络路径上的数据的丢失率；以及向服务器设备发送用于表示这些网络路径中的每个网络路径上的数据的丢失率的数据。另外地或替代地，客户端设备包括一个或多个处理器，所述一个或多个处理器被配置为：接收针对第一块的编码单元的第一集合，其中，与为了恢复第一块而需要的编码单元的最小数量相比，编码单元的第一集合包括更少的数量；在接收到编码单元的第一集合之后，接收针对第二块的编码单元的第二集合；以及在接收到编码单元的第二集合之后，接收编码单元的第三集合，其中编码单元的第三集合包括针对第一块的一个或多个编码单元。



CN 105340208 A

1. 一种计算机可读存储介质，其上存储有当被执行时使得客户端设备的处理器执行以下操作的指令：

经由多个并行网络路径，从服务器设备接收经前向纠错的数据；

确定所述网络路径中的每个网络路径上的所述数据的丢失率；以及

向所述服务器设备发送用于表示所述网络路径中的每个网络路径上的所述数据的所述丢失率的数据。

2. 根据权利要求 1 所述的计算机可读存储介质，其中，所述使得所述处理器发送所述用于表示所述丢失率的数据的指令包括使得所述处理器执行以下操作的指令：

发送用于标识所接收的编码单元所针对的多个块中的每个块的数据、所述块中的每个块需要的编码单元的数量、以及用于定义接收的针对与所述块中的每个块有关的网络分组的最高序列号的数据。

3. 根据权利要求 1 所述的计算机可读存储介质，其中，所述使得所述处理器发送所述用于表示所述丢失率的数据的指令包括使得所述处理器执行以下操作的指令：

针对所述并行网络路径中的每个网络路径，发送用于标识接收的针对经由各自的并行网络路径接收的分组流的最高序列号的数据。

4. 根据权利要求 1 所述的计算机可读存储介质，其中，所接收的数据包括针对一个数据块的多个编码单元，所述计算机可读存储介质还包括使得所述处理器执行以下操作的指令：

基于所述用于表示所述网络路径中的每个网络路径上的所述数据的所述丢失率的数据的聚合，接收针对所述数据块的一个或多个另外的编码单元。

5. 根据权利要求 1 所述的计算机可读存储介质，其中，所接收的数据 包括针对一个数据块的多个编码单元，所述计算机可读存储介质还包括使得所述处理器执行以下操作的指令：

为所述服务器设备计算所述块需要的另外的编码单元的数量；以及

向所述服务器设备发送用于表示另外的编码单元的所述数量的数据。

6. 根据权利要求 5 所述的计算机可读存储介质，其中，所述用于表示另外的编码单元的所述数量的数据包括：所述用于表示所述网络路径中的每个网络路径上的所述数据的所述丢失率的数据。

7. 根据权利要求 5 所述的计算机可读存储介质，其中，所述使得所述处理器计算另外的编码单元的所述数量的指令包括使得所述处理器执行以下操作的指令：

基于所述块的大小和所接收的针对所述块的编码单元的大小，计算为了恢复所述块而需要的编码单元的数量；以及

将另外的编码单元的所述数量计算成为了恢复所述块而需要的编码单元的所述数量与所接收的编码单元的所述数量之间的差值。

8. 根据权利要求 1 所述的计算机可读存储介质，其中，所接收的数据包括针对一个数据块的多个编码单元，所述计算机可读存储介质还包括使得所述处理器执行以下操作的指令：

接收针对所述块的编码单元；

确定所述块是否是活动的；

当所述块不是活动的时，丢弃所述编码单元；以及

当所述块是活动的时，将所述块添加到针对所述块的编码单元的集合中，以用于恢复所述块。

9. 根据权利要求 1 所述的计算机可读存储介质，其中，所述使得所述处理器进行接收的指令包括使得所述处理器执行以下操作的指令：

接收针对多个块的编码单元，使得针对所述多个块的所述编码单元是彼此之间交织的。

10. 根据权利要求 9 所述的计算机可读存储介质，其中，所述使得所述处理器接收所述编码单元的指令包括使得所述处理器执行以下操作的指令：

在所述并行网络路径中的第一路径上接收针对第一块的第一编码单元；

在接收到所述第一编码单元之后，在所述第一路径上接收针对第二块的第二编码单元；以及

在接收到所述第二编码单元之后，在所述第一路径上接收针对所述第一块的第三编码单元。

11. 一种计算机可读存储介质，其上存储有当被执行时使得服务器设备的处理器执行以下操作的指令：

经由多个并行网络路径，向客户端设备发送经前向纠错的数据；

从所述客户端设备接收用于表示在所述网络路径中的每个网络路径上发送的所述数据的丢失率的数据；以及

基于所述用于表示所述丢失率的数据，修改在所述并行网络路径上针对后续数据传输发送的经前向纠错的数据的量。

12. 根据权利要求 11 所述的计算机可读存储介质，其中，所述使得所述处理器接收所述用于表示所述丢失率的数据的指令包括使得所述处理器执行以下操作的指令：

针对所述并行网络路径中的每个网络路径，接收用于标识接收的针对经由各自的并行网络路径发送的分组流的最高序列号的数据。

13. 根据权利要求 11 所述的计算机可读存储介质，其中，所述使得所述处理器接收所述用于表示所述丢失率的数据的指令包括使得所述处理器执行以下操作的指令：

接收用于标识被所述客户端设备接收的编码单元所针对的多个块中的每个块的数据、所述块中的每个块需要的编码单元的数量、以及用于定义 被所述客户端设备接收的针对与所述块中的每个块有关的网络分组的最高序列号的数据。

14. 根据权利要求 11 所述的计算机可读存储介质，其中，所发送的数据包括针对一个数据块的多个编码单元，所述计算机可读存储介质还包括使得所述处理器执行以下操作的指令：

基于所述用于表示在所述网络路径中的每个网络路径上发送的所述数据的所述丢失率的数据的聚合，发送针对所述数据块的一个或多个另外的编码单元。

15. 根据权利要求 11 所述的计算机可读存储介质，其中，所发送的数据包括针对一个数据块的多个编码单元，所述计算机可读存储介质还包括使得所述处理器执行以下操作的指令：

从所述客户端设备接收用于表示所述块需要的另外的编码单元的数量的数据；以及

向所述客户端设备发送针对所述块的另外的编码单元的所述数量。

16. 根据权利要求 15 所述的计算机可读存储介质，其中，所述用于表示另外的编码单元的所述数量的数据包括：所述用于表示在所述网络路径中的每个网络路径上发送的所述数据的所述丢失率的数据。

17. 根据权利要求 11 所述的计算机可读存储介质，其中，所发送的数据包括针对一个数据块的多个编码单元，所述计算机可读存储介质还包括使得所述处理器执行以下操作的指令：

从所述客户端设备接收用于表示接收的编码单元的数量的数据；

基于所述块的大小和针对所述块的所述接收的编码单元的大小，计算为了恢复所述块而需要的编码单元的数量；

将另外的编码单元的数量计算成为了恢复所述块而需要的编码单元的所述数量与接收的编码单元的所述数量之间的差值；以及

向所述客户端设备发送另外的编码单元的所述数量。

18. 根据权利要求 11 所述的计算机可读存储介质，其中，所述使得所述处理器进行发送的指令包括使得所述处理器执行以下操作的指令：

发送针对多个块的编码单元，使得针对所述多个块的所述编码单元是彼此之间交织的。

19. 根据权利要求 18 所述的计算机可读存储介质，其中，所述使得所述处理器发送所述编码单元的指令包括使得所述处理器执行以下操作的指令：

在所述并行网络路径中的第一路径上发送针对第一块的第一编码单元；

在发送了所述第一编码单元之后，在所述第一路径上发送针对第二块的第二编码单元；以及

在发送了所述第二编码单元之后，在所述第一路径上发送针对所述第一块的第三编码单元。

20. 根据权利要求 11 所述的计算机可读存储介质，还包括使得所述处理器执行以下操作的指令：

在整个的数据块可用于所述处理器之前，发送针对所述数据块的编码单元。

21. 根据权利要求 11 所述的计算机可读存储介质，还包括使得所述处理器执行以下操作的指令：

发送针对第一块的编码单元的第一集合，其中，与为了恢复所述第一块而需要的编码单元的最小数量相比，所述编码单元的第一集合包括更少的数量；

在发送了所述编码单元的第一集合之后，发送针对第二块的编码单元的第二集合；以及

在发送了所述编码单元的第二集合之后，发送编码单元的第三集合，其中，所述编码单元的第三集合包括针对所述第一块的一个或多个编码单 元。

22. 根据权利要求 21 所述的计算机可读存储介质，其中，所述使得所述处理器发送所述编码单元的第一集合的指令包括使得所述处理器执行以下操作的指令：

在完全地形成所述第一块之前，发送所述编码单元的第一集合。

23. 根据权利要求 21 所述的计算机可读存储介质，其中，所述第一块和所述第二块包

括用于媒体内容的多个数据块中的块，其中，所述使得所述处理器发送所述编码单元的第一集合、所述编码单元的第二集合和所述编码单元的第三集合的指令包括使得所述处理器执行以下操作的指令：

经由在其上发送所述多个数据块的多个并行网络路径中的路径，发送所述编码单元的第一集合、所述编码单元的第二集合和所述编码单元的第三集合。

24. 根据权利要求 23 所述的计算机可读存储介质，还包括使得所述处理器执行以下操作的指令：

从所述客户端设备接收用于表示所述网络路径中的每个网络路径上的所述数据的丢失率的数据。

25. 根据权利要求 21 所述的计算机可读存储介质，还包括使得所述处理器执行以下操作的指令：

在接收到所述编码单元的第一集合之后，接收反馈数据的第一集合，其中，所述反馈数据的第一集合包括：用于标识所述第一块的数据、用于指示为了恢复所述第一块而需要的编码单元的数量的数据、以及用于定义接收的针对与所述第一块有关的网络分组的最高序列号的数据；以及

在接收到所述编码单元的第二集合之后，接收反馈数据的第二集合，其中，所述反馈数据的第二集合包括：用于标识所述第二块的数据、用于指示为了恢复所述第二块而需要的编码单元的数量的数据、以及用于定义接收的针对与所述第二块有关的网络分组的最高序列号的数据。

用于多径流式传输的基于 FEC 的可靠传输控制协议

[0001] 本申请要求享有于 2013 年 1 月 17 日提交的美国临时申请序列号 No.61/753,884 以及于 2013 年 5 月 1 日提交的美国临时申请序列号 No.61/818,106 的利益，上述每份申请的全部内容以引用方式并入本文。

技术领域

[0002] 本公开内容涉及媒体数据的传输。

背景技术

[0003] 设备可以使用多个路径和低延时处置，通过数据通信网络来执行数据在端系统之间的快速传输。很多数据通信系统和高层数据通信协议提供可靠数据传输的便利通信提取，并提供速率控制，即，它们基于网络状况自动地调整它们的分组传输速率。诸如普遍存在的传输控制协议 (TCP) 之类的依据较低层的分组化数据传输的它们的传统底层实施方式遭受以下状况中的至少一种状况发生 :(a) 发送方和接收方之间的连接具有较大的往返时间 (RTT) ;(b) 数据的量很大，并且网络遭受突发和瞬时丢失。

[0004] 一种广泛使用的可靠传输协议是传输控制协议 (TCP)。TCP 是一种普遍使用的具有确认机制的点到点分组控制方案。当发送方和接受方之间存在很少的丢失，并且发送方和接受方之间的 RTT 很小时，TCP 很好地适用于一对一的可靠通信。然而，当甚至存在非常少的丢失时，或者当发送方和接受方之间存在任何显著的延时时，TCP 的吞吐量将急剧地下降。

[0005] 通过使用 TCP，发送方发送有序分组，接受方对每个分组的接收进行确认。如果一个分组丢失，则不向发送方发送确认，并且基于后续接收的分组的接收或者基于超时，发送方将重新发送该分组。在使用诸如 TCP 之类的协议的情况下，这种确认范式允许在不具有整体失败的情况下，发生分组丢失，这是由于响应于确认的缺失或者响应于来自接受方的显式请求，可以仅对丢失的分组进行重传。

[0006] TCP 提供了可靠性控制和速率控制二者。也就是说，实现 TCP 的设备确保将所有原始数据都传送到接收方，并基于诸如拥塞和分组丢失之类的网络状况来自动地调整分组传输速率。在使用 TCP 的情况下，可靠性控制协议和速率控制协议是相互缠绕而不可分离的。此外，根据增加的 RTT 和分组丢失的 TCP 的吞吐量性能远不是最优的。

发明内容

[0007] 概括地说，本发明描述了与应用于在多个并行网络路径上发送的数据的前向纠错 (FEC) 有关的技术。在一个例子中，可以通过把要传输的数据组织成数据块，可靠地从发送方向接收方传输数据，其中，每个数据块包括多个编码单元。可以在多个路径上从发送方向接收方发送针对第一数据块的编码单元，即，第一数据块的一些编码单元在一个路径上进行发送，而第一数据块的其它编码单元在第二路径上进行发送等等。发送方可以检测接收方对编码单元的接收的确认。在发送方处，可以检测接收方从针对第一数据块的已发送

的编码单元中接收到第一数据块的足够的编码单元，以在接收方处恢复该第一数据块的概率，并且可以将该概率与门限概率进行测试比对，以确定是否满足了预定的测试条件。在测试的步骤之后，并在发送方接收到接收方处对第一数据块的恢复的确认之前，当满足预定的测试条件时，发送方可以在多个路径上从该发送方发送第二数据块的编码单元。在一些例子中，在已经发送了足够数量的第一块的编码单元以便接收方恢复第一块之前，发送方可以发送第二块的编码单元，因而发送方可以在发送了针对第二块的至少一些编码单元之后，发送针对第一块的另外的编码单元。此外，在一个块完全形成之前，发送方就可以向接收方发送该块的编码单元。在一些例子中，预定的测试条件是将所述概率与门限概率进行比较，当该概率大于门限概率时，确定满足了预定的测试条件。

[0008] 在一些例子中，随着每个块的生成，发送方还可以动态地确定各个块的大小或者持续时间，确定在生成这些块中的数据时所按照的速率、以及确定在所述多个路径中的各个路径上向接收方发送编码单元时所按照的速率。

[0009] 在一个例子中，一种计算机可读存储介质上存储有指令，其中当这些指令被执行时，使得客户端设备的处理器经由多个并行网络路径从服务器设备接收经前向纠错的数据，确定这些网络路径中的每个网络路径上的数据的丢失率，以及向服务器设备发送用于表示这些网络路径中的每个网络路径上的数据的丢失率的数据。

[0010] 在另一例子中，一种计算机可读存储介质上存储有指令，其中当这些指令被执行时，使得服务器设备的处理器经由多个并行网络路径，向客户端设备发送经前向纠错的数据，从客户端设备接收用于表示在所述多个网络路径中的每个网络路径上发送的数据的丢失率的数据，以及基于用于表示所述丢失率的所述数据，修改在所述并行网络路径上针对后续数据传输发送的经前向纠错的数据的量。

[0011] 在附图和下面的说明书中阐述了一个或多个例子的细节。根据该描述和附图，以及根据权利要求书，其它特征、目的和优点将变得显而易见。

附图说明

[0012] 图 1 是可以使用本公开内容的技术的示例性网络、发送方端系统和接收方端系统的框图。

[0013] 图 2 是模块化可靠传输协议架构以及使用这样的协议进行操作的相关系统的说明。

[0014] 图 3 是发送方的基于 FEC 的可靠性控制协议架构以及使用这样的协议进行操作的相关系统的例子。

[0015] 图 4 是接收方的基于 FEC 的可靠性控制协议架构以及使用这样的协议进行操作的相关系统的例子。

[0016] 图 5 示出了可以由实现 TF 可靠性控制协议的系统使用的一组可能的格式。

[0017] 图 6 是描绘了实现发送方 TF 可靠性控制协议的系统的逻辑的流程图。

[0018] 图 7 是描绘了实现接收方 TF 可靠性控制协议的系统的逻辑的流程图。

[0019] 图 8 是活动块的说明。

[0020] 图 9 是可以由交织式可靠性控制协议使用的一组可能的格式的说明。

[0021] 图 10 是实现基本发送方交织式可靠性控制协议的系统的逻辑的说明性例子。

[0022] 图 11 是实现基本接收方交织式可靠性控制协议的系统的逻辑的说明性例子。

[0023] 图 12 是一种多径流式传输系统的框图。

[0024] 图 13 示出了可以由实现多径可靠性控制协议的多径流式传输系统使用的一组可能的数据格式。

[0025] 图 14 是一种多径流式传输发送方的框图。

[0026] 图 15 是一种多径流式传输接收方的框图。

[0027] 图 16 描绘了在多径的基于 FEC 的可靠性传输控制方法的操作期间，源块的不同分类的快照。

具体实施方式

[0028] 很多研究人员的研究已经表明，当使用 TCP 时，数据传送的吞吐量（以分组每秒为单位）与以下二者的乘积成反比：RTT（以秒为单位）和端到端连接上的丢失率的平方根。例如，美国和欧洲之间的典型端到端地面连接具有 200 毫秒的 RTT 和平均 2％的分组丢失率，并且 IP 分组的大小通常大约为 10 千比特。在这些状况下，TCP 连接的吞吐量是大约每秒最多 300-400 千比特 (kbps)，而无论端到端之间有多大的可用带宽。这种情形在卫星链路上更加严重，其中在卫星链路上，除了具有高 RTT 之外，还由于各种大气效应而丢失信息。

[0029] 作为另一例子，已知移动网络 (3G 或 LTE 网络）中的移动设备经历较大的 RTT、RTT 的动态波动、以及可用带宽的动态波动。移动设备可能因为各种各样的原因而具有这些体验，所述原因包括：移动设备在小区内或者移动穿过小区边界的位置改变，这可能造成覆盖的波动；由于其它移动设备靠近或者离开向该移动设备提供覆盖的小区而造成的网络负载的变化；以及各种其它原因。TCP 在这些类型的状况之下具有较差性能的主要原因在于：TCP 使用的速率控制协议在这些状况下不能很好地工作，例如，即使可用带宽在较短的时间段内较高，TCP 协议也不能够足够快速地反应来增加其传输速率以在该可用带宽再次降低之前充分利用该较高的可用带宽。

[0030] 由于 TCP 使用的可靠性控制协议和速率控制协议是不可分离的，因此这意味着整个 TCP 协议不适用于这些状况。一种尝试克服 TCP 的这些限制的方式是在单独的路径上使用多个 TCP 连接，以进一步增加聚合的端到端吞吐量。此外，不同的应用对于传输的要求也不同，然而 TCP 相当普遍地用于所有网络状况下的各种应用，因而在很多情形下导致较差的性能。

[0031] 例如，对于实时视频流式传输应用而言，视频可以在移动设备上的视野中生成，并可能使用通过 WiFi 系留或者连接到该生成移动设备的多个移动设备，在多个 TCP 连接上，可能在不同的 3G 或 4G/LTE 连接上，流式传输到将重构该原始视频流的接收设备。然而，由于可用带宽的波动和 RTT 的波动，这些多个 TCP 连接可能仍不能充分地利用可用带宽。对于这样的流式传输应用而言，实时流式传输应用对端到端延迟的要求可能是非常严格的，因而存在如下进一步的复杂度和要求：该流包括一系列的数据块，以及需要接收在不同的 TCP 连接上发送的针对各个数据块的足够的编码单元，以使得在接收设备处得以对该流的该数据块进行重构，以及通常在接收设备处以具有以下两个时间点之间的最小可能延迟来顺序地消费或回放这些数据块：当在发送方处使各个数据块（部分或全部）可用时以及当

在接收设备处重构该数据块并使其可用于回放或消费时。这些要求可能使基于 TCP 的解决方案的能力受到最慢 TCP 连接的约束，故基于 TCP 的总解决方案可能是非常逊色的。

[0032] 本公开内容包括改进的可靠性控制协议和速率控制协议的描述，这两种协议可以独立地使用，并且相同的可靠性控制协议可以与各种不同的速率控制协议一起使用，使得所选择的实际速率控制协议可以是基于应用要求和该应用所运行的网络状况。Micah Adler、Yair Bartal、John Byers、Michael Luby、Danny Raz 于 1997 年 6 月在第五届以色列计算和系统理论研讨会 (the Fifth Israeli Symposium on Theory of Computing and Systems) 的议程中发表的论文“A Modular Analysis of Network Transmission Protocols”（下文称为“Adler”，并以引用方式并入本文）介绍了用于构建传输协议的模块化方法，其主张将可靠传输协议划分成独立的可靠性控制协议和速率控制协议。

[0033] 对于任何可靠性控制协议而言，其性能的两项主要测量值是要求多少缓冲以及其“有效吞吐量”。在发送方和接收方二者处的可靠性控制协议中引入了缓冲。例如，发送方处的缓冲发生于：当数据被首次发送之后就对该数据进行缓冲，直到发送方获得了在接收方处已经接收到该数据的确认为止。接收方处的缓冲由于类似的原因而发生。对缓冲感兴趣的原因有两点：(1) 其直接影响发送方和接收方可靠性控制协议使用多少存储器；(2) 其直接影响发送方和接收方可靠性控制协议引入多大延时。有效吞吐量被定义为：将要传送的数据的大小除以在该传送期间在接收方端系统处接收到的已发送数据的量后得到的值。例如，如果在分组中发送的用于传送原始数据的数据的量是该原始数据的大小，则有效吞吐量＝ 1.0，并且如果没有发送冗余数据，则可以实现有效吞吐量＝ 1.0。

