(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104756451 B
(45)授权公告日 2018.05.29

(21)申请号 201380057367.0
(22)申请日 2013.10.08
(65)同一申请的已公布的文献号
申请公布号 CN 104756451 A
(43)申请公布日 2015.07.01
(30)优先权数据
13/664,192 2012.10.30 US
(85)PCT国际申请进入国家阶段日
2015.04.30
(86)PCT国际申请的申请数据
PCT/IB2013/059215 2013.10.08
(87)PCT国际申请的公布数据
WO2014/068426 EN 2014.05.08
(73)专利权人 瑞典爱立信有限公司
地址 瑞典斯德哥尔摩
(72)发明人 P.阿南德 A.巴拉钱德兰
(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 杨美灵 姜甜
(51)Int.Cl.
H04L 12/803(2006.01)
(56)对比文件
US 2011090789 A1,2011.04.21,
CN 101938781 A,2011.01.05,
US 2012207024 A1,2012.08.16,
US 2011205898 A1,2011.08.25,
US 2007230436 A1,2007.10.04,
US 2008198746 A1,2008.08.21,
审查员 任盈之

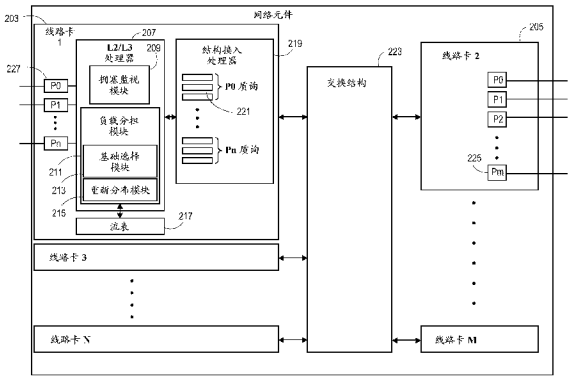
权利要求书3页 说明书9页 附图4页

(54)发明名称

用于LAG接口上网络流的动态负载平衡的方法

(57)摘要

一种方法由网络元件实现以通过将数据流重新分布到与链路聚合群组相关联的端口集中更不拥塞的端口，为链路聚合群组改进负载分担。网络元件在网络元件的入口端口接收数据流中的数据分组。执行负载分担过程以选择网络元件的出口端口。检查选择的出口端口是否拥塞。检查自收到数据流中前一数据分组以来的时间是否超过阈值。识别端口集中更不拥塞的出口端口。更新流表以将数据流绑定到更不拥塞的出口端口，并且将数据分组转发到更不拥塞的出口端口。



CN 104756451 B

1.一种由网络元件通过将数据流重新分布到与链路聚合群组相关联的端口集中更不拥塞的端口，为所述链路聚合群组改进负载分担而实现的方法，所述方法包括以下步骤：

在所述网络元件的入口端口接收(101)数据流中的数据分组；

执行(103)负载分担过程以选择所述网络元件的出口端口；

检查(105)所述选择的出口端口是否拥塞；

如果所述选择的出口端口拥塞，则检查(109)自收到所述数据流中前一数据分组以来的时间是否超过阈值；

如果所述时间超过所述阈值，则识别(111)所述端口集中更不拥塞的出口端口；

更新(117)流表以将所述数据流绑定到所述更不拥塞的出口端口；以及

将所述数据分组转发(119)到所述更不拥塞的出口端口。

2.如权利要求1所述的方法，还包括：

在所述流表中将所述数据分组的时戳进行存储(115)以便与随后的数据分组到达时间进行比较。

3.如权利要求1所述的方法，其中所述负载分担过程还包括以下步骤：

对所述数据分组的报头数据进行散列处理以生成标识符；以及

在所述流表中查找所述标识符以获得所述选择的出口端口。

4.如权利要求1所述的方法，其中检查所述选择的出口端口是否拥塞包括以下步骤：

检查拥塞数据库以获得量化的拥塞值或队列