[0034] Adler 概述了在很大程度上独立于所使用的速率控制协议的可靠性控制协议，其在下文被称为“无编码 (no-code) 可靠性控制协议”。无编码可靠性控制协议在某些方面类似于 TCP 中嵌入的可靠性控制协议，在该意义上，原始数据被划分成一些块，并且每个块在分组的有效载荷中发送，然后需要接收各个块的精确副本以确保可靠传送。使用无编码可靠性控制协议的问题在于：尽管有效吞吐量是最优的（基本上等于一），但当存在分组丢失时，无编码可靠性控制协议引入的缓冲可能非常大。Adler 证明了无编码可靠性控制协议处于不使用编码来传输数据的可靠性控制协议之中的最佳常数因子内，在该意义上，该协议具有最佳的有效吞吐量，并且依据使发送方和接收方处所需要缓冲的量最小化而言，可证明是处于最佳常数因子内的。

[0035] 在可靠性控制协议中已经使用的一种技术是前向纠错 (FEC) 编码，例如里德－所罗门编码 (Reed-Solomon code) 或 Tornado 编码或者链式反应编码（其是信息附加编码）。使用 FEC 编码，将原始数据划分成比分组的有效载荷更大的块，然后根据这些块来生成编码单元，并在分组中发送这些编码单元。与不使用编码的可靠性控制协议相比，这种方法的一项基本优点在于：反馈可能是更简单的和更不频繁的，即，对于每个块而言，接收方仅仅只需要向发送方指示接收到的编码单元的数量，而不用确切地列出接收到哪些编码单元。此外，与原始数据块的长度相比，聚合生成和发送更多的编码单元的能力，在设计可靠性控制协议时是一种强大的工具。

[0036] 诸如里德－所罗门编码或 Tornado 编码之类的纠删编码针对固定长度的块生成固定数量的编码单元。例如，对于包括 B 个输入单元的块而言，可以生成 N 个编码单元。这 N 个编码单元可以包括 B 个原始输入单元和 N-B 个冗余单元。如果存储装置准许，则发送方可以

只对针对各个块的编码单元的集合进行一次计算，并使用轮播协议（carousel protocol）来发送这些编码单元。

[0037] 使用一些 FEC 编码的一个问题在于：它们要求过多的计算能力或存储器来进行操作。另一问题在于：必须在编码过程之前确定需要的编码单元的数量。如果分组的丢失率估计过高的话，这可能导致低效率，而如果分组的丢失率估计过低的话，则可能导致失败。

[0038] 对于传统 FEC 编码而言，可以生成的可能的编码单元的数量与一个块被划分成的输入单元的数量具有相同的数量级。通常，但不完全是，这些编码单元中的大部分或者全部在发送步骤之前的预处理步骤中被生成。这些编码单元具有如下属性：可以根据这些编码单元的任何子集来重新生成输入单元，所述编码单元在长度上等于原始块或者比原始块稍微更长。

[0039] 美国专利 No. 6,307,487（下文称为“Luby I”，并以引用方式并入本文）中描述的链式反应解码可以提供一种解决上述问题的前向纠错的形式。对于链式反应编码而言，可以生成的可能的编码单元的池在数量级上大于输入单元的数量，并且可以非常迅速地生成根据该可能池的随机或者伪随机选择的编码单元。对于链式反应编码而言，可以与发送步骤同时地，在“按需的”基础上飞速生成编码单元。链式反应编码允许根据随机或伪随机生成的编码单元集合的子集来重新生成内容的全部输入单元，所述编码单元在长度上比原始内容稍微更长。

[0040] 诸如美国专利 No. 6,320,520、6,373,406、6,614,366、6,411,223、6,486,803 和美国专利公开 No. 2003/0058958（下文称为“Shokrollahi I”）之类的其它文档描述了各种链式反应编码方案，并以引用方式并入本文。

[0041] 使用链式反应编码的发送方可以针对正在发送的每个块，持续地生成编码单元。可以通过用户数据报协议（UDP）单播或者 UDP 多播（如适用）将这些编码单元发送给接受方。假定每个接受方都装备有解码单元，该解码单元对分组中接收的合适数量的编码单元进行解码以获得原始块。

[0042] 现有的简单的基于 FEC 的可靠性控制协议（下文称为“TF 可靠性控制协议”）发送针对给定数据块的编码单元，直到从接收方接收到已经接收了足够的编码单元来恢复该块的确认为止，然后发送方移动到下一个块。

[0043] 假设 RTT 是从当发送方发送分组时开始，到发送方从接收方接收到该分组已经到达的确认为止的秒的数量，以及假设 R 是发送方的当前发送速率（以分组 / 秒为单位），以及假设 B 是块的大小（以分组为单位）。使用 TF 可靠性控制协议，包含为了恢复一个块而需要的最后分组之后发送的针对该块的编码单元的无用分组的数量是 $N = R*RTT$。因而，$f = N/(B+N)$ 部分的发送的分组被浪费了，因而有效吞吐量至多是 1-f。例如，如果 R = 1,000 分组 / 秒，RTT = 1 秒，并且 B = 3,000 个分组，则 f = 0.25，即，浪费了 25％的所接收的分组。因此，在这个例子中，有效吞吐量是不足的 0.75（与 1.0 的最大可能有效吞吐量相比）。

[0044] 在这个例子中，还应当注意的是，块的大小 B 连同速率 R 一起意味着简单的基于 FEC 的可靠性控制协议所引入的延时至少是 4 秒（发送每个块总共需 4 秒钟），并且要求缓冲至少一个块，即 3,000 个数据分组。此外，为了增加有效吞吐量，要求增加缓冲，或者相反地，为了减少缓冲，则要求减少有效吞吐量。

[0045] 以 Luby 等人名义申请的美国专利 No. 7,447,235（下文称为“Luby II”）描述了改

进的基于 FEC 的可靠性协议。然而，用于在多个路径上进行流式传输的改进的可靠性控制协议是期望的。此外，提供可以与改进的可靠性控制协议进行组合的相应的速率控制协议，以获得使可靠性和有效吞吐量最大化，并使端到端延时最小化的适合于在多个路径上进行流式传输的传输协议是期望的。

[0046] 在根据本公开内容的例子中，可以使用用于多径传输的交织式可靠性控制协议来提供相对于 TCP、TF 可靠性控制协议和无编码可靠性控制协议、以及 Luby II 中描述的基于 FEC 的可靠性协议的改进。此外，介绍了可以与改进的可靠性控制协议进行组合的改进的速率控制协议和生成速率协议，以提供使可靠性和有效吞吐量最大化，并使端到端延时最小化的适合于在多个路径上进行流式传输的传输协议。

[0047] 在使用可靠性控制协议的情况下，数据块被发送为从发送方到接收方的一系列的编码单元，并且接收方对这些编码单元或者块的恢复进行确认，从而使发送方得以确定接收方是否接收到该数据，并当没有接收到时，对该数据进行重传，或者发送可用于恢复所接收的数据的其它数据。一些交织式可靠性控制协议的一种属性在于以交织的方式来发送针对不同块的编码单元。交织式可靠性控制协议具有如下属性：当与实质上任何速率控制协议进行组合时，它们提供高效的可靠数据传输，其使端系统处的缓冲（和随之发生的延时）最小化，并使该传输的有效吞吐量最大化。

[0048] 根据本公开内容的技术，交织式可靠性控制协议可以与合适的速率控制协议一起使用，以确保数据的可靠传送，同时维持高吞吐量，即使当存在高丢失率时和 / 或当存在较大的 RTT 时。例如，速率控制协议可以与按照固定的速率进行发送一样简单，并且交织式可靠性控制协议将保证数据按照等于固定速率乘以成功到达的分组的比例后的速率进行传送，同时使该传送期间的缓冲和延时最小化。

[0049] 作为由此处介绍的交织式可靠性控制协议所提供的量化性改进的例子，假使速率控制协议按照每秒 R 个分组的固定速率来发送分组，发送方和接收方之间的往返时间是 RTT 秒，因而 N = R*RTT 是在途的未被确认分组的数量。对于无编码可靠性控制协议而言，发送方处的总缓冲区大小 B 至少是 N*ln(N)，以及有效吞吐量近似是 1.0，并且在所需要的缓冲的量和有效吞吐量之间不存在可能的其它权衡点。此处，ln(x) 被定义为 x 的自然对数。在使用 TF 可靠性控制协议的情况下，发送方处的总缓冲区大小至少是 B，并且有效吞吐量近似是 B/(B+N)，其中 B 是选择的块大小（以分组为单位），并且可以对 B 进行选择以在要求的缓冲与有效吞吐量间进行权衡。相比而言，对于诸如 Luby II 中所描述的那些协议之类的交织式可靠性控制协议而言，发送方处的总缓冲区大小至多是 B，并且有效吞吐量近似是 N/(N+X)，其中 X 是选择的用于在要求的缓冲与有效吞吐量间进行权衡的正整数参数，并且 B = N*(1+ln((N/X)+1)) 是以分组为单位的缓冲区大小。

[0050] 作为例子，如果速率 R 是 1,000 分组 / 秒，RTT 是一秒，则 N = 1,000 个分组。对于无编码可靠性控制协议而言，发送方处的缓冲区大小至少是 7,000 个分组。对于 TF 可靠性控制协议而言，如果将 B 选择为 4,000 个分组，则有效吞吐量近似是 0.80。对于 Luby II 中所描述的交织式可靠性控制协议而言，其中将 X 选择为 50，B = 4,000 个分组（与 TF 可靠性控制协议相同的值），并且有效吞吐量超过 0.95，即，至多 5％的已接收分组被浪费。因而，在这个例子中，与具有几乎相同的最优有效吞吐量的无编码可靠性控制协议相比，交织式可靠性控制协议要求少得多的缓冲，而对于具有相同数量的缓冲而言，其远远超过 TF 可

靠性控制协议的有效吞吐量，即，与 TF 可靠性控制协议浪费 25％的传输相比，交织式可靠性控制协议至多浪费 5％的传输。

[0051] 事实上，可以与交织式可靠性控制协议一起使用任何速率控制协议，以提供可靠传输协议，例如，按照固定速率进行发送，使用类似于 TCP 的基于窗的拥塞控制，使用诸如 TCP 友好速率控制 (TFRC) 之类的基于等式的拥塞控制协议，或者使用实质上任何其它速率控制协议（包括本文所介绍的速率控制协议）。

[0052] 下文的后续描述描述了可以适合于在多个独立的路径上，从发送方向接收方发送数据的速率控制协议，以及描述了可以与这些速率控制协议进行组合的增强型交织式可靠性控制协议。另外，还描述了生成速率协议，其可以用于确定应当生成多快的数据以便传输。最后，本描述描述了用于对这些协议进行组合，以提供适合于在多个独立的路径上，从发送方向接收方流式传输数据的总传输协议的方式的例子，该总传输协议使可靠性和有效吞吐量最大化，并使端到端延时最小化。

[0053] 在本描述中，可靠传输协议是以如下方式，通过基于分组的网络，从发送方端系统向接收方端系统可靠地传送数据的协议：即使当所发送的分组中的一些分组可能没有被接收时，也传送全部数据。

[0054] 图 1 是描绘了可靠传输协议可以在其上操作的网络 130、一组发送方端系统 100(1)-100(J) 和一组接收方端系统 160(1)-160(K) 的例子的概念图。发送方端系统 100 还可以被描述为服务器设备，以及接收方端系统 160 可以被描述为客户端设备。通常，这样的协议还包括用于调整分组发送速率的一些机制，其中这个发送速率可以取决于包括以下各项的各种因素：在其中构建该协议的应用、用户输入参数、以及发送方端系统和接收方端系统之间的网络状况。

[0055] 通常，诸如 TCP 之类的可靠传输协议涉及若干步骤。这些步骤包括用于端系统执行以下操作的方式：通告数据的可用性、发起向其它端系统的数据传送、沟通将要传送什么数据、以及执行该数据的可靠传送。存在用于端系统执行以下操作的各种标准方式：通告可用性、发起传送和沟通将要传送什么数据，例如，会话通告协议、会话发起协议等等。由于这些步骤是众所周知的，故此处不需要进一步详细地描述。

[0056] 分组数据的可靠传输包括：在传送过程中的每个点，决定要在分组中发送什么数据，以及按照什么速率来发送这些分组。在每个时间点所做的决定可以取决于从接收方端系统发送的反馈和其它因素。通常，在发送方端系统处，数据被呈现为数据流，并且可靠传输协议旨在以该流被发送时的相同顺序，可靠地将该流传送到接收方端系统。通常，在发起传送之前，该流的总长度是不知道的。

[0057] 本公开内容描述了用于可靠传输协议的模块化架构的例子。Adler 描述了如何将任何可靠传输协议视作为可靠性控制协议和速率控制协议的组合。可靠性控制协议是总传输协议中的在传送期间决定在各个分组中放置什么数据的那部分。速率控制协议决定何时发送各个数据分组。在很多传输协议中，可靠性控制协议和速率控制协议在操作时不可分离地缠绕在一起，即，这是 TCP 的情况。然而，即使这样的缠绕的协议可以概念性地划分成可靠性控制协议和速率控制协议，也仍然是这种情况。

[0058] Adler 通过独立地设计可靠性控制协议和速率控制协议，主张了可靠传输协议的设计。这样的方法的优点在于：相同的可靠性控制协议可以与各种速率控制协议一起使用，

因而相同的可靠性控制协议可以与适合于其中使用该总可靠传输协议的应用和网络状况的速率控制协议一起使用。用于该设计的这种模块化方法是非常有利的，这是因为相同的可靠性控制协议可以与不同的应用和网络环境中的速率控制协议的不同集合一起使用，因而避免了对用于每种应用和网络环境的整个可靠传输协议进行完全地重新设计。例如，TCP用于不同的网络环境中的各种应用，但由于如根据其速率控制协议所确定的其实现的较差吞吐量，因此对于这些应用和网络环境中的一些而言，其性能较差。不幸的是，因为在TCP架构中，可靠性控制协议和速率控制协议如此紧密地缠绕在一起，所以在TCP运行效果较差的那些情形中，不能简单地使用TCP内的不同速率控制协议来提高其吞吐量性能。

[0059] 在图1的例子中，发送方端系统100中的一个可以使用本公开内容的技术，以经由网络130向接收方端系统160中的一个传输数据。例如，发送方端系统100(1)可以使用本公开内容的技术中的一种或多种技术，向接收方端系统160(1)发送媒体数据。在一个例子中，发送方端系统100(1)可以通过网络130，在多个并行路径上向接收方端系统160(1)发送经前向纠错的媒体数据。这样的经前向纠错的媒体数据可以包括多个块，每个块可以是使用多个编码单元来进行FEC编码的。因而，对于每个块而言，发送方端系统100(1)可以向接收方端系统160(1)发送多个编码单元。

[0060] 接收方端系统160(1)可以发送用于表示这些并行网络路径中的每个网络路径上的丢失率的数据。下面参照图9和图13进一步详细地描述这样的数据的一个例子。通常，该数据可以指示：针对每个块，接收的编码单元的数量以及接收的最高序列号。另外地或替代地，该数据可以指示：针对通过各个并行网络路径的每个分组流，接收的最高序列号。类似地，发送方端系统100(1)可以接收由接收方端系统160(1)发送的关于这些丢失率的数据。发送方端系统100(1)可以使用这个关于丢失率的数据，以修改如何形成后续的编码单元，例如包括：如果丢失率比预期的要高，则包括另外的FEC数据；或者如果丢失率比预期的要低，则减少FEC数据的量。这种对包括在编码单元中的FEC数据的修改可以基于并行网络路径上的丢失率的聚合。

[0061] 另外地或替代地，发送方端系统100(1)可以在一个块被完全形成之前，就发送该块的编码单元。例如，假定当前块的大小为K个编码单元，则在当前块的全部K个编码单元都已经可用于发送方100(1)之前，发送方端系统100(1)就可以发送针对当前块的一个或多个编码单元。另外，在接收到针对当前块的K个编码单元之前，接收方端系统160(1)就可以接收后续块的一个或多个编码单元。

[0062] 图2是描绘了Adler中所主张的示例性模块化可靠传输协议架构的框图。发送方传输协议210被划分成发送方可靠性控制协议220和发送方速率控制协议230。发送方可靠性控制协议220确定在各个数据分组中发送什么内容，以及发送方速率控制协议230确定何时发送各个数据分组。发送方可靠性控制协议220可以在各个数据分组中放置另外的可靠性控制信息，接收方传输协议290内的接收方可靠性控制协议280可以使用该信息。

[0063] 发送方可靠性控制协议220还可以从接收方传输协议290内的相应接收方可靠性控制协议280接收可靠性控制信息250，其用于帮助确定在各个数据分组中发送什么内容。类似地，发送方速率控制协议230可以在各个数据分组中放置另外的速率控制信息，接收方传输协议290内的接收方速率控制协议270可以使用该信息。发送方速率控制协议230还可以从接收方传输协议290内的相应接收方速率控制协议270接收速率控制信息250，其

用于帮助确定何时发送各个数据分组。

[0064] 在发送方可靠性协议 220 和接收方可靠性协议 280 之间沟通的可靠性控制信息可以取决于诸如分组丢失率之类的各种因素，并且可以包含如稍后相当详细解释的各种信息。类似地，在发送方速率控制协议 230 和接收方速率控制协议 270 之间沟通的速率控制信息可以取决于诸如分组丢失率和测量的往返时间 (RTT) 之类的各种因素。此外，可靠性控制信息和速率控制信息可以交迭，在该意义上，可以使用在数据分组 240 中或者在反馈分组 250 中发送的信息以用于可靠性控制和速率控制二者。通常，从发送方传输协议 210 向接收方传输协议 290 发送的可靠性控制信息和速率控制信息可以与数据分组 240 中的数据一起发送，或者在单独的控制分组 240 中发送，或者是这二者。应当对这些协议进行设计，以使需要从发送方向接收方发送和从接收方向发送方发送的控制信息的量最小化。

[0065] 对于很多应用而言，数据被传送为流，即，当数据到达发送方端系统时，其将以与其到达发送方端系统相同的顺序，尽可能快速地可靠地被传送给接收方端系统。对于一些应用而言，应当使总传输协议所引入的延时最小化，例如，对于流式传输应用而言，或者对于交互式应用而言，其中在交互式应用的情况下，要在两个端系统之间尽可能快速地来回发送很少的数据突发。因而，应当使传输协议所引入的总延时最小化。

[0066] 通常，发送方可靠性控制协议 220 和接收方可靠性控制协议 280 均要求缓冲区来临时地存储数据。通常，在发送方可靠性控制协议 220 处缓冲的数据至少包括 ：该流中的发送方可靠性控制协议 220 尚未从接收方可靠性控制协议 280 接收到恢复确认的最早数据，一直到该流中的发送方可靠性控制协议 220 已经开始在数据分组中发送的最新数据。通常，接收方可靠性控制协议 280 处的缓冲区的大小至少是 ：该流中的从尚未被恢复的最早数据到其数据分组已经被接收到的最新数据的数据的量。

[0067] 发送方可靠性控制协议 220 的缓冲要求直接影响发送方可靠性控制协议 220 要求多少的临时存储空间，以及发送方可靠性控制协议 220 向总的可靠数据传送引入多大的延时。接收方可靠性控制协议 280 的缓冲要求具有类似的影响。因而，使发送方可靠性控制协议 220 和接收方可靠性控制协议 280 二者的缓冲要求最小化是重要的。

[0068] 可靠性控制协议确定在各个数据分组中发送什么内容。为了高效地利用端系统之间的连接，发送方可靠性控制协议 220 在分组中尽可能少地发送冗余数据是重要的，以便确保在接收方可靠性控制协议 280 处接收到的任何数据分组在恢复原始数据流的部分时都是有用的。可靠性控制协议的有效吞吐量被定义为 ：原始数据流的长度除以在该原始数据流的恢复期间，由接收方可靠性控制协议 280 接收到的数据分组的总长度后得到的值。有效吞吐量目标是 ：对于可靠性控制协议而言，获得 1.0 或者近似于 1.0 的有效吞吐量，在该情况下，接收最小量的数据来恢复原始数据流。在一些可靠性控制协议中，有效吞吐量可以小于 1.0，在该情况下，发送的数据分组中的一些被浪费。因而，设计可靠性控制协议，使得有效吞吐量尽可能靠近 1.0，以便高效地使用从发送方端系统行进到接收方端系统的数据分组所消耗的带宽是重要的。

[0069] 已经在可靠性控制协议中使用的一种解决方案在于诸如里德 - 所罗门编码或 Tornado 编码之类的前向纠错 (FEC) 编码，或者诸如 Luby I 或 Shokrollahi I 中所描述的那些编码之类的链式反应编码 (其是信息附加码)。原始数据被划分成比分组的有效载荷更大的块，然后根据这些块来生成编码单元，并在分组中发送这些编码单元。诸如里德 - 所

罗门编码或 Tornado 编码之类的纠删编码针对固定长度的块生成固定数量的编码单元。例如，对于包括输入单元的块而言，可以生成 N 个编码单元。这 N 个编码单元可以包括 B 个原始输入单元和 N-B 个冗余单元。

[0070] 基于 FEC 的可靠性控制协议是使用 FEC 编码的可靠性控制协议。图 3 是描绘了发送方的基于 FEC 的可靠性控制协议 220 的例子的框图，以及图 4 是描绘了接收方的基于 FEC 的可靠性控制协议 280 的例子的框图。发送方可靠性控制逻辑单元 310 将原始数据流划分成数据块 330，然后指示 FEC 编码器 320 生成针对各个块的编码单元。发送方可靠性控制逻辑单元 310 确定如何将编码单元和可靠性控制信息 340 传递给处置发送方速率控制协议 230 的设备，该设备还处置由图 4 中示出的接收方的基于 FEC 的可靠性控制逻辑单元 410 发送的可靠性控制信息 350。

[0071] 发送方可靠性控制逻辑单元 310 应当确保图 4 中示出的接收方的基于 FEC 的可靠性控制协议 280 接收到足够的编码单元，以确保每个块都被恢复。全部块可以具有基本相同的长度，或者块长度可以在传送期间根据各种参数动态地变化，所述参数包括 ：在使该数据流可用于发送方时所按照的速率、数据分组的发送速率、网络状况、应用要求和用户要求。

[0072] 假使给定的数据块在长度上是 B 个编码单元。对于一些 FEC 编码而言，为了恢复原始数据块而要求的编码单元的数量精确地是 B，而对于其它 FEC 编码而言，为了恢复原始数据块而要求的编码单元的数量稍微大于 B。为了简化基于 FEC 的可靠性控制协议的描述，假定 B 个编码单元足够用于恢复数据块，其中应当理解的是，可以在具有稍微减少的有效吞吐量和稍微增加的缓冲要求的情况下，使用要求 B 个以上的编码单元来对块进行解码的 FEC 编码。

[0073] 图 4 中的接收方可靠性控制逻辑单元 410 负责确保接收到 B 个编码单元，以便对数据块进行解码，然后使用 FEC 解码器 420 来恢复数据块 430。接收方可靠性控制逻辑单元 410 负责接收从发送方的基于 FEC 的可靠性控制协议 220 发送的编码单元和可靠性控制信息 340，以及生成和发送可靠性控制信息 350，其中该可靠性控制信息 350 最终被发送给发送方可靠性控制逻辑单元 310，并由发送方可靠性控制逻辑单元 310 进行处理。

[0074] TF 可靠性控制协议将数据流划分成大体上大小相等的块。总架构在于在任何时间点只存在一个活动数据块，并且发送方生成和发送针对该数据块的编码单元，直到其从接收方接收到用于指示足够的编码单元已经到达从而能重构该块的消息，在该时间点，发送方移动到下一个块。因而，生成并发送针对给定的块的全部编码单元，并且该块在生成和发送针对后续块的任何编码单元之前被恢复。

[0075] 图 5 是描绘了可以由 TF 可靠性控制协议使用的格式的可能集合的框图。在这个例子中，发送方数据格式描述了发送方 TF 可靠性控制协议在可向接收方 TF 可靠性控制协议发送编码单元和相应的可靠性控制信息时所采用的格式。这包括 ：块编号 510，其指示该编码单元是根据哪个块生成的 ；编码单元 ID 520，其指示如何根据该块生成该编码单元 ；以及编码单元 530，接收方 TF 可靠性控制协议内的 FEC 解码器可以使用其来恢复该块。接收方反馈格式描述了接收方 TF 可靠性控制协议在可向发送方 TF 可靠性控制协议发送可靠性控制信息时所采用的格式。这包括 ：块编号 540，其是当前块的块编号，接收方 TF 可靠性控制协议正在接收针对该当前块的编码单元以恢复该块 ；以及需要的编码单元 550，其是

接收方 TF 可靠性控制协议为了恢复该块而需要的另外的编码单元的数量。

[0076] 图 6 是描述了用于实现发送方 TF 可靠性控制协议的过程的例子的流程图。根据这个示例性过程，发送方设备（例如，图 1 的发送方端系统 100 中的一个）持续地检查，以判断是否到了发送发送方数据的时间（步骤 610），这由相应的发送方速率控制协议来确定。如果到了发送发送方数据的时间，则根据活动块来生成编码单元，并发送该发送方数据 (620)。用于该发送方数据的形式的例子是图 5 中示出的格式。该过程还持续地检查，以判断是否已经接收到接收方反馈 630。用于该接收方反馈数据的形式的例子是图 5 中示出的格式。如果存在接收方反馈，则对其进行处理以更新关于接收方需要多少另外的编码单元来恢复该活动块的信息。然后，进行检查以判断需要的编码单元的数量是否是零 640，如果是，则检查该数据流中的下一个块是否可用 650。如果其不可用，则准备下一个块，直到其就绪为止 660，然后接着，去激活当前活动块，并激活下一个块 670。通常，在发送当前活动块时，就可以准备下一个块。

[0077] 应当理解的是，本文所描述的每种协议都可以由设备或者由适当的处理器执行的软件或固件来实现。例如，实施方式可以使用诸如路由器和主计算机之类的网络设备来进行，也可以在无线发射机、转播发射机和其它无线设备上实现。本文所描述的协议可以用软件来实现，具有方法，和 / 或具有被配置为实现这样的协议的装置。

[0078] 图 7 是描述了用于实现接收方 TF 可靠性控制协议的示例性过程的流程图。根据该接收方 TF 可靠性控制协议，接收方设备（例如，接收方端系统 160（图 1）中的一个）可以持续地检查，以判断是否已经接收到发送方数据 710，该发送方数据采用图 5 中所示的发送方数据格式。如果接收到，则检查该发送方数据内的编码单元是否是根据该活动块 720。如果该编码单元不是根据该活动块，则丢弃该编码单元 760，因而这是浪费的发送方数据，这是由于其不可用于恢复任何块。

[0079] 如果该编码单元是根据该活动块，则将其添加到针对该活动块的已经接收的编码单元的集合中，并将针对该块的编码单元的所需要的数量递减一 730。然后，进行检查以判断所需要的编码单元的数量是否是零 740，并且如果是，则使用 FEC 解码器来恢复该活动块，并准备接收针对下一个活动块的编码单元 750。接收方 TF 可靠性控制协议还持续地检查，以判断是否到了发送接收方反馈的时间 770，这由相应的接收方速率控制协议来确定。如果到了该时间，则准备并发送接收方反馈 780，该接收方反馈采用图 5 中所示的接收方反馈格式的格式。

[0080] 应当注意的是，这只是总 TF 可靠性控制协议的部分描述。例如，其没有规定接收方 TF 可靠性控制协议发送接收方反馈所遵循的条件。这可以由以下事件来触发：已接收的发送方数据的接收、时常到期的定时器、或者这些事件或任何其它事件（如由接收方速率控制协议所确定的）的任意组合。通常，足够频繁地发送接收方反馈，从而能在定期的基础上保持向发送方 TF 可靠性控制协议告知在接收方 TF 可靠性控制协议处接收编码单元的过程，但也不能太频繁，以便消耗与发送方数据几乎一样多的带宽，所述发送方数据包含从发送方 TF 可靠性控制协议向接收方 TF 可靠性控制协议发送的编码单元。

[0081] 应当注意的是，在以下意义上，可以将 TF 可靠性控制协议视作为是“浪费的”。假设 B 是各个数据块的大小（以编码单元为单位），假设 R 是速率控制协议在发送分组时所按照的速率，以及假设 RTT 是发送方端系统和接收方端系统之间的往返时间，以及假设 N =

R*RTT。假使在发送方和接收方之间不存在分组丢失。然后，在发送方 TF 可靠性控制协议已经发送了针对活动块的 B 个编码单元（其足够用于恢复该块）之后，其继续发送 N 个另外的编码单元，直到其从接收方 TF 可靠性控制协议接收到用于指示足够的编码单元已经到达从而能恢复该块的指示为止，而全部这 N 个编码单元都被浪费了。为了恢复长度为 B 的块，要求发送 B+N 个编码单元，因而有效吞吐量是 B/(B+N)。

[0082] 如果与 N 相比，B 相对较小，则有效吞吐量远不是最优的，并且发送方和接收方之间大量被使用的带宽是浪费的。另一方面，如果与 N 相比，B 较大，则发送方 TF 可靠性控制协议和接收方 TF 可靠性控制协议中的缓冲区的大小可以较大，这还意味着在接收方处该数据流的传输延时也较大。作为例子，假使编码单元的大小是 1 千比特，速率 R 是 1,000 个编码单元每秒＝ 1 兆字节每秒＝ 8 兆比特每秒，并且 RTT 是一秒。则 N ＝ R*RTT ＝ 1 兆字节。如果将块的大小设置为 B ＝ 3 兆字节，则有效吞吐量仅仅近似是 (B/(B+N)) ＝ 0.75，即，浪费了大约 25％的已发送编码单元。为了将有效吞吐量增加到例如 0.98，使得只浪费大约 2％的已发送编码单元，则要非常大的缓冲区大小 B ＝ 49 兆字节。这种大小的缓冲区则导致可靠性控制协议增加至少 50 秒的延时。

[0083] 存在关于上文所描述的 TF 可靠性控制协议的多种可能的变型。例如，在已经从一个块发送了 B 个编码单元之后，发送方 TF 可靠性控制协议可以停止发送编码单元，并等待接收用于指示是否已经接收到足够的编码单元从而能恢复该块的接收方反馈。如果不存在丢失，则这种变型将不发送将会被浪费的任何编码单元，但即使在这种情况下，在各个块之间也存在 RTT 时间的间隙，并且如果带宽没有用于任何其它目的，则这种协议仍然导致 R*RTT 的浪费的带宽量。此外，与理想情况相比，总传输时间将要慢 B/(B+N) 的因子。如果存在丢失，则这种变型将进一步增加延时，并使传输速度降低，这是因为最终将必须发送另外的编码单元，以替代丢失的编码单元来恢复该块。

[0084] 相对于无编码可靠性控制协议，TF 可靠性控制协议更具优势，这是因为可以在不需要接收方反馈的情况下，通过任何后续接收的根据相同的块所生成的编码单元来补偿任何丢失的编码单元。TF 可靠性控制协议是浪费的，其主要原因在于该协议的串行本质，在该意义上，每个块的传送在下一个块的传送开始之前完成。本文所描述的改进的可靠性控制协议可以用于以智能的方式来交织对这些块的处理。

[0085] 图 8 示出了交织的一种说明性例子。在这个例子中，存在两个活动块，第一活动块 AB 1(810) 和第二活动块 AB 2(820)。图 8 的下半部分示出了随着时间的数据分组发送模式的例子，其中根据每个分组是包含针对 AB1 还是 AB 2 的编码单元，将相应的分组标记为 AB 1 或 AB 2。在这个例子中，首先发送包含针对 AB 1 的编码单元的四个分组 (830(1)、830(2)、830(3) 和 830(4))，然后是包含针对 AB 2 的编码单元的两个分组 (830(5) 和 830(6))，接着是包含针对 AB 1 的编码单元的一个分组 (830(7))，包含针对 AB 2 的编码单元的一个分组 (830(8)) 和包含针对 AB 1 的编码单元的一个分组 (830(9))。通常，应当对针对不同块的编码单元之间的交织进行设计，以使有效吞吐量最大化，并使总的缓冲要求（以及相应引入的延时）最小化。

[0086] 如图 8 中所示，发送方设备（例如发送方端系统 100（图 1）中的一个）可以向接收方设备（例如接收方端系统 160（图 1）中的一个）发送经交织的数据。以这种方式，图 8 描述了一种方法的例子，该方法包括：发送针对第一块的编码单元的第一集合，其中，与为了

恢复第一块而需要的编码单元的最小数量相比，编码单元的第一集合包括更少的数量；在发送了编码单元的第一集合之后，发送针对第二块的编码单元的第二集合，其中，在该数据流中，第二块位于第一块之后；以及在发送了编码单元的第二集合之后，发送编码单元的第三集合，其中，编码单元的第三集合包括针对第一块的一个或多个编码单元。同样，图 8 还描绘了一种方法的例子，该方法包括：接收针对第一块的编码单元的第一集合，其中，与为了恢复第一块而需要的编码单元的最小数量相比，编码单元的第一集合包括更少的数量；在接收到编码单元的第一集合之后，接收针对第二块的编码单元的第二集合；以及在接收到编码单元的第二集合之后，接收编码单元的第三集合，编码单元的第三集合包括针对第一块的一个或多个编码单元。

[0087] 图 8 描绘了其中在完全形成一个块之前，就发送针对该块的编码单元的例子。例如，可以在完全形成活动块 AB 1 之前，就发送针对活动块 AB 1 的编码单元，即，当发送方发送针对 AB 1 的第一编码单元 830(1)、830(2) 时，并非 AB 1 中的全部在发送方处都是可用的，然而，可以在已经完全形成 AB 1 之前，就发送 AB 1 的编码单元。另外地或替代地，可以将 AB 2 视作为未完全形成，然而，可以在已经完全形成 AB 2 之前，就发送 AB 2 的编码单元。例如，可以在 AB 2 中的全部在发送方处完全可用之前，就发送针对 AB 2 的编码单元 830(4)、830(5)。同样，图 8 描绘了以交织的方式，传输针对 AB 1 和 AB 2 的编码单元的例子。

[0088] 图 9 是描绘了交织式可靠性控制协议可以使用的格式的示例性集合的框图。发送方数据格式描述了发送方交织式可靠性控制协议在可向接收方交织式可靠性控制协议发送编码单元和相应的可靠性控制信息时所采用的格式。这个例子包括：块编号 910，其指示该编码单元是根据哪个块生成的；序列号 920，其指示已经发送了多少个根据这个块的编码单元；编码单元 ID930，其指示如何根据该块生成该编码单元；以及编码单元 940，接收方交织式可靠性控制协议内的 FEC 解码器可以使用其来恢复该块。接收方反馈格式描述了接收方交织式可靠性控制协议在可向发送方交织式可靠性控制协议发送可靠性控制信息时所采用的格式。对于每个活动块而言，这包括：块编号 (950(1),950(2))、为了恢复该块而需要多少另外的编码单元 (960(1),960(2))、以及迄今为止从该块接收到的最高序列号 (970(1),970(2))。

[0089] 以这种方式，图 9 表示其中接收设备（例如接收方端系统 160 中的一个）发送以及发送设备（例如发送方端系统 100 中的一个）接收用于表示多个并行网络路径中的每个网络路径上的数据的丢失率的数据的例子，其中用于表示丢失率的数据包括：用于标识所接收的编码单元所针对的多个块中的每个块的数据、这些块中的每个块需要的编码单元的数量、以及用于定义接收的针对与这些块中的每个块有关的网络分组的最高序列号的数据。

[0090] 图 10 是描绘了基本发送方交织式可靠性控制协议的逻辑的例子的流程图。图 10 中的示例性协议可以由发送设备（例如图 1 中的发送方端系统 100 中的一个）来执行。在该协议的这个例子中，基本发送方交织式可靠性控制协议持续地进行检查，以判断是否到了发送发送方数据的时间 1005，这是由相应的发送方速率控制协议来确定的。如果到了发送发送方数据的时间，则基本发送方交织式可靠性控制协议使用以下规则集合来确定根据哪个活动块生成编码单元并进行发送。

[0091] 基本发送方交织式可靠性控制协议针对各个活动块 i 跟踪以下变量 (1010)：B_i

是为了恢复该块而需要的编码单元的数量；R_i 是基本发送方交织式可靠性控制协议基于接收的接收方反馈知道的基本接收方交织式可靠性控制协议已经从该块中接收到的编码单元的数量；L_i = B_i - R_i 是基本发送方交织式可靠性控制协议知道的基本接收方交织式可靠性控制协议为了恢复该块而需要接收的未被确认的编码单元的剩余数量；U_i 是已发送的针对该块的编码单元的数量，其中基本发送方交织式可靠性控制协议尚未接收到针对这些编码单元的确认；X_i 是确定基本发送方交织式可靠性控制协议将如何积极地发送针对该块的编码单元的参数。

[0092] 这些变量可以如下所述地进行确定：根据块的大小和各个编码单元的大小来确定 B_i 的值。应当注意的是，处理可以在尚未完全地可用于发送方的块上开始，即，尚未完全地形成的块可以变得活动，并且可以发送针对它们的编码单元。在这种情况下，可以将 B_i 确定成为了恢复块 i 而需要的编码单元的数量（如果该块具有当前可用于发送方的大小的话），在该情况下，随着块 i 的更多部分变得可用于发送方，B_i 可以在发送过程期间动态地增长，直到完全地形成块 i 为止，在该时间点，B_i 的值保持不变。因而，当块 i 的初始部分首先可用时，B_i 可以开始于 1，并随着块 i 的更多部分变得可用于发送方，B_i 增长，直到 B_i 达到其最终值为止，即，B_i 达到完整的块 i 的大小。

[0093] 通常，对于给定的块而言，有时对于全部块而言，每个编码单元具有相同的大小，并且对该大小进行选择以适合于数据分组的有效载荷，例如，编码单元的长度可以是 1024 字节。每个块的大小通常可以是相同的，或者其也可以变化，或者其可以取决于发送方处该数据流的到达速率，或者其可以取决于数据分组的发送速率，或者其可以取决于这些和其它因素的组合。基于在步骤 1030 中接收的接收方反馈，确定 R_i 的值。U_i 的值是下面二者之间的差值：最后发送的包含针对该块的编码单元的发送方数据中的序列号和在针对该块的接收方反馈中接收的最高序列号。

[0094] 以这种方式，可以基于块的大小和针对该块的编码单元的大小，计算为了恢复该块而需要的编码单元的数量（例如，B_i）。同样，可以将为了恢复该块而需要的另外的编码单元的数量（例如，L_i）计算成：为了恢复该块而需要的编码单元的数量（例如，B_i）和已接收的编码单元的数量（例如，R_i）之间的差值。

[0095] X_i 的值取决于总可靠性控制协议，如稍后所解释的，在 X_i 的选择中存在权衡。X_i 的值可以在针对该块的全部编码单元的发送期间保持不变，或者其可以以各种不同的方式来改变值，稍后将解释这些方式中的一些方式。本质上，每个时间点的 X_i 是除了为了恢复该块而需要的最小值之外，在不具有来自基本接收方交织式可靠性控制协议的任何另外的接收方反馈的情况下，基本发送方交织式可靠性控制协议愿意发送多少个另外的编码单元的测量值。由于 L_i 是除了已经确认的已接收编码单元之外，为了恢复块 i 而需要的编码单元的数量，并且由于 U_i 是在途的并且尚未被确认的针对块 i 的编码单元的数量，则 L_i+X_i - U_i 是基本发送方交织式可靠性控制协议愿意在这个时间点发送的针对块 i 的另外的编码单元的数量。

[0096] 关于 X_i 的值的权衡如下所述。随着 X_i 增加，有效吞吐量减少，这是由于除了为了恢复活动块 i 而需要的最小值之外，基本接收方交织式可靠性控制协议可接收多达 X_i 个的编码单元。另一方面，随着 X_i 增加，活动块的总大小减小，这是因为随着 X_i 增加，被分配用于完成活动块 i 的可靠接收的分组时隙的数量也增加。因为针对块 i 的 X_i 个编码

单元可能丢失，并且基本接收方仍然能够恢复该块，而无需等待接收方反馈来触发另外的编码单元的传输，所以这允许一旦块 i 变得活动，就更快速地恢复块 i。事实证明，与用于诸如 TF 可靠性控制协议或者无编码可靠性控制协议之类的其它可靠性控制协议的相应权衡相比，取决于 X_i 的总缓冲区大小和有效吞吐量之间的权衡要有利得多。

[0097] 在步骤 1015 处，进行测试以确定是否存在满足不等式 L_i+X_i－U_i>0 的活动块 i。L_i 的值是基于接收方反馈已经确认的编码单元，接收方为了恢复该块而需要的编码单元的数量。U_i 是在途的针对这个块的未被确认编码单元的数量，因而 L_i-U_i 是当在途的这些编码单元中没有任何一个编码单元丢失时，将必须发送的另外编码单元的数量，因而如果 L_i-U_i 是零或者更小，则接收方将能够恢复该块（如果在途的全部编码单元都到达的话）。另一方面，这些编码单元中的一些编码单元可能丢失，并且 X_i 是发送方愿意主动发送以使免受丢失，从而避免必须发送由后续的接收方反馈所触发的针对该块的另外的编码单元的另外的编码单元的数量。

[0098] 因而，如果 L_i+X_i－U_i>0，则发送方愿意发送针对块 i 的更多编码单元，而如果其是零或者负值，则发送方不愿意发送针对块 i 的更多编码单元。因而，如果在步骤 1015 中，存在满足 L_i+X_i－U_i>0 的活动块 i，则在步骤 1020 中，针对最早的这样的活动块，生成编码单元并发送相应的发送方数据。如果不存在这样的活动块，则在步骤 1025 中，根据全部活动块当中的最早活动块，生成编码单元并发送相应的发送方数据。优选地，以这样的方式来设置这些参数，以尽可能避免不具有满足步骤 1015 中的条件的块，这会强制步骤 1025 的执行，这是因为在本质上，应当在万不得已时才进行步骤 1025 以清除基本发送方交织式可靠性控制协议内的缓冲区。步骤 1020 和 1025 中的发送数据可以经由多个路径被发送给相同的接收方。

[0099] 如果存在尚未完全形成的活动块，则应当修改步骤 1015，以便还考虑这些尚未完全形成的块。在一个变型中，步骤 1015 中示出的针对活动但尚未完全形成的块的相应条件应当被替代，无论针对尚未完全形成的活动块的全部源编码单元是否被发送，并且如果它们尚未被全部发送，则可以发送剩余的未发送的源编码单元。因而，在这个变型中，针对块 i 可以发送的编码单元的数量多达 B_i，其中 B_i 是部分形成的块 i 中的源编码单元的当前数量。在第二变型中，步骤 1015 中示出的针对活动但尚未完全形成的块的相应条件应当被替代为设置 X_i ＝ 0，并基于增强型接收方反馈，对已经被确认为没有接收到的编码单元进行重传，该增强型接收方反馈还指示在已经被确认为已丢失或已接收的编码单元当中，哪些编码单元是丢失的。因而，在这个变型中，如果块 i 是活动的，但尚未完全形成，则利用条件 L_i－U_i>0 来替换步骤 1015 中的条件 L_i+X_i－U_i>0。在这种情况下，如果所有源编码单元都已经发送，但一些被确认为已丢失，则如果满足条件 L_i－U_i>0，就可以重新发送这些编码单元。

[0100] 该协议的一个变型如下所述。活动块的编号开始于一，即，激活数据流的第一块。仅仅当不存在满足步骤 1015 中的条件的活动块时，才激活该数据流中的新块。使用这种简单策略，当需要时，块才变成活动块，因而活动块的数量和相应的缓冲区大小自调整到为了保证针对块 i 的有效吞吐量 B_i/(B_i+X_i) 而需要的数量。

[0101] 该协议的另一变型如下所述。在这个变型中，总缓冲区大小始终保持相同的大小（如果全部块具有相同大小，则这意味着始终存在固定数量的活动块），而有效吞吐量可以

变化。只要不存在满足步骤1015中的条件的活动块，就增加针对该活动块的X_i的值，直到存在满足步骤1015中的条件的活动块为止。只要合适，就减小针对活动块i的X_i的值，其约束条件是始终存在满足步骤1015中的条件的活动块。存在很多可能的方式来增加和减小X_i的值，例如：平等地增加全部值；成比例平等地增加全部值；与针对最后活动块的值相比，更多地增加针对第一活动块的值；与针对第一活动块的值相比，更多地增加针对最后活动块的值。类似地策略可以用于减小X_i的值。本领域技术人员也可以考虑多种其它变型。

[0102] 存在该协议的这些变型的很多其它组合和扩展，它们非常众多，难以在此一一描述，但对于本领域技术人员而言应当是显而易见的。

[0103] 在步骤1030中，进行检查以判断是否已经接收到任何接收方反馈，如果是，则在步骤1035中，基于此来更新这些参数中的全部参数，即针对全部活动块i的参数R_i、U_i和X_i。在步骤1040中，进行检查以判断最早活动块是否已经被确认为完全恢复了，如果是，则在步骤1045中，去激活最早活动块，并且处理返回至步骤1040，而如果否，则处理继续至步骤1050。在步骤1050中，进行检查，以判断另一个块是否已经对变得活动准备就绪，如果是，则在步骤1060中，使该下一个块活动，并且处理返回至步骤1050，而如果否，则处理继续至步骤1005。通常，在发送当前活动块的同时，可以准备下一个块或者若干个后续块，并在最早活动块将被去激活时或者之前，准备被激活。

[0104] 以这种方式，图10描绘了一种方法的例子，该方法包括：经由多个并行网络路径，向客户端设备发送经前向纠错的数据；从客户端设备接收用于表示在这些网络路径中的每个网络路径上发送的数据的丢失率的数据；以及基于该用于表示丢失率的数据，修改在这些并行网络路径上针对后续数据传输发送的经前向纠错的数据的量。针对这些并行网络路径中的每个网络路径，所接收的用于表示丢失率的数据可以包括：用于标识接收的针对经由各自的并行网络路径发送的分组流的最高序列号的数据（例如，如下面参照图13所讨论的）、和/或用于标识被客户端设备接收的编码单元所针对的多个块中的每个块的数据、这些块中的每个块需要的编码单元的数量、以及用于定义被客户端设备接收的针对与这些块中的每个块有关的网络分组的最高序列号的数据（例如，如上面参照图9所讨论的）。

[0105] 图11是描绘了基本接收方交织式可靠性控制协议的逻辑的例子的流程图。在该协议的这个版本中，基本接收方交织式可靠性控制协议持续地进行检查，以判断是否已经接收到发送方数据1105，例如，其可以采用图9中示出的发送方数据格式。如果是，则在步骤1110中，更新其关于全部活动块的信息，并且进行检查以判断该发送方数据内的已接收编码单元是否是根据活动块1115。如果该编码单元是根据已经恢复的块，或者是根据与当前活动块相比的该数据流中的太靠前的块，则在步骤1135中，丢弃该编码单元，因而这是浪费的发送方数据，这是由于其没有用于恢复任何块。否则，在步骤1120中，将该编码单元添加到针对生成其所根据的活动块的编码单元池中，并对为了恢复该活动块而需要多少编码单元进行更新。

[0106] 将需要的针对块i的编码单元的数量计算成：B_i减去已接收编码单元的数量后得到的值。存在各种方式来向基本接收方交织式可靠性控制协议传达B_i的值，例如，可以将B_i的值包括在每个发送方数据内，可以在单独的控制消息中发送B_i的值，对于全部块而言，B_i的值可以是相同的，并在会话发起期间进行传达等等。

[0107] 然后在步骤 1125 中，进行检查以判断针对最早活动块的编码单元的需要的数量是否为零，如果是，则在步骤 1130 中，使用 FEC 解码器来恢复该活动块，以及准备接收针对新的下一个活动块的编码单元。基本接收方交织式可靠性控制协议还持续地检查，以判断是否到了发送接收方反馈的时间 (1140)，这是由相应的接收方速率控制协议来确定的。如果到了该时间，则在步骤 1145 中，准备并发送接收方反馈，其可以例如采用图 9 中示出的接收方数据格式。

[0108] 应当注意的是，上文只是总基本交织式可靠性控制协议的部分描述。例如，其没有规定基本接收方交织式可靠性控制协议发送接收方反馈所遵循的条件。这可以由以下事件来触发 :已接收的发送方数据的接收、时常到期的定时器、或者这些事件或任何其它事件（如由接收方速率控制协议所确定的）的任意组合。通常，足够频繁地发送接收方反馈，从而能在定期的基础上保持向基本发送方交织式可靠性控制协议告知在基本接收方交织式可靠性控制协议处接收编码单元的过程，但也不能太频繁，以便消耗与发送方数据几乎一样多的带宽，所述发送方数据包含从基本发送方交织式可靠性控制协议向基本接收方交织式可靠性控制协议发送的编码单元。

[0109] 与 TF 可靠性控制协议或者无编码可靠性控制协议相比，基本交织式可靠性控制协议可以在有效吞吐量和缓冲区大小之间具有更佳的权衡。例如，假使针对基本交织式可靠性控制协议，存在至多两个活动块。假设 B 是各个数据块的大小（以编码单元为单位），假设 R 是速率控制协议在发送分组时所按照的速率，以及假设 RTT 是发送方端系统和接收方端系统之间的往返时间，以及假设 N ＝ R*RTT，以及假使 X 是针对活动块的固定常数。在这个例子中，假定全部这些参数都具有固定值，但通常它们可以在数据传送期间动态地变化，并假定 B> ＝ N。

[0110] 假使在发送方和接收方之间不存在分组丢失。然后，基本发送方交织式可靠性控制协议发送针对最早活动块的 B+X 个编码单元，然后根据下一个活动块的编码单元，直到其接收到指示基本接收方交织式可靠性控制协议已经成功地恢复最早活动块的接收方反馈为止。在这个时间点，基本发送方交织式可靠性控制协议去激活最早活动块，已经被发送的一些编码单元所针对的下一个活动块变成最早活动块，并且下一个块被激活变成活动块。因而，使用 B+X 个编码单元来恢复长度为 B 的块，因而浪费了 X 个已发送的编码单元。

[0111] 另一方面，如果 B> ＝ N，则始终存在一个满足图 9 的步骤 1015 中所示出的不等式的活动块。因而，有效吞吐量是 B/(B+X)，而缓冲区的总大小是 2*B（如果存在两个活动块的话）。作为例子，假使编码单元的大小是 1 千字节，速率 R 是每秒 1,000 个编码单元＝每秒 1 兆字节＝每秒 8 兆比特，并且 RTT 是一秒。则 N ＝ R*RTT ＝ 1 兆字节。如果块的大小是 1 兆字节，其意味着 B ＝ 1,000 个编码单元，并且 X 被设置为 10 个编码单元，则有效吞吐量近似是 (B/(B+X)) ＝ 0.99，即最多浪费 1％的已发送编码单元，而总缓冲区大小仅仅为 2MB，其意味着在这个例子中，基本发送方交织式可靠性控制协议增加大约 2 秒的延时。应当注意的是，与相同情形下的发送方 TF 可靠性控制协议相比，这个缓冲区大小要小 25 倍。

[0112] 在上面所描述的不存在分组丢失的例子中，可以将 X 的值设置成零，将有效吞吐量增加到 1.0。然而，当存在任何分组丢失时，结果证明，设置 X>0 可以具有明显的优势。例如，如果在上面的例子中，每发送 1,000 个编码单元有最多 10 个编码单元丢失，则分析显示在 X ＝ 10 的情况下，实现相同的有效吞吐量和缓冲区大小，而在 X ＝ 0 时，这不一定必然成

立。当分组丢失更大地变化并且是未知时，具体而言，当每 B 个分组中丢失的分组数量可能超过 X 时，仍然证明与使用 TF 可靠性控制协议或者无编码可靠性协议所能够实现的有效吞吐量和缓冲区大小相比，基本交织式可靠性控制协议所能够实现的有效吞吐量和缓冲区大小是相当好的并且更好量化。

[0113] 作为另一例子，假使发送速率 R（以分组每秒为单位）和往返时间 RTT 保持不变，并且 N = R*RTT。假使分组丢失是随机的，使得每个分组都具有概率 p 会丢失。此外，假使每个块 i 具有大小 B_i，根据分组单位而言，它们是相同的大小 C，并且每个 X_i 具有相同的值 Y。此外，假使使用上面所描述的当需要时仅仅激活一个新块的协议的变型。考虑一个块从其第一次被激活到其被去激活的时间（因为没有从接收方接收到其已经被恢复的确认）。在某个时间 t，当该块的 C-N 个分组已经被确认时，存在 F = N+Y 个分组处于未被确认的途中，并且发送方知道接收方需要这些分组中的 N = F-Y 个分组来恢复该块。在时间 t+RTT，该块的 F 个分组处于途中，在时间 t，接收方已经接收到这些分组中的 (1-p)*F 个，并且发送方已经接收到了确认。

[0114] 因而，在时间 t+RTT，发送方知道接收方需要的剩余分组的数量现在是 N-(1-p)*F = p*F-Y，因而在途的分组的数量现在是 p*F。继续该逻辑，在时间 t+i*RTT，发送方知道接收方需要的剩余分组的数量是 pˆi*F-Y，因而在途的分组的数量是 pˆi*F。当发送方知道接收方需要的分的组数量变成低于零时，则该块已完成传输，故在时间 t+i*RTT 时，i 满足 pˆi*F - Y< = 0。在该不等式成立时的 i 的最小值，是当 i 近似是 ln((N/Y)+1)/ln(1/p) 时。

[0115] 由于在每个 RTT 中，接收方接收到近似 (1-p)*N 个分组，这意味着截止接收到针对该块的确认的时间，该发送方协议可以在数据流中处理的超出该块的最远距离至多是 (ln((N/Y)+1)/ln(1/p))*(1-p)*N 个分组。应当注意的是，对于 p 的全部值而言，(1-p)/ln(1/p)< = 1，这意味着缓冲区的大小在长度上至多是 C+ln((N/Y)+1)*N 个分组。当然，这里全部假定随机过程的行为正是其预期的行为，但这确实给出了该协议如何工作的一个粗略想法，至少是由于 Y 不是非常的小。在这种情况下，有效吞吐量是 C/(C+Y)。因而，例如，如果 RTT = 1，R = 1,000，C = 1,000，Y = 50，则 N = 1,000，ln(1,000/50) 近似是 3，并且缓冲区大小大约为 1,000+(3+1)*1,000 = 5,000，有效吞吐量为 1,000/(1,000+50)，其大约为 0.95。

[0116] 存在关于上面所描述的基本交织式可靠性控制协议的很多变型，在阅读完本描述之后，这些变型应当是显而易见的。例如，如上所述，发送方可靠性控制协议可以一次使用两个以上的活动块，这具有能够减少在发送方可靠性控制协议和接收方可靠性控制协议处使用的缓冲区的总大小的潜在优点，其代价是在管理更多的活动块时更加复杂。

[0117] 作为变型的另一例子，使用随机过程来确定将发送根据哪个活动块的编码单元可能是有益的。这是因为分组丢失模式可能是系统性的，并且不一定是随机的，因而对于在选择下一次发送哪个编码单元时使用的确定性过程而言，存在分组丢失模式，使得一些块从未被恢复，但仍然向接收方传输分组。例如，考虑如下丢失模式：只要确定性过程发送根据特定活动块的编码单元，则该编码单元就被丢失，但只要发送针对任何其它活动块的编码单元，则该编码单元就能到达接收方处。

[0118] 然后，在这个例子中，接收方从未恢复该特定活动块，即使接收方仍然接收编码单

元。为了克服这种类型的系统性丢失，发送方可靠性控制协议对发送的下一个编码单元所根据的活动块进行随机化是有利的。一种实现该目标的简单方式是发送方可靠性控制协议对要发送的多批的 Q 个编码单元一起进行缓冲，然后以随机顺序来发送每批的 Q 个编码单元。还可以使用更复杂的方法，例如，对于要被发送的每个编码单元而言，指派一个在下一次发送编码单元时将被发送的动态变化的概率，其中随着该编码单元没有被选中的次数增多，该概率增加。另一变型是修改基本发送方交织式可靠性控制协议的如图 10 中所示的步骤 1020，使得从满足步骤 1015 中的条件的活动块当中，随机地生成被发送的编码单元（使用适当选择的概率分布，其可以有利于更早的活动块，并可以随时间而动态地变化）。

[0119] 如果使用参数 X_i 来确定何时发送针对活动块 i 的编码单元，则存在关于如何在传输期间调整 X_i 的很多变型。一个例子是将 X_i 固定为某个值，并在整个传输过程中都保持该值。例如，可以将 X_i 设置为零，或者某个其它固定值，比如 10。另一例子是在开始传输根据活动块 i 的编码单元时，将 X_i 固定为某个值，然后在每次要发送编码单元，并且没有满足用于发送根据活动块 i 的编码单元的条件时，对 X_i 进行递增。存在关于如何对 X_i 进行递增的很多变型。作为例子，在前 N 次，对 X_i 递增零，以后每次则递增 N/B。在一些步骤中，X_i 的递增量还可以是负值。

[0120] 作为其它变型，不是针对每个活动块 i 仅仅只使用参数 X_i，如基本交织式可靠性控制协议中所描述的，可以使用其它方式来确定是否应当发送根据特定活动块的编码单元。例如，可以保持分组丢失概率的平均值，然后可以基于最近的分组丢失概率是当前分组丢失概率的良好预测的假定，来确定允许发送的根据一活动块的编码单元的数量。例如，如果平均丢失概率当前是 p，则一种策略是修改基本发送方交织式可靠性控制协议的如图 10 中所示的步骤 1015，使得条件为 L_i+X_i/(1-p)-U_i*(1-p)>0。

[0121] 这种特定选择背后的理由在于：如果针对活动块 i 的 U_i 个编码单元在途，它们中的仅仅 1-p 部分将到达基本接收方交织式可靠性控制协议，而如果发送 X_i/(1-p) 个另外的分组，则 X_i 将到达基本接收方交织式可靠性控制协议。因而，基本接收方交织式可靠性控制协议在总体平均上将接收针对活动块 i 的 B_i+X_i 个编码单元，并且可以将 X_i 个另外的编码单元的值设置得足够大以考虑分组丢失率的变化，从而避免依赖于接收方反馈来传输用于恢复该块的足够数量的编码单元。

[0122] 交织式可靠性控制协议的其它变型考虑分组可能无法以与发送顺序相同的顺序到达接收方处的概率。因而，与前一个接收方反馈相比，来自接收方的后续接收方反馈可能例如反向报告针对给定活动块的更大数量的已接收编码单元，即使从该块接收的最高序列号是相同的。因而，可以在发送方和接收方二者中，对基本交织式可靠性控制协议中的逻辑进行修改，以容适说明重新排序的分组。

[0123] 如先前所描述的，通常通过恰当地设置参数，来避免如图 10 中所示的基本发送方交织式可靠性控制协议的步骤 1025，使得在各个时间点，至少有一个活动块满足条件 1015。关于步骤 1025 的一种变型是改变选择的活动块，其中将根据该活动块生成编码单元并进行发送。例如，在步骤 1025 中，可以随机地选择活动块，或者该选择可以循环遍历活动块的集合。

[0124] 图 10 的步骤 1040 到 1060 描述了用于去激活已恢复的块，并激活另外的块以进行发送的方法。一种简单方法是当最早块由于被恢复而被去激活时，始终激活下一个块，因而

始终保持相同数量的活动块。可以节省总缓冲区大小以及相应的延时的一种变型是仅仅当发送超出最新的当前活动块的块的编码单元时，才激活下一个块。

[0125] 对于基本交织式可靠性控制协议的一些变型而言，在任何时间点的活动块的数量是固定的。一种变型是允许活动块的数量根据各种因素进行变化，这些因素包括：在使数据可用于传输时所按照的速率、发生了多大的分组丢失率、分组的发送速率的变化等等。例如，在低分组丢失状况和低发送速率状况下，活动块的数量可以保持为较小，但随着丢失状况变得严重或者发送速率增加，可以允许活动块的数量临时地增长。因而，缓冲和延时根据该协议操作时的状况动态地变化。

[0126] 即使活动块的数量保持固定，也可以允许活动块的聚合大小发生变化。在这种情况下，每个后续活动块的大小可以与前一个块不同。例如，随着数据可用速率增长，后续活动块的大小也可以增长，并且随着发送速率增大，后续活动块的大小也可以增长。每个活动块的长度可以取决于时间，例如，在新的块形成之前至多可以过去那么长时间，其可以取决于长度，即，每个块至多是那么长，或者其可以是这些因素和其它因素的组合。

[0127] 一个块的结束和下一个块的开始可以由交织式可靠性控制协议自动地决定，其可以由应用来确定，或者由这些因素和其它因素的某种组合来确定。例如，数据流的块可能对于应用具有逻辑含义，例如用于MPEG流的图片组块或者I帧，因而，交织式可靠性控制协议将数据流划分成块的方式，可以遵守逻辑应用块的边界。或者，应用可以向交织式可靠性控制协议指示块之间的优选边界，并且交织式可靠性控制协议尝试尽可能地遵守这些边界，但仍然可以允许其按照除应用所提供的这些边界之外的点，划分块之间的边界。

[0128] 交织式可靠性控制协议的另一变型是允许该协议不按顺序地向接收方可靠地传送全部块，而是尝试在服从其它约束的情况下，尽可能地实现这个目标。例如，在流式传输应用中，尽可能可靠地传送数据流可能是重要的，但还存在其它约束，例如关于数据流的时间约束。例如，可以是这样的情况，在一定的时间之后，该数据的某些部分不再是相关的，或者存在关于交织式可靠性控制协议可以引入多少延时的较强的限制，例如，在交互式视频会议应用中。在这些情况下，可以对发送方交织式可靠性控制协议和接收方交织式可靠性控制协议进行修改，以允许在这些块被完全恢复之前，跳过这些块中的一些块。

[0129] 例如，发送方交织式可靠性控制协议可以被约束为只允许活动块活跃给定的时间量，或者其可以具有对应用所提供的每个块的硬时间约束，在该时间之后，其不再被允许发送针对该块的编码单元，或者其可以被允许针对每个块只发送所提供的最大数量的编码单元，或者这些约束的任意组合。类似的约束可应用于接收方交织式可靠性控制协议。对于这些应用而言，可以对交织式可靠性控制协议进行修改以遵守这些约束。

[0130] 在交织式可靠性控制协议的一些变型中，存在一个发送方和一个接收方。其它变型包括但不限于：一个发送方和多个接收方；一个接收方和多个发送方；多个发送方和多个接收方。例如，在一个发送方/多个接收方的变型中，当发送信道是广播信道或多播信道时，可以对发送方交织式可靠性控制协议进行修改，使得在图10的步骤1010中，发送方针对每个活动块i，将R_i的值计算成在来自任何接收方的已接收的被确认的编码单元的最小数量。

[0131] 作为针对一个发送方/多个接收方的变型的另一例子，当发送方向各个接收方发送单独的分组流时，可以对发送方可靠性控制协议进行修改，使得在图10的步骤1010中，

发送方针对每个活动块 i 和针对每个接收方 j，将 R_ij 的值计算成来自接收方 j 的针对活动块 i 的已接收的被确认的编码单元的数量，以及计算 L_ij = B_i - R_ij，并且可以将 U_ij 计算成针对发送给接收方 j 的活动块 i 的已发送但还未被确认的编码单元的数量，然后可以改变步骤 1015 中的条件，以确定是否存在活动块 i，使得对于某个接收方 j，L_ij+X_i-U_ij>0。

[0132] 作为另一例子，对于很多发送方 / 一个接收方的变型而言，可以对接收方可靠性控制协议进行修改，使得接收方同时地从多个发送方接收针对相同或者不同活动块的编码单元，并通过广播信道或多播信道向全部发送方发送接收方反馈，或者使用具有潜在的单独接收方反馈的单独分组流向各个发送方进行发送。作为另一例子，对于多个发送方 / 多个接收方的变型而言，可以对上面针对一个发送方 / 多个接收方的情形和多个发送方 / 一个接收方的情形所描述的修改的步骤进行组合。

[0133] 另一变型是发送方可以同时地发送多个数据流，每个数据流使用发送方交织式可靠性控制协议的单独实例，或者考虑不同数据流的发送方交织式可靠性控制协议的版本，例如，可以限制用于针对全部流的全部分组的聚合发送速率，因而发送方可以决定对发送分组划分优先次序，以使某些数据流优先于其它数据流。类似地，接收方可以同时地接收多个数据流，每个数据流使用接收方交织式可靠性控制协议的单独实例，或者考虑不同数据流的接收方交织式可靠性控制协议的版本，例如，可以限制用于针对全部流的全部分组的聚合接收速率，因而发送方可以决定对接收分组以及处理和发送接收方反馈划分优先次序，以使某些数据流优先于其它数据流。

[0134] 上述变型中的任何变型可以与彼此进行组合。例如，其中由于例如时间和 / 或带宽限制而可能不能将一些块可靠地传送到接收方的协议可以与多个发送方 / 多个接收方的变型进行组合。

[0135] 以这种方式，图 11 描绘了一种方法的例子，该方法包括 ：经由多个并行网络路径从服务器设备接收经前向纠错的数据 ；确定这些网络路径中的每个网络路径上的数据的丢失率 ；以及向服务器设备发送用于表示在这些网络路径中的每个网络路径上的数据的丢失率的数据。

[0136] 图 12 是一种多径流式传输系统的框图，其中该多径流式传输系统可以根据本发明的技术，使用多径的基于 FEC 的可靠性传输控制方法。在这个例子中，视频生成器 (1205) 生成视频流 (1210)，该视频流由发送方传输协议 (1215) 来接收。发送方传输协议 (1215) 针对该视频流 (1210)，确定要发送什么数据，以及针对要发送的每块数据，确定沿着哪个路径流来发送。

[0137] 对于每个路径流而言，用于该路径流 (1220(1)、1220(2)) 的发射机通过网络 (1222) 发送针对该路径流的数据，以便至少部分地由接收方传输协议 (1225) 进行接收。接收方传输协议 (1225) 尽可能全保真地恢复原始视频流，并可能通过其它网络和通过服务器的其它集合，向其它设备产生这个视频流 (1230)，以便端用户设备进行回放。应当注意的是，视频只是一种示例性数据流，可以类似地处置非视频流 (例如，音频流或其它数据流)。

[0138] 如本文所解释的，通过使用多个路径，以及通过协调使用哪个路径 (当使用多个路径时) 等等，可以获得诸如降低的总延时或者更高的总数据吞吐量之类的改进，其通常意味着更高质量的流式传输体验。即使在多个路径具有不同属性 (例如，各自的延时、带

宽和丢失率），并且那些属性除了随路径发生变化之外，还可以随时间发生变化的情况下，也可以是该情况。这样，最初由视频生成器以一种顺序发射的分组，可能以不同的顺序被接收，这是由于与其它路径相比，一些路径可能更快速地传送它们的分组。

[0139] 在这种架构中，发送方传输协议（1215）和接收方传输协议（1225）可以具有已确立下来的数据格式的集合，它们使用这些数据格式的集合来与彼此进行通信，并且可以存在从发送方传输协议（1215）向接收方传输协议（1225）流动的数据，以及从接收方传输协议（1225）向发送方传输协议（1215）流动的控制信息和反馈信息。

[0140] 路径流的数量和发射机的数量可以是任意数量，并且这些数量可以是不同的，例如，可以存在十个路径流和三个发射机，这些路径流中的四个路径流可以通过第一发射机（1220(1)），并且三个路径流可以通过第二发射机和第三发射机中的每个发射机。发射机（1220）可以与发送方传输协议（1215）并置在相同的硬件设备内，或者发射机（1220）可以具有与托管发送方传输协议（1215）的硬件设备相分离的硬件设备，并且这些设备可以基于诸如 TCP 或 UDP 之类的标准传输协议，例如使用蓝牙或 WiFi 来与彼此进行通信。发射机（1220）将它们的数据发送给接收方传输协议（1225）所通过的网络可以针对于不同的发射机而不同，例如，这些发射机中的一些发射机可以使用 3G 网络来进行发送，其它发射机可以使用 LTE，其它发射机可以使用 WiFi。这些发射机可以使用相同或不同类型的不同运营商网络，例如，第一发射机可以使用 AT&T 网络，而第二发射机可以使用 Verizon 网络。在接收方传输协议 1225 之前，可以存在中间接收设备，例如，第一发射机可以向第一 CDN 操作的第一服务器发送，而第二发射机可以向第二 CDN 操作的第二服务器发送，并且第一服务器和第二服务器可以接收针对于接收方传输协议 1225 的数据。

[0141] 现在参照图 13 来描述多径的基于 FEC 的交织式可靠性传输控制协议方法。图 13 是描绘了一种示例性多径可靠性控制协议数据分组格式和相应的反馈信息格式的概念图。

[0142] 如图 13 的顶部所示出的，每个数据分组包括：流标识符，FID(1305)；针对该 FID 的序列号，针对 FID 的 SEQN(1310)；源块编号，SBN(1315)；源块长度（以源符号为单位），SBL(1318)；编码符号标识符（其还被称为编码单元标识符），ESI(1320)；和一个或多个编码符号（1325）。（编码符号有时还被称为编码单元）。FID(1305) 标识这个分组将被发送到的路径流，并且针对 FID 的 SEQN(1310) 是针对被发送到这个流的每个分组来递增 1 后的数量，因而针对 FID 的 SEQN(1310) 的范围由 FID(1305) 来界定。

[0143] SBN(1315) 标识在这个分组中携带的编码符号（1325）是根据哪个源块生成的，其中，SBN 通常是针对要发送的每个后续的源数据块来递增 1 后的数量。SBL(1318) 标识该源块中的源符号的数量。当分组携带源符号时，可以省略 SBL(1318)，并且对于一些应用而言，这是优选的，这是由于可能是如下情况：例如参见下面的开放源块的描述，在发送源块的源符号中的至少一些源符号的时间，源块中的源符号的数量是未知的，即，开放源块。

[0144] 或者，可以在全部数据分组中携带 SBL(1318)，但是针对在确定源块的大小之前所发送的携带源符号的分组，可以将 SBL(1318) 的值设置成零，或者针对携带源符号的全部分组，可以将 SBL(1318) 设置成零，或者可以在发送携带源符号的分组时，将 SBL(1318) 设置成源块中的源符号的当前数量。针对携带修复符号的全部分组，优选地将 SBL(1318) 设置成源块中的源符号的数量。还可以将 SBL(1318) 划分成两个子字段，一比特的标志指示该源块是开放的还是闭合的，即，如果在发送该分组时，没有确定该源块大小，则将该标志

设置成零（开放源块），而如果在发送该分组时，确定了该源块大小，则将该标志设置成一（闭合源块），并且 SBL(1318) 的剩余部分提供该源块中的源符号的数量。

[0145] 除了在携带针对这个源块的修复符号的全部分组中，利用这个 SBL 标志来指示该源块是闭合的之外，发送方反馈逻辑单元 (1420) 可以设置这个 SBL 标志，以便在携带源块的最后源符号的分组中指示该源块是闭合的，并且在该标志指示该源块是闭合的每个分组中，将 SBL 大小设置成该闭合源块中的源符号的实际数量。ESI(1320) 标识在这个分组中携带了针对由 SBN(1315) 所标识的源块的哪些编码单元，因而 ESI(1320) 的范围由 SBN(1315) 来界定。在每次沿着特定路径流来发送新的数据分组时，将该路径流的 FID 放置在该分组中，将针对该 FID 的 SEQN 递增一，并放置在该分组中，将要发送其编码符号的活动源块的 SBN 放置在该分组中，将这些编码符号的相应 ESI 放置在该分组中，以及将这些编码符号放置在该分组中，上述全部操作在发送该分组之前完成。

[0146] 接收方传输协议 (1225) 生成用于向发送方传输协议 (1215) 发送的反馈。在图 13 的底部示出了一种可能的接收方反馈信息格式。如图所示，接收方传输协议 (1225) 针对每个 FID(1350(1)、1350(2))，反向报告针对该 FID 的接收的相应的最高 SEQN(1355(1)、1355(2))。另外，接收方传输协议 (1225) 针对每个活动源块，反向报告该源块的 SBN(1360(1)、1360(2)) 以及迄今为止针对该源块的接收的编码符号 (1365(1))、1365(2)) 的数量。

[0147] 另外，接收方传输协议 (1225) 反向报告最低活动源块的源块编号 (1370)。当接收方传输协议 (1225) 以足够的确定性认为具有最低源块编号的当前活动源块是可恢复的，并且不需要针对当前源块的另外的编码符号时，接收方传输协议 (1225) 通常增加接收方传输协议 (1225) 反向报告的最低活动源块编号 (1370)，因而接收方传输协议 (1225) 将该源块指定成不活动的。在没有以足够及时的方式接收足够的编码符号的环境中，也有可能是，即使当存在接收方传输协议尚未恢复的具有更低源块编号的源块时，接收方传输协议 (1225) 也增加最低活动源块编号 (1370)。

[0148] 在阅读完本公开内容时，应当显而易见的是，其它变型也是可能的。可以对该系统进行扩增以允许发送方传输协议 (1215) 在如图 13 的顶部所示的发送方多径数据分组格式中以信号的形式发送最低活动源块编号。例如，可以向如图 13 的顶部所示的多径数据分组格式添加另外的参数“最低活动源块编号”，以便允许从发送方传输协议 (1215) 到接收方传输协议 (1225) 的这种信令。然后，这种功能允许发送方传输协议 (1215) 发送信号跳过一些源块，其中对于这些源块而言，不可能以满足该系统的端到端延时要求的方式来完成这些源块的发送和恢复。

[0149] 存在上面内容的很多变型。例如，可以使用字节范围来替代 ESI，以指示在分组有效载荷中携带什么源数据。作为另一例子，可以使用整个数据流内的块的字节范围来替代 SBN 和 SBL，以指示使用什么源块来生成在该分组有效载荷中发送的任何 FEC 数据，并结合 ESI 来指示在该分组有效载荷中发送什么特定数据。作为另一变型，根据不同块的编码符号可以包括在相同的数据分组内，并且 SBN、SBL 和 ESI（或者它们的等同物）的多元组可以包括在分组报头中，以标识在该数据分组中携带的编码符号。例如，可以对针对各个活动源块的分组中所携带的符号的比例进行选择，以便与当前针对每个这样的源块所发送的符号的数量成比例，即，对将针对活动块 i 发送的分组中的符号的比例进行选择，以便与 L_i+X_

i-U_i 的当前值成比例。

[0150] 图 14 以更详细的方式示出了多径流式传输发送方的框图。图 1 的发送方端系统 100 中的任何一个或者全部可以包括类似于图 14 的多径流式传输发送方的组件。如图 14 中所示，由视频生成器 (1205) 生成的视频流 (1210) 临时地存储在发送方传输协议 (1215) 内的源数据缓冲区 (1405) 内。FEC 编码器 (1410) 生成针对已经形成的源块的 FEC 修复符号，并将获得的 FEC 修复符号放置到修复符号缓冲器 (1415) 中，直到需要对它们进行传输为止。FEC 编码器 (1410) 可以在按需的基础上操作，当需要对修复符号进行传输时，生成针对活动源块的修复符号。或者，FEC 编码器 (1410) 可以预先生成针对活动源块的多个修复符号，使得它们准备好只要需要就被发送，以及减少调用 FEC 编码器 (1410) 的开销。

[0151] FEC 编码器 (1410) 生成针对源块的另外的修复符号的触发事件可以由来自发送方反馈逻辑单元 (1420) 的信号来触发，其中每当可能发送另外的编码符号时，发送方反馈逻辑单元 (1420) 确定要发送哪些另外的编码符号。因而，这些步骤表示一种方法的步骤的例子，该方法包括 ：为服务器设备计算该块需要的另外的编码单元的数量 ；以及向发送设备 (例如服务器设备) 发送用于表示该另外的编码单元的数量的数据。

[0152] 当源块的起始边界和结束边界二者都已经被确定时，认为该源块是闭合的，即，当源块是闭合的时，该源块内的数据的范围已经确定，并且其具有位于该视频数据流的上下文内的起始字节索引和结束字节索引。例如，第一源块可以开始于视频数据流内的字节索引 0，并且结束于字节索引 4,432，在该情况下，在该源块内存在包括索引 0 到 4,431 的字节索引的 4,432 个数据字节。继续这个例子，第二源块开始于字节索引 4,432，但其结束字节索引可能没有被确定，直到源块生成器单元 (1425) 在某个稍后的时间点进行确定为止，以及直到针对第二源块的结束字节索引已经被确定为止，第二源块被视作为是开放的。

[0153] 因而，通常，可以将视频数据流视作为闭合源块序列跟着至多一个开放源块，其中该开放源块处于被确定的过程之中。此外，该源块序列包括零个或多个不活动源块，其跟着一个或多个活动源块 (除非在传送的结束位置，此时已经成功地传送了完整的视频流)，其中不活动源块是那些已成功经成功传送给接收方传输协议 (1225)，并基于从接收方传输协议 (1225) 发送的反馈而确认已经成功传送到发送方传输协议 (1215) 的那些源块，或者是已经被认为太晚以至于没有传送，并因而不在要求在接收方处进行恢复的那些源块。源块生成器单元 (1425) 确定何时关闭最近的活动源块，从而开始新的活动开放源块。当使用系统性 FEC 编码时，即，当源块的源符号处于可以被 FEC 解码器用于恢复该源块的编码符号当中时，则可能的并且优选的是，允许发送针对活动开放源块的编码符号，特别是源符号。

[0154] 很多众所周知的 FEC 编码都是系统性的，例如，如 IETF RFC 5510 中所详细说明的里德 - 所罗门编码，或者如 IETF RFC 6330 中所详细说明的 RaptorQ 编码，或者 IETF RFC 5053 中详细说明的 Raptor 编码。发送针对活动开放源块的源符号是优选的，这是因为这可以减少视频流的传送的端到端延时，以及在相同的端到端延时预算内，提供更高质量和更可靠的传送。存在优点的一个原因在于，甚至在整个源块可用或者知道其大小之前，就可以开始传送该源块。一旦源块生成器单元 (1425) 关闭了该活动开放源块，FEC 编码器 (1410) 就可以生成针对这个源块的修复符号，并将其存储在修复符号缓冲区 (1415)，并根据发送方反馈逻辑单元 (1420) 方法，当要发送针对该源块的另外的编码符号时，可以发送这些修复符号。

[0155] 或者，可以不存在修复符号缓冲区 (1415)，并且当根据发送方反馈逻辑单元 (1420) 方法，要发送针对这个源块的另外的编码符号时，FEC 编码器 (1410) 可以飞速生成修复符号以立即进行传输。图 16 描述了不活动源块和活动源块，以及闭合源块（不活动源块和活动源块的混合）和至多一个开放源块（其是活动的）。

[0156] 源块生成器单元 (1425) 可以使用各种方法来确定何时关闭当前活动开放源块，并开始下一个活动开放源块。例如，当源块生成器单元 (1425) 从发送方反馈逻辑单元 (1420) 接收到信息时，源块生成器单元 (1425) 可以决定关闭当前活动开放源块，该信息指示发送方反馈逻辑单元 (1420) 已经从接收方反馈单元 (1525) 接收到指示在发送包含针对当前活动开放源块的编码符号的第一分组的时间点或该时间点之后发送的分组已经被接收的反馈。这个时间点可以由发送方反馈逻辑单元 (1420) 通过以下方式来确定 :在发送包含根据当前活动源块的编码符号的第一分组的时间，记录针对各个流的当前序列号，然后确定只要发送方反馈逻辑单元 (1420) 从接收方反馈逻辑单元 (1525) 接收到反馈，就提供对源块生成器单元 (1425) 的指示，其中针对流的最高序列号至少与该流在所述记录的时间的当前序列号一样高。

[0157] 使用这种方法，在当前活动开放源块闭合时，并且在确定该源块的大小时，该源块的大小近似是数据的 RTT 量。或者，在固定时间量之后（例如，前一个源块被关闭之后一秒），源块生成器单元 (1425) 可以确定关闭当前活动源块。在这种情况下，如果发送速率是可变的，则很可能每个源块将具有不同的大小，而如果发送速率是固定的时，则很可能源块将具有近似相等的大小。作为另一种选择，只要开放源块的大小达到预定的大小（例如，100,000 字节），源块生成器单元 (1425) 就可以关闭当前活动开放源块。

[0158] 作为其它替代方案，源块生成器单元 (1425) 可以使用上述方法的组合来关闭当前活动开放源块，例如，只要其大小已经达到预定的大小，或者自从前一个源块被关闭之后经过了预定的时间量（无论哪一个先发生），就关闭该源块。作为另一例子，当针对该源块的反馈的指示第一次从发送方反馈逻辑单元 (1420) 向源块生成器单元 (1425) 指示时，或者自从前一个源块关闭之后经过了固定的时间量之后（无论哪一个先发生），源块生成器单元 (1425) 可以关闭当前活动开放源块。

[0159] 图 14 的发送方反馈逻辑单元 (1420) 处置从图 15 的接收方反馈逻辑单元 (1525) 接收的反馈，其中该反馈以图 13 的底部所示出的格式来提供。发送方反馈逻辑单元 (1420) 基于所接收的最低活动 SBN(1370) 来更新活动源块的集合。发送方反馈逻辑单元 (1420) 确定何时发送该视频流的下一个数据分组，用于发送下一个数据分组的路径流，以及在该数据分组内发送的编码符号所根据的活动源块（或者多个源块）。发送方反馈逻辑单元 (1420) 可以确定发送下一个编码符号所根据的活动源块，如下所述。

[0160] 使用与先前所使用的类似的符号，针对 SBN = I 的每个活动源块，使 BI 是为了以某种期望水平的确定性来恢复源块 I 而需要进行接收的编码符号的数量。例如，使用里德 - 所罗门 FEC 编码（例如，如 IETF RFC 5510 中所描述的），BI 的值可以等于源块的源符号的数量，并且整个源块的恢复具有完全的确定性，而对于其它编码（例如，IETF RFC 6330 中所描述的 RaptorQ 编码）而言，BI 的值可以等于源块的源符号的数量，其具有几乎的确定性，并且 BI 的较大值允许提高的确定性。

[0161] 发送方反馈逻辑单元 (1420) 可以基于使用的 FEC 编码的属性，并基于源块 I 中的

源符号的数量，来计算 BI 的值。发送方反馈逻辑单元 (1420) 可以将 RI 计算成发送方反馈逻辑单元 (1420) 已经从接收方反馈逻辑单元 (1525) 接收的针对源块 I 的已接收编码符号 (1365) 的数量的最高值。发送方反馈逻辑单元 (1420) 可以计算 LI ＝ BI-RI，其是为了以某种指定程度的确定性来恢复源块 I，除了发送方知道的接收方已经接收的数量之外，接收方必须接收的另外的编码符号的数量。

[0162] 假设 UI 是针对源块 I 已经发送的、但在发送方处尚未从接收方接收到确认的编码符号的数量。发送方反馈逻辑单元 (1420) 可以如下所述地基于从接收方反馈逻辑单元 (1525) 接收的反馈来计算 UI。发送方反馈逻辑单元 (1420) 可以针对每个流 ID 值 J，确定针对源块 I 所发送的位于 FID ＝ J 的序列号范围之内、并且发送方反馈逻辑单元 (1420) 已经从接收方反馈逻辑单元 (1525) 接收到采用图 13 的底部所示出的反馈信息格式的编码符号的数量，其中该范围位于发送方针对 FID ＝ J 所发送的当前序列号 C 与用于 FID ＝ J 的最高序列号 S 之间。发送方反馈逻辑单元 (1420) 可以通过以下方式来执行这个计算：针对处于 FID ＝ J 的 S 到 C 的范围中的每个序列号 K，保存在具有 FID ＝ J 和序列号 K 的分组中携带多少针对源块 I 的编码符号。

[0163] 基于此，发送方反馈逻辑单元 (1420) 可以将发送给路径流 J 的位于序列号 S+1 到 C-1 的范围之内的针对源块 I 的这样的编码符号的数量进行求和。然后，发送方反馈逻辑单元 (1420) 可以将不同的路径流上的这些量进行求和，以确定针对源块 I，总共发送了多少还没有被确认的编码符号 UI。应当注意的是，使用用于各个流或路径的流标识符和流序列号以及本文所描述的方法在数据分组和在反馈信息中提供的信息使发送方得以针对每个路径准确地计算已发送但尚未被确认（丢失或者已接收）的编码符号的数量，因而使发送方得以准确地估计在所有路径上已发送但尚未被确认的编码符号的聚合数量。如果在分组被发送到一个路径的顺序和从该路径接收到该分组（如果没有丢失的话）的顺序之间的差值很小，则发送方估计准确性较高，通常都是这种情况。如先前所提及的，在不考虑在其上发送分组的路径的情况下，在多个路径上发送的分组的聚合的发送顺序和接收顺序可能是非常不同的。因而，提供和使用每一路径信息和反馈的益处之一是使发送方得以更准确地估计总共要发送多少数据，以使发送冗余数据最小化，并使该流的块恢复的端到端延时最小化。因而，当在多个并行网络路径上可靠地流式传输数据时，发送方反馈逻辑单元所接收的数据表示跟踪、报告和使用路径特定信息的例子，其中这些网络路径可能承受分组丢失和变化的数据吞吐量和路径延时。

[0164] 如前所述，假设 XI 是发送方反馈逻辑单元 (1420) 已确定能主动针对源块 I 发送的编码符号的数量。发送方可以基于指定的规则（例如，XI 是 BI 的某个固定部分，如 XI ＝ 0.05*BI，或者类似于先前所描述的那些规则的其它规则），计算 XI 的值。然后，如果 LI+XI-UI>0，则发送方反馈逻辑单元 (1420) 确定可以发送针对活动闭合源块 I 的另一编码符号。

[0165] 数据速率调整器单元 (1430) 针对每个流，确定何时可以发送针对该流的下一个数据分组。数据速率调整器单元 (1430) 与各个发射机 (1220(1)、1220(2) 等）进行通信以进行这个确定。例如，如果使用 Linux 或者其它类似于 Unix 的操作系统，则当发送 UDP 数据分组时，发射机 (1220(1)) 可以利用 TIOCOUTQ ioctl() 来确定其 UDP 发送队列的大小。通过监测发送队列大小，发射机 (1220(1)) 可以避免在发射机 (1220(1)) 中进行过度缓冲，

并且仅仅当发射机(1220(1))内部发送方队列较低时,才排队新的输出分组,因而避免了发射机(1220(1))累积太多的尚未被发送的数据。如果发射机(1220(1))发送队列保持为非空,但很小,则可以实现满额的发送吞吐量。因而,用于各个流的各发射机(1220(1))向数据速率调整器单元(1430)指示其何时可以接受另个数据分组以进行传输,其中在该时间点,数据速率调整器单元(1430)确定可以向与该发射机(1220(1))相关联的流发送数据分组。数据速率调整器单元(1430)从可能的发射机(1220(1)、1220(2)等)中的每个发射机接收指示,并使用上述方法,针对每个可能的流,确定何时可以发送针对该流的下一个数据分组。

[0166] 替代地或另外地,发射机(1220(1))可以使用SO_SNDBUF值,将发送方窗大小设置为足够小的值。然后,可以进行等待以便可以例如通过使用select()或poll()系统调用重写UDP套接字,以确定其何时可以发送与该发射机(1220(1))相关联的流的另一数据分组。以这种方式,不需要经常地轮询发射机发送方队列的大小。

[0167] 所有发射机(1220(1)、1220(2)等)可以与数据速率调整器单元(1430)进行通信,其中当发射机已经指出其具有容量来发送与该发射机相关联的流的另一个数据分组时,数据速率调整器单元(1430)确定向各个发射机(1220(1)、1220(2)等)发送哪些数据分组。发射机(1220(1)、1220(2)等)和数据速率调整器单元(1430)之间的通信可以发生在具有高带宽、低延时和低分组丢失率的局域网络上,因此如果该通信使用TCP,则其也能够完全地满足。在这样的设置下,发射机然后运行以下循环:

[0168] 无限地重复:

[0169] 1、进行等待,直到发射机发送队列较低(通过监测TIOCOUTQ)或者通过使用低发送缓冲区,如上面所解释的select(),或者二者。

[0170] 2、向数据速率调整器单元(1430)发送针对新分组的请求(例如,针对数据速率调整器单元(1430)的send()或write())。

[0171] 3、等待来自数据速率调整器单元(1430)的响应,其包括要发送的数据分组。(例如,使用select())

[0172] 4、通过UDP向网络(1222)发送该数据分组(例如,通过使用send()或write()系统调用)

[0173] 数据速率调整器单元(1430)执行以下操作:

[0174] 无限地重复:

[0175] 1、等待来自任何发射机的请求。(使用例如select())

[0176] 2、针对请求发送新数据分组的每个发射机,构造要发送的新数据分组,并将其提供给该发射机。

[0177] 数据速率调整器单元(1430)还可以向视频生成器(1205)提供信息,以便增加正在生成的视频流数据速率,减少正在生成的视频流数据速率,或者保持视频数据速率不变,其中这个信息可以是例如基于源数据缓冲区(1405)中的数据的量、以及数据速率调整器单元(1430)在沿着不同的流发送数据分组时所按照的聚合速率。

[0178] 如果存在可用于发送但尚未发送的源块I的至少一个源符号,则发送方反馈逻辑单元(1420)确定可以发送针对该活动开放源块I的另一编码符号。在数据速率调整器单元(1430)指示将发送携带编码符号的下一个数据分组的时间,发送方反馈逻辑单元(1420)

使用全部反馈和逻辑可用的传输信息，执行上面的全部计算（尽管发送方反馈逻辑单元(1420)可以在将发送下一个数据分组的时间之前的任何时间点，进行这些计算中的一些或者全部）。当数据速率调整器单元(1430)指示将向特定流发送下一个数据分组时，发送方反馈逻辑单元(1420)确定活动源块I，使得I是全部活动源块（其中，在如上所述由发送方反馈逻辑单元(1420)所确定的时间，可以发送针对这些活动源块的编码符号）之中的最低源块编号，然后将针对源块I的一个或多个编码符号放置到下一个数据分组中，并发送到该流。

[0179] 图14中的各种逻辑框中的任何一个或者全部逻辑框可以用硬件、软件、固件或者其组合来实现。当用软件或固件来实现时，应当理解的是，还可以提供必要的硬件，例如，一个或多个计算机可读介质包括用于执行所描述的功能的指令以及用于执行这些指令的一个或多个处理单元。

[0180] 以这种方式，图14表示包括一个或多个处理器的设备的例子，其中所述一个或多个处理器被配置为：在并行网络路径中的第一路径上发送针对第一块的第一编码单元；在发送了第一编码单元之后，在第一路径上发送针对第二块的第二编码单元；以及在发送了第二编码单元之后，在第一路径上发送针对第一块的第三编码单元。

[0181] 图15更详细地描绘了一种示例性多径流式传输接收方的框图。图1的接收端系统160中的任何一个或者全部可以包括类似于图15的多径流式传输接收方的组件。用于流1的相应接收方(1505(1))接收用于流1的发射机(1220(1))所发送的分组（如果它们在发送和接收之间没有丢失的话），针对其它流也是类似的。接收方传输协议(1225)将全部已接收的分组聚合到接收数据缓冲区(1505)中。FEC解码器(1510)被执行为恢复活动源块，其中已经接收到针对这些活动源块的足够的编码符号来恢复该源块，并且将所恢复的源块放置在所恢复的源数据缓冲区(1520)中，优选地以源块编号增加的顺序，使得所恢复的视频流(1230)与原始视频流(1210)具有相同的顺序。

[0182] 接收方反馈逻辑单元(1525)监测接收数据缓冲区(1505)中的已接收分组，并且采用例如图13的底部中所示出的格式来生成反馈信息，其中该反馈信息被发送到相应的发送方反馈逻辑单元(1420)。接收方反馈逻辑单元(1525)还可以基于针对一个源块所接收的全部数据分组当中的已接收的最大SBL(1318)值，并还可能基于用于指示该源块是开放的还是闭合的SBL标志，确定何时可以恢复该源块，因而当其确定可以恢复一个源块时，接收方反馈逻辑单元(1525)可以调用FEC解码器(1510)。通常，优选的是，如果接收方反馈逻辑单元没有接收到源块是闭合的指示（例如，如根据SBL标志所确定的（如果存在SBL标志的话，其指示该源块是闭合的），或者通过接收到携带该源块的修复符号的分组，其提供说明该源块是闭合的隐式指示），则接收方反馈逻辑单元(1525)不会宣称该源块是可恢复的。

[0183] 当可以恢复一个源块时，或者当由于例如端到端时间约束而要跳过一个源块时，接收方反馈逻辑单元(1525)可以将提供给发送方反馈逻辑单元(1420)的反馈信息中的最低活动SBN(1370)重置成合适的更高的SBN值。接收方反馈逻辑单元(1525)针对每个活动源块，确定接收的针对该源块的编码符号的数量。接收方反馈逻辑单元(1525)针对每个流，确定接收的针对该流的最高序列号。在连续的基础上，使用例如图13的底部所提供的接收方多径反馈信息格式，从接收方反馈逻辑单元(1525)向发送方反馈逻辑单元(1420)

发送这些信息中的全部。

[0184] 图 15 中的各种逻辑框中的任何一个或者全部逻辑框可以用硬件、软件、固件或者其组合来实现。当用软件或固件来实现时，应当理解的是，还可以提供必要的硬件，例如，一个或多个计算机可读介质包括用于执行所描述的功能的指令以及用于执行这些指令的一个或多个处理单元。

[0185] 以这种方式，图 15 表示包括一个或多个处理器的设备的例子，其中所述一个或多个处理器被配置为：在并行网络路径中的第一路径上接收针对第一块的第一编码单元；在接收到第一编码单元之后，在第一路径上接收针对第二块的第二编码单元；以及在接收到第二编码单元之后，在第一路径上接收针对第一块的第三编码单元。

[0186] 以这种方式，图 15 表示包括一个或多个处理器的设备的例子，其中所述一个或多个处理器被配置为：经由多个并行网络路径，从服务器设备接收经前向纠错的数据；确定在这些网络路径中的每个网络路径上的数据的丢失率；以及向服务器设备发送用于表示在这些网络路径中的每个网络路径上的数据的丢失率的数据。

[0187] 应当认识到的是，取决于例子，本文描述的技术中的任何技术的某些动作或事件可以以不同的顺序来执行，可以被添加、合并或完全省略（例如，对于技术的实践来说，并非全部描述的动作都是必要的）。此外，在某些例子中，可以同时执行（例如，通过多线程处理、中断处理或多处理器）而不是顺序执行动作或事件。

[0188] 在一个或多个例子中，可以用硬件、软件、固件、或其任意组合来实现所描述的功能。如果用软件来实现，则这些功能可以作为一条或多条指令或代码存储在计算机可读介质上、或者通过计算机可读介质发送、并由基于硬件的处理单元执行。计算机可读介质可以包括计算机可读存储介质，其与诸如数据存储介质的有形介质、或包括例如根据通信协议便于将计算机程序从一个地点传送到另一个地点的任何介质的通信介质相对应。以这种方式，计算机可读介质通常可以对应于 (1) 非暂时性的有形计算机可读存储介质或者 (2) 诸如信号或载波之类通信介质。数据存储介质可以是可以由一个或多个计算机或一个或多个处理器访问以取回用于本公开内容中描述的技术的实现的指令、代码和 / 或数据结构的任何可用介质。计算机程序产品可以包括计算机可读介质。

[0189] 通过举例而非限制的方式，这样的计算机可读存储介质可以包括 RAM、ROM、EEPROM、CD-ROM 或其它光盘存储装置、磁盘存储装置或其它磁存储设备、闪存、或者能够用于存储具有指令或数据结构形式的期望的程序代码并能够由计算机进行存取的任何其它介质。另外，任何连接都可以适当地称为计算机可读介质。例如，如果指令是使用同轴电缆、光纤光缆、双绞线、数字订户线 (DSL) 或者诸如红外线、无线电和微波之类的无线技术，从网站、服务器或其它远程源发送的，则所述同轴电缆、光纤光缆、双绞线、DSL 或者诸如红外线、无线电和微波之类的无线技术包括在所述介质的定义中。然而，应当理解的是，计算机可读存储介质和数据存储介质不包括连接、载波、信号或者其它暂时性介质，而是旨在针对于非暂时性、有形的存储介质。如本文所使用的，磁盘和光盘包括压缩光盘 (CD)、激光光盘、光盘、数字多功能光盘 (DVD)、软盘和蓝光光盘，其中，磁盘通常磁性地复制数据，而光盘则用激光来光学地复制数据。上面的组合也应当被包括在计算机可读介质的范围之内。

[0190] 指令可以由诸如一个或多个数字信号处理器 (DSP)、通用微处理器、专用集成电路 (ASIC)、现场可编程门阵列 (FPGA) 或者其它等同的集成或者分立逻辑电路之类的一个或

多个处理器来执行。因此，如本文所使用的，术语“处理器”可以指代适合于实现本文所描述的技术实施方式的任何前述结构或者任何其它结构。另外，在一些方面中，本文所描述的功能，可以在被配置用于编码和解码的专用硬件和／或软件模块内提供，或者并入到组合的编解码器中。另外，可以将这些技术全部地实现在一个或多个电路或者逻辑元件中。

[0191] 本公开内容的技术可以实现在各种各样的设备或装置中，包括无线手持机、集成电路（IC）或IC组（例如，芯片组）。在本公开内容中描述了各种组件、模块或单元，以强调被配置为执行所公开技术的设备的功能方面，但不必要求通过不同的硬件单元来实现。更确切地，如上文所描述的，各个单元可以并入编解码器硬件单元或者由包括如上文描述的一个或多个处理器的互操作硬件单元的集合结合适当的软件和／或固件提供。

[0192] 已经描述了各个例子。这些和其它例子落入所附权利要求的保护范围内。

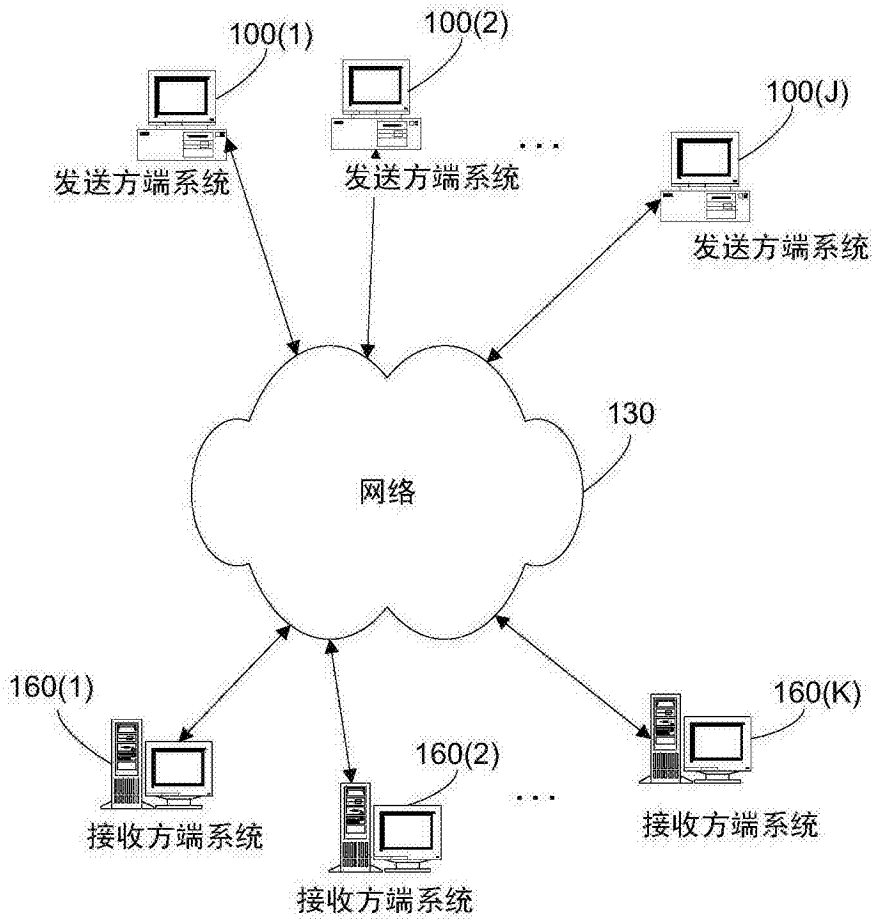

100(1)
100(2)
100(J)
发送方端系统
发送方端系统
发送方端系统
130
网络
160(1)
160(2)
160(K)
接收方端系统
接收方端系统
接收方端系统

图 1

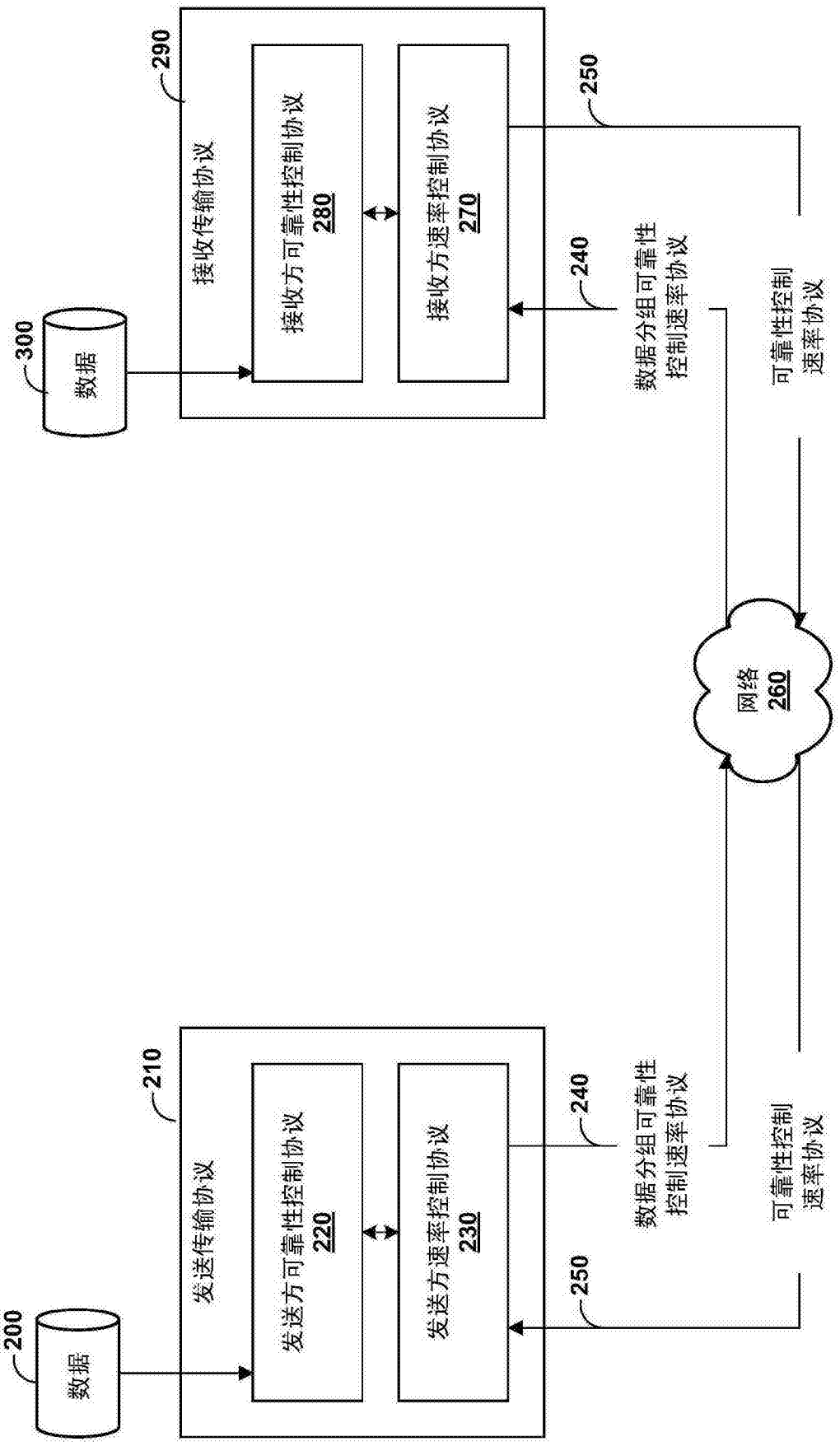

290
接收传输协议
接收方可靠性控制协议
280
接收方速率控制协议
270
250
240
300
数据
数据分组可靠性
控制速率协议
可靠性控制
速率协议
网络
260
210
发送传输协议
发送方可靠性控制协议
220
发送方速率控制协议
230
240
250
200
数据
数据分组可靠性
控制速率协议
可靠性控制
速率协议

图 2

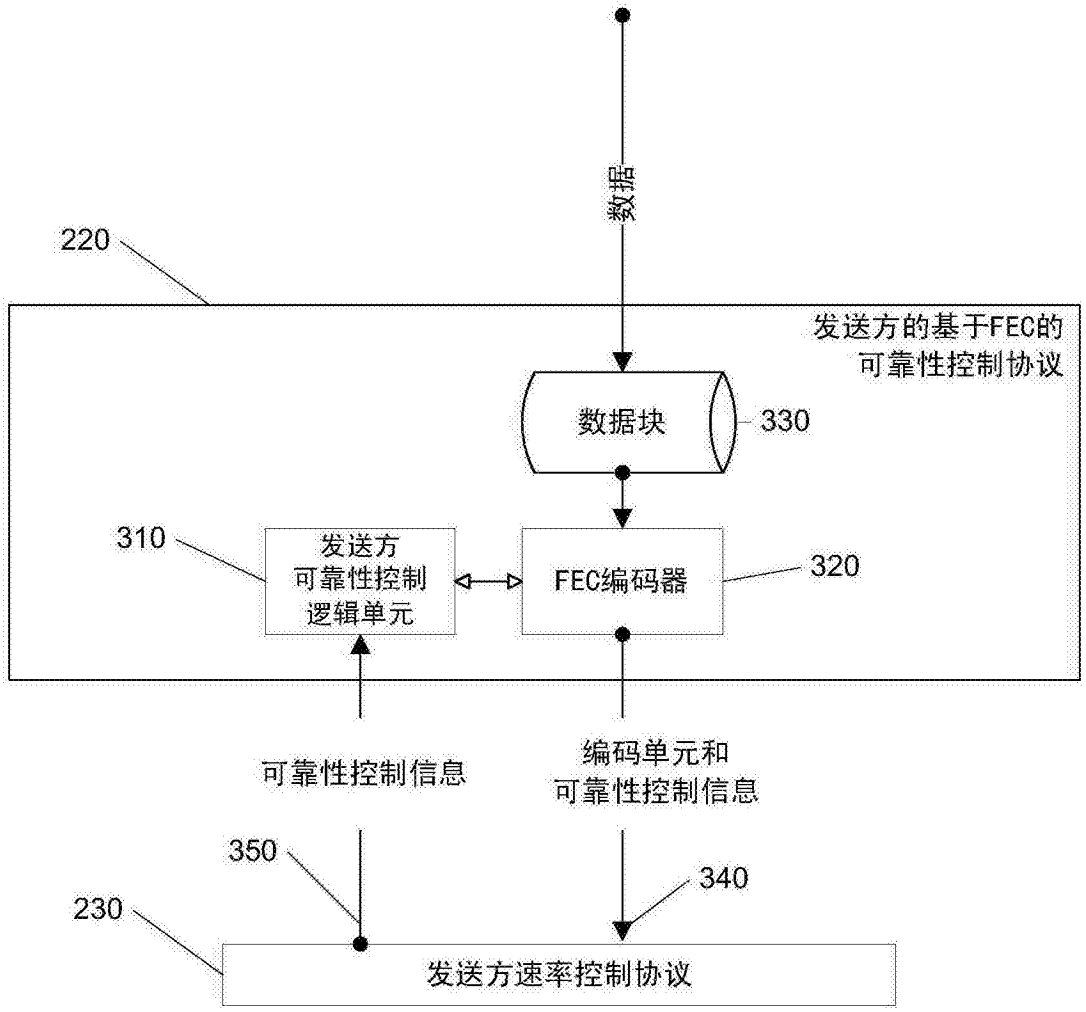

数据
220
发送方的基于FEC的
可靠性控制协议
数据块
330
310
发送方
可靠性控制
逻辑单元
FEC编码器
320
可靠性控制信息
编码单元和
可靠性控制信息
350
340
230
发送方速率控制协议

图 3

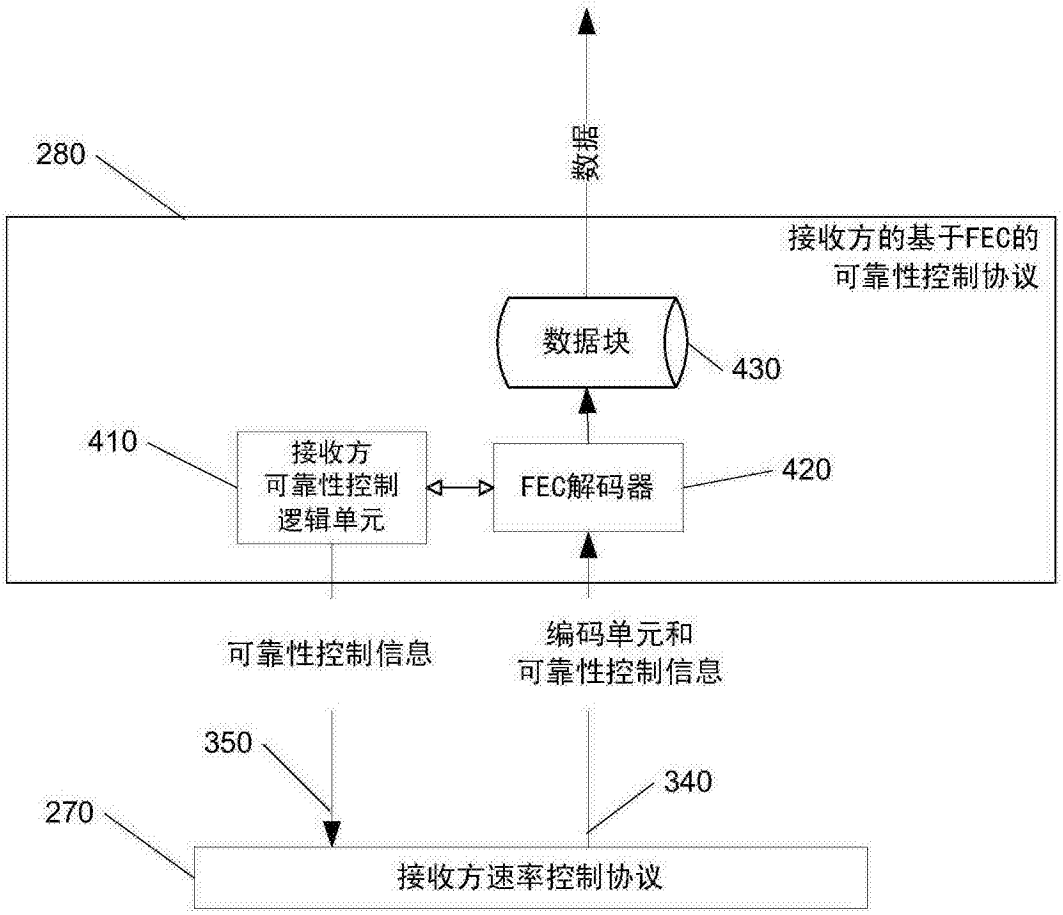
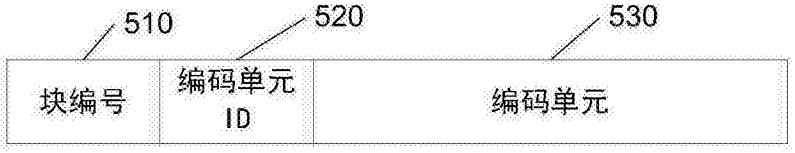
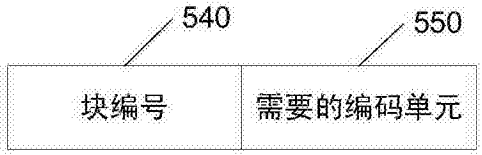


图 4



发送方数据格式



接收方反馈格式

图 5

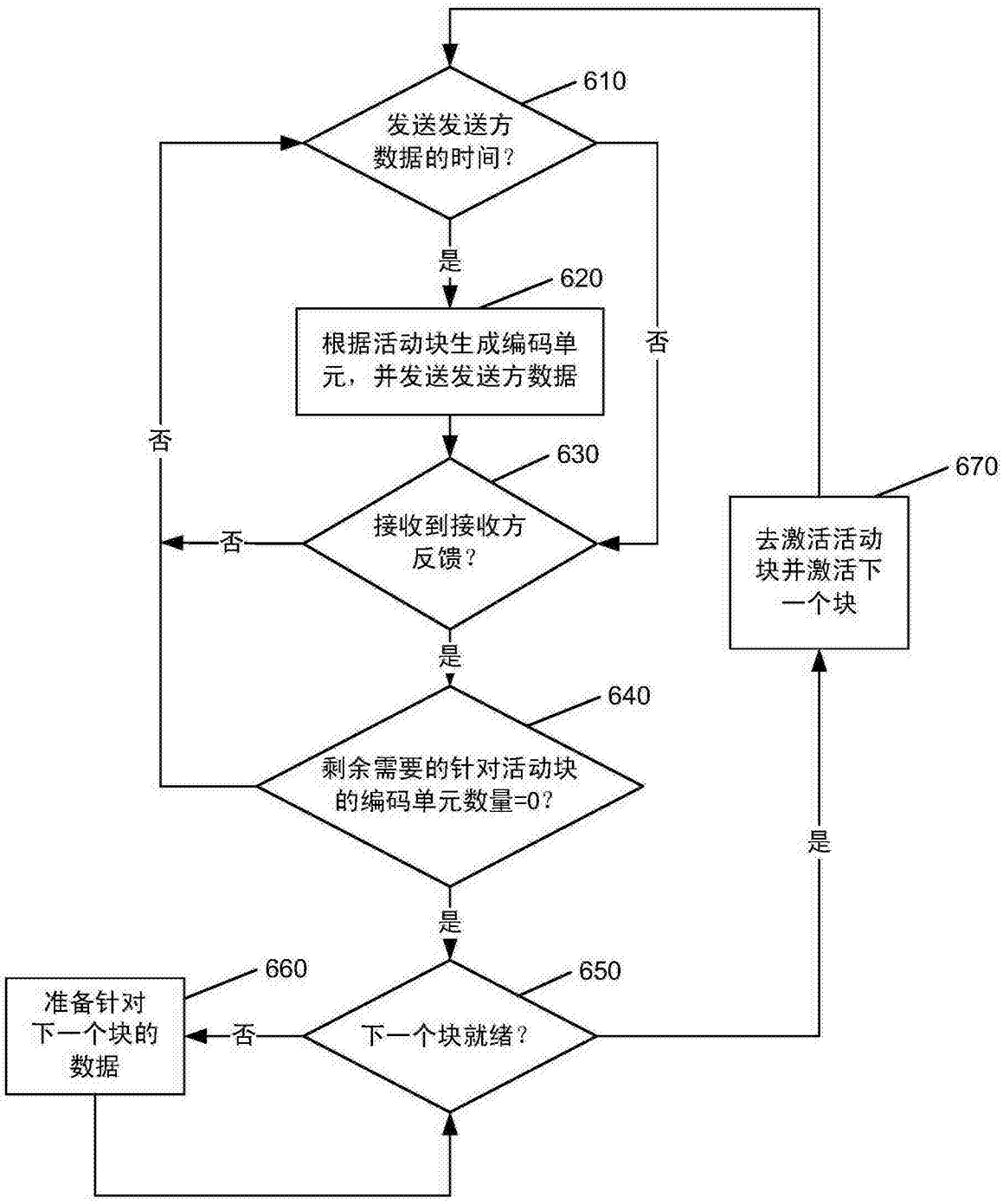

610
发送发送方数据的时间？
是
否
620
根据活动块生成编码单元，并发送发送方数据
630
接收到接收方反馈？
否
是
640
剩余需要的针对活动块的编码单元数量=0？
否
是
650
下一个块就绪？
否
是
660
准备针对下一个块的数据
670
去激活活动块并激活下一个块

图 6

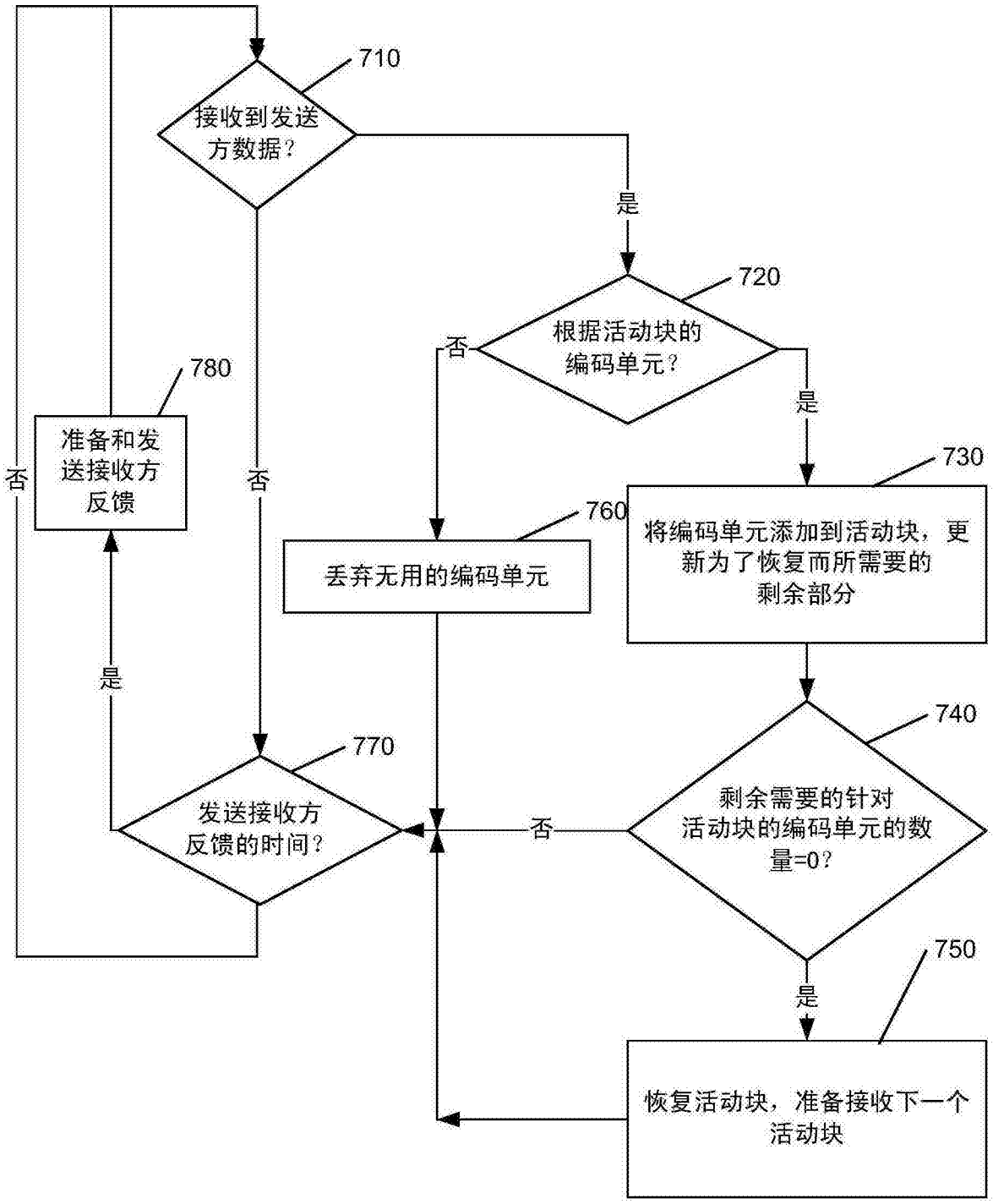
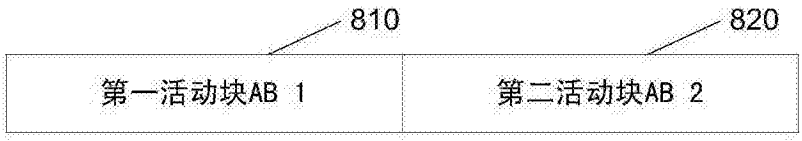

710
接收到发送方数据？
是
720
根据活动块的编码单元？
否
是
780
准备和发送接收方反馈
否
否
730
将编码单元添加到活动块，更新为了恢复而所需要的剩余部分
760
丢弃无用的编码单元
是
770
发送接收方反馈的时间？
否
740
剩余需要的针对活动块的编码单元的数量=0？
是
750
恢复活动块，准备接收下一个活动块

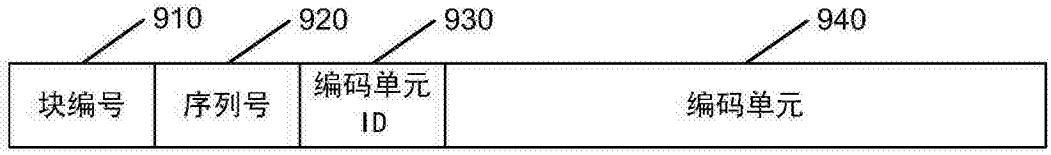
图 7



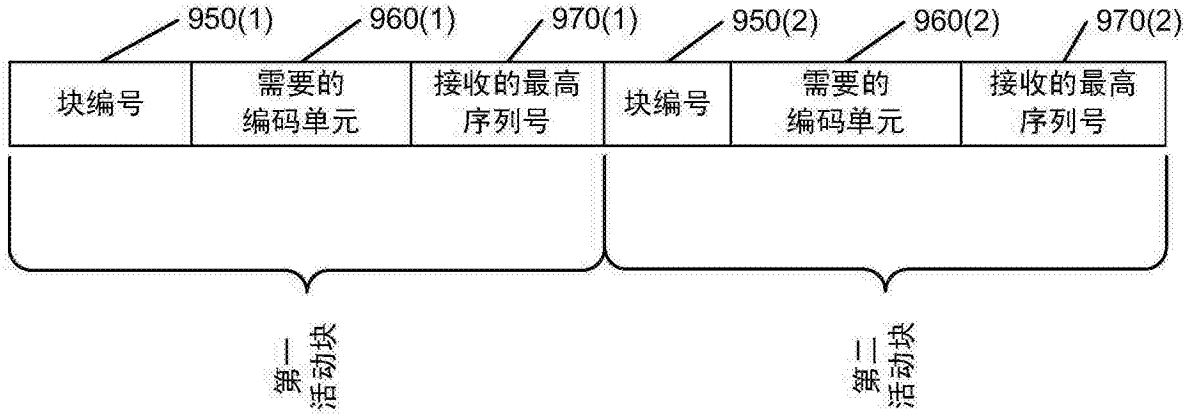
两个活动块

随着时间的示例性数据分组发送模式

图 8



发送方数据格式



接收方反馈格式

图 9

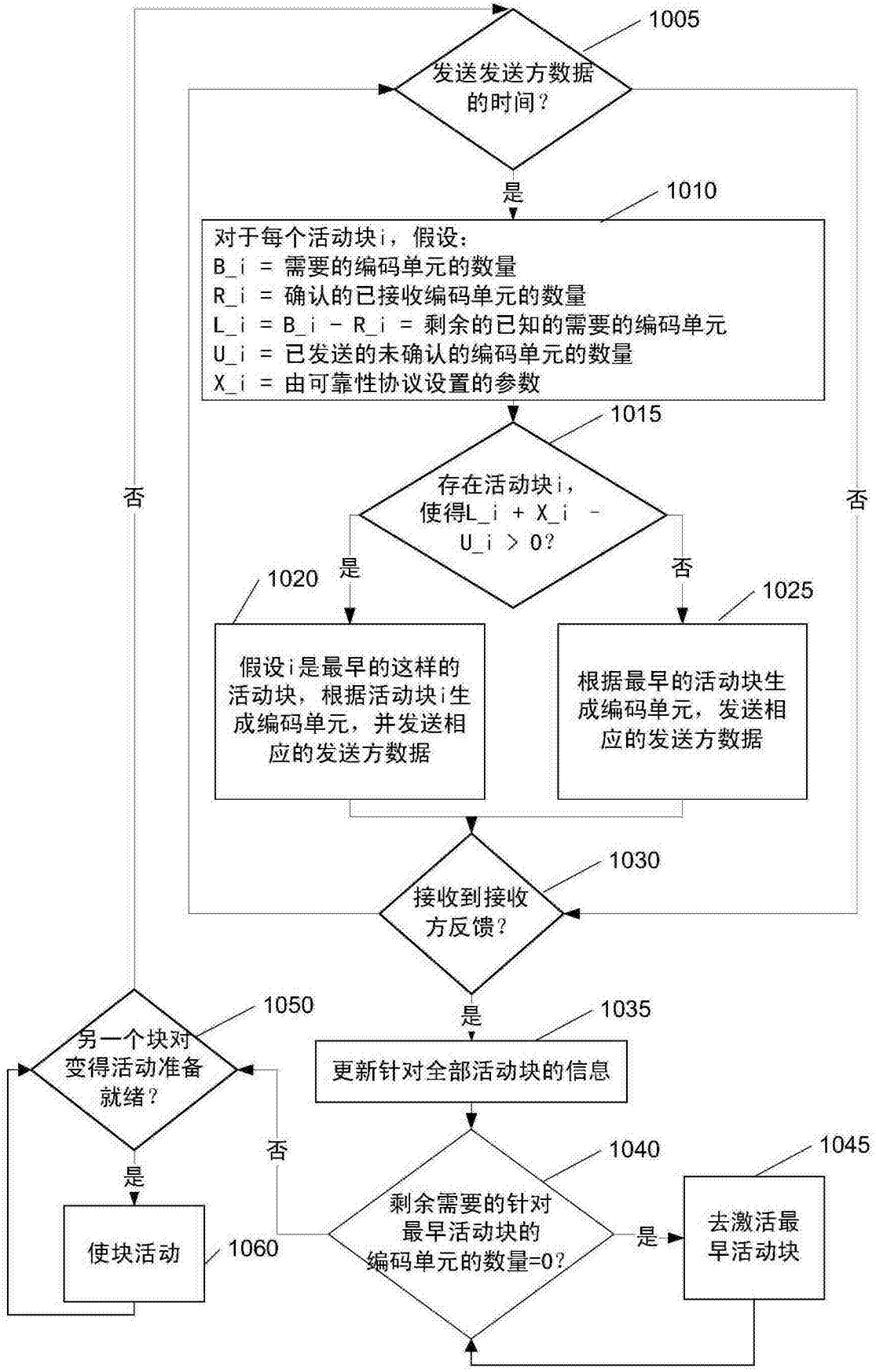

1005
发送发送方数据的时间？
是
1010
对于每个活动块i，假设：
B_i = 需要的编码单元的数量
R_i = 确认的已接收编码单元的数量
L_i = B_i - R_i = 剩余的已知的需要的编码单元
U_i = 已发送的未确认的编码单元的数量
X_i = 由可靠性协议设置的参数
1015
存在活动块i，使得L_i + X_i - U_i > 0？
否
是
否
1020
假设i是最早的这样的活动块，根据活动块i生成编码单元，并发送相应的发送方数据
1025
根据最早的活动块生成编码单元，发送相应的发送方数据
1030
接收到接收方反馈？
否
是
1035
更新针对全部活动块的信息
1040
剩余需要的针对最早活动块的编码单元的数量=0？
是
1045
去激活最早活动块
否
1050
另一个块对变得活动准备就绪？
是
1060
使块活动

图 10

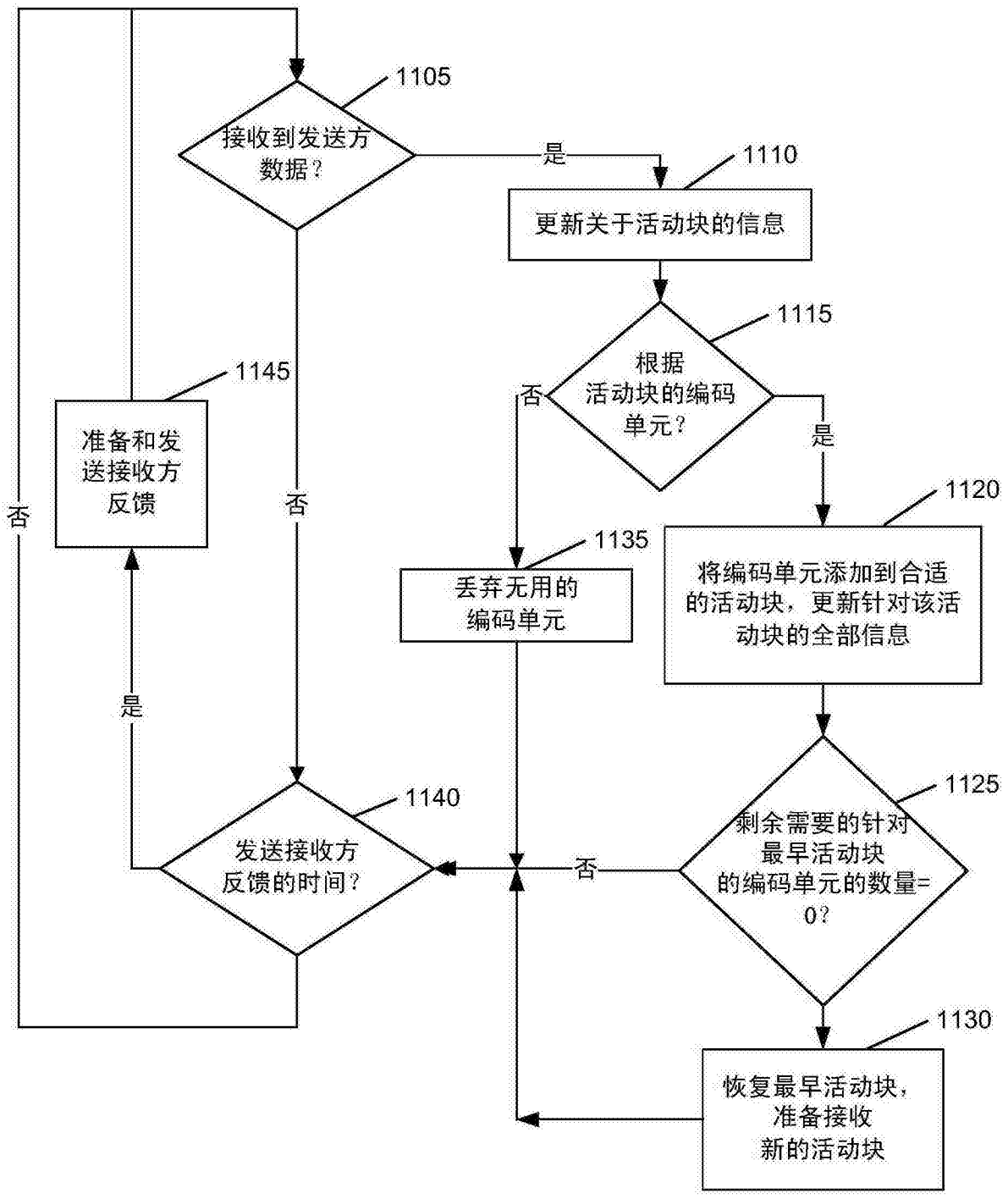

1105
接收到发送方数据？
是
1110
更新关于活动块的信息
1115
根据活动块的编码单元？
否
是
1120
将编码单元添加到合适的活动块，更新针对该活动块的全部信息
1125
剩余需要的针对最早活动块的编码单元的数量=0？
否
1130
恢复最早活动块，准备接收新的活动块
1135
丢弃无用的编码单元
1140
发送接收方反馈的时间？
否
是
1145
准备和发送接收方反馈
否
是

图 11

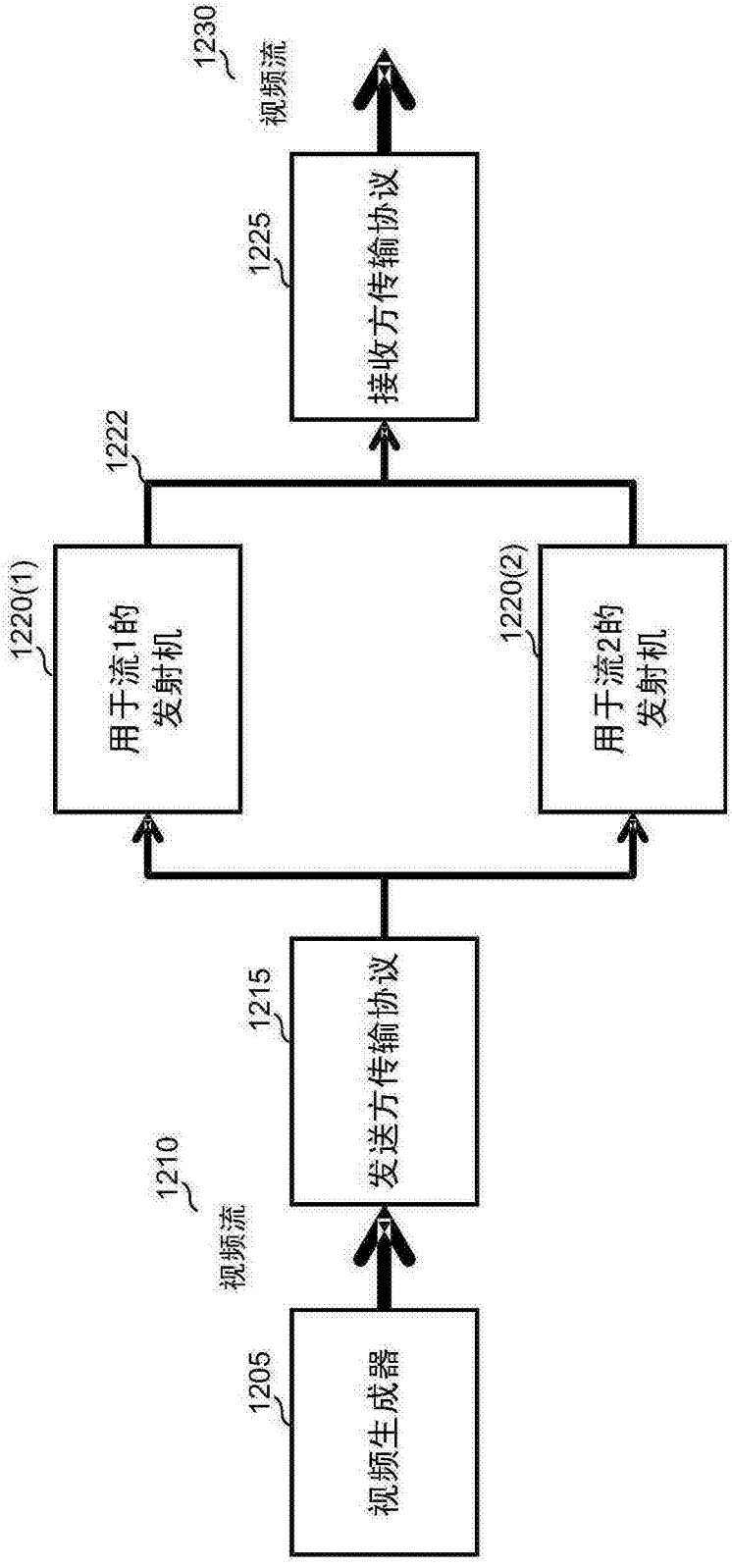

1230
视频流
1225
接收方传输协议
1222
1220(1)
用于流1的发射机
1220(2)
用于流2的发射机
1215
发送方传输协议
1210
视频流
1205
视频生成器

图 12

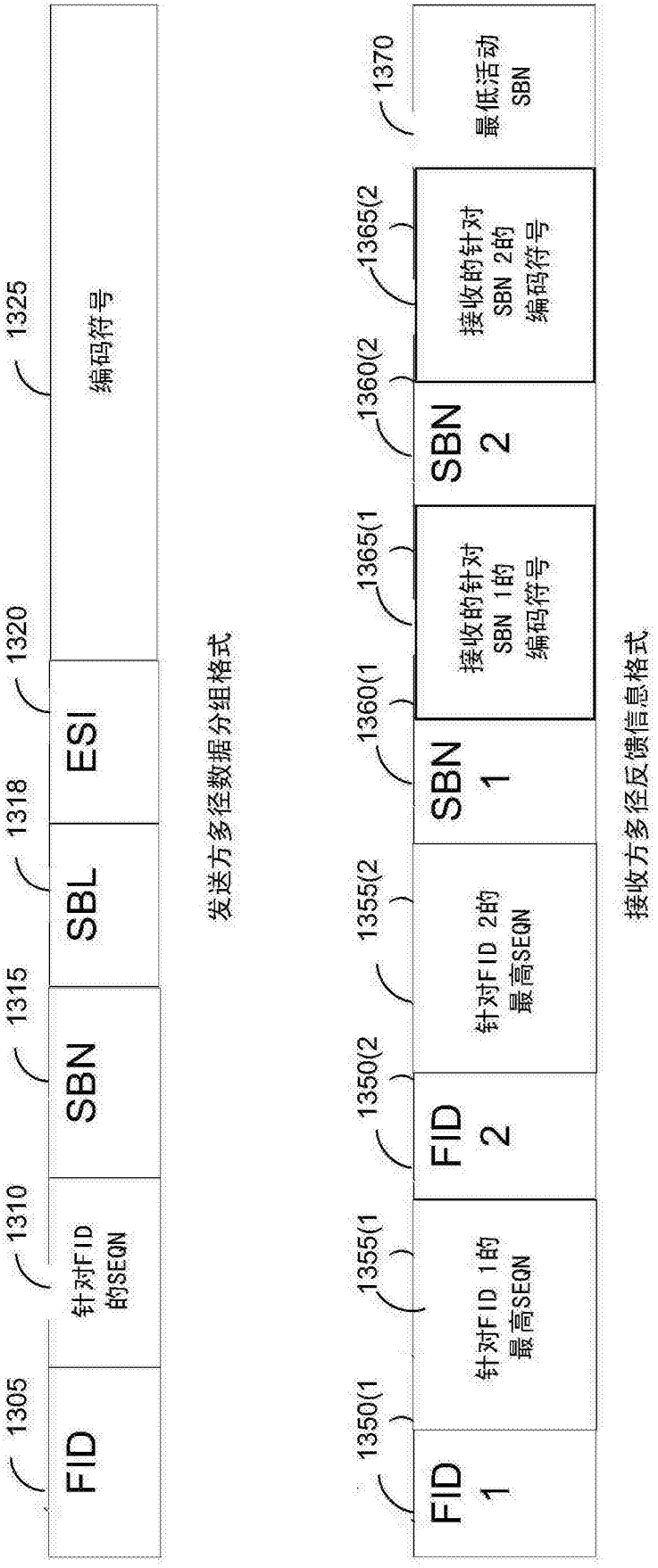

1305
FID
1310
针对FID的SEQN
1315
SBN
1318
SBL
1320
ESI
1325
编码符号
发送方多径数据分组格式
1350(1
FID 1
1355(1
针对FID 1的最高SEQN
1350(2
FID 2
1355(2
针对FID 2的最高SEQN
1360(1
SBN 1
1365(1
接收的针对SBN 1的编码符号
1360(2
SBN 2
1365(2
接收的针对SBN 2的编码符号
1370
最低活动SBN
接收方多径反馈信息格式

图 13

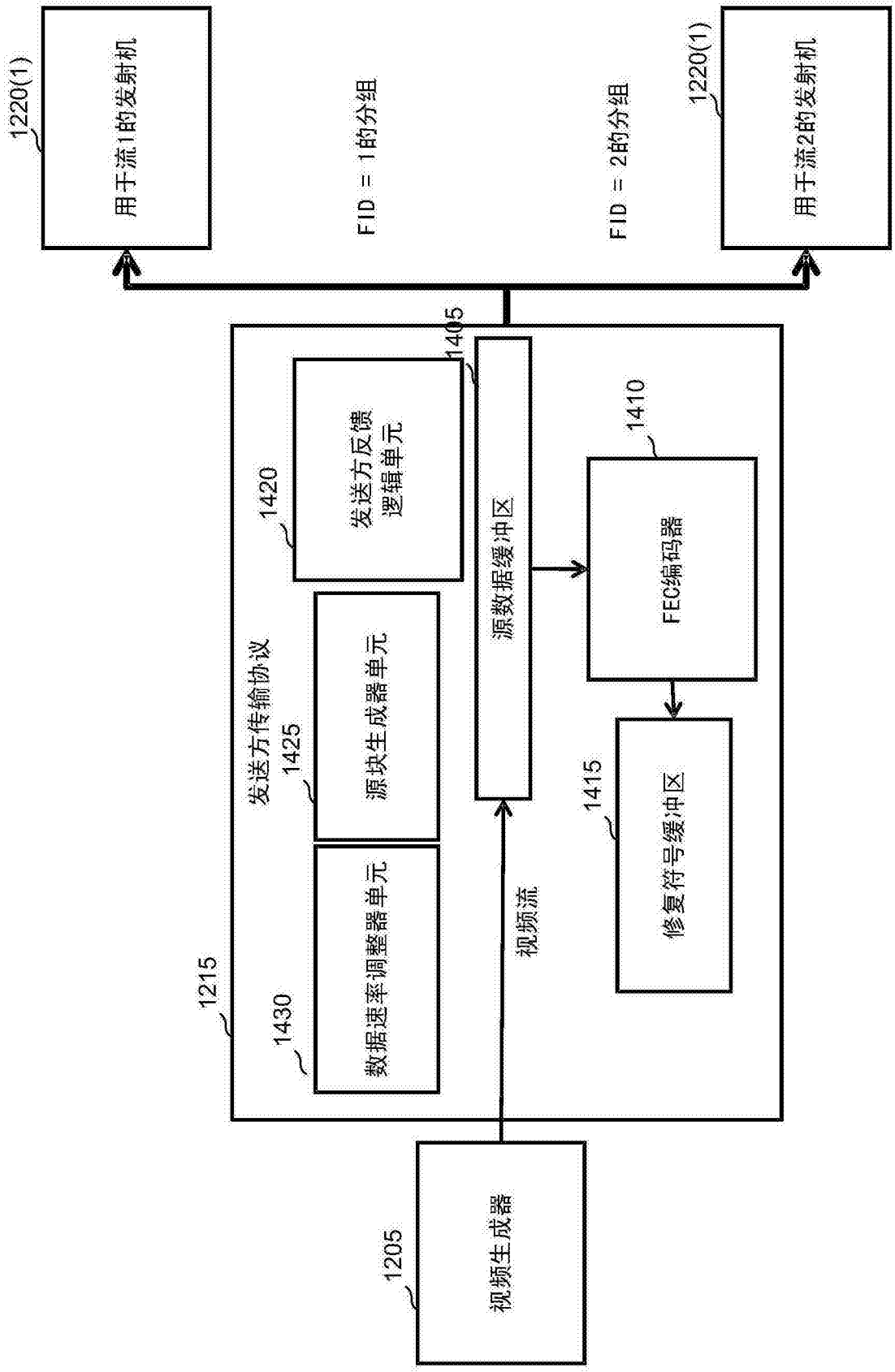

1220(1)
用于流1的发射机
FID = 1的分组
FID = 2的分组
1220(1)
用于流2的发射机
1405
1420
发送方反馈逻辑单元
1410
源数据缓冲区
FEC编码器
发送方传输协议
1425
源块生成器单元
1415
修复符号缓冲区
1215
1430
数据速率调整器单元
视频流
1205
视频生成器

图 14

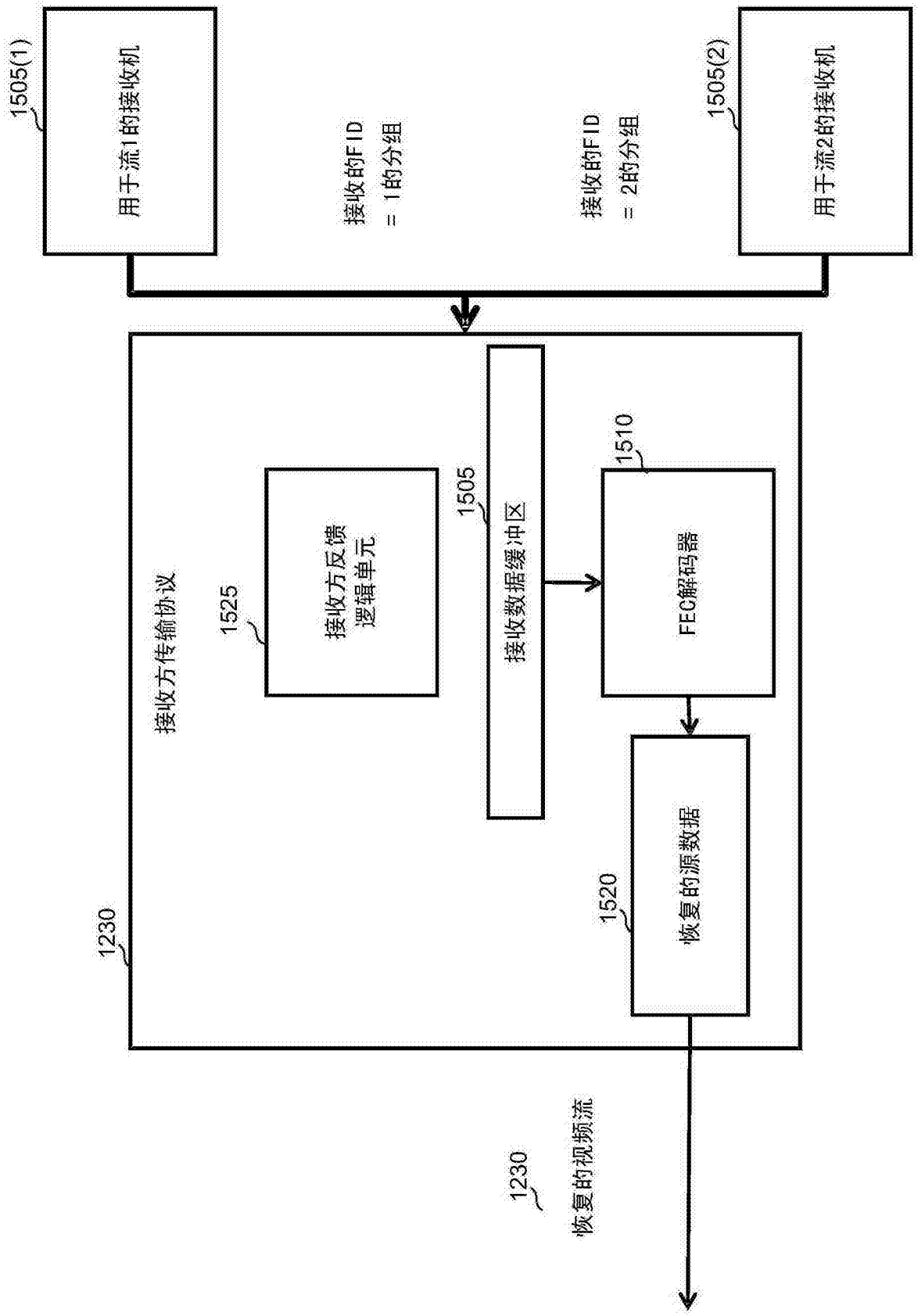

1505(1)
用于流1的接收机
接收的FID
= 1的分组
接收的FID
= 2的分组
1505(2)
用于流2的接收机
接收方传输协议
1525
接收方反馈
逻辑单元
1505
接收数据缓冲区
1510
FEC解码器
1520
恢复的源数据
1230
1230
恢复的视频流

图 15

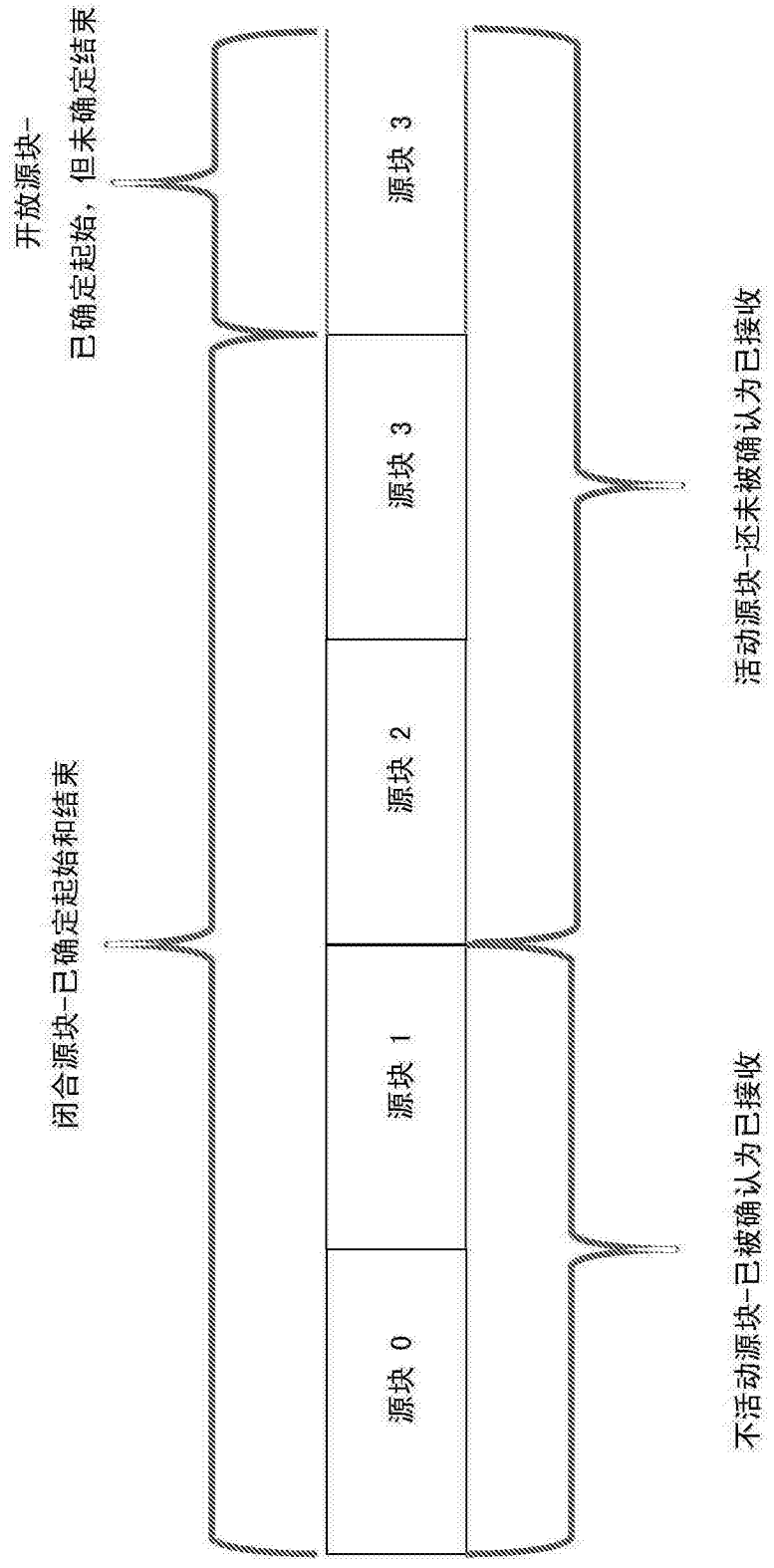

开放源块-
已确定起始，但未确定结束
闭合源块-已确定起始和结束
源块 0
源块 1
源块 2
源块 3
源块 3
不活动源块-已被确认为已接收
活动源块-还未被确认为已接收

图 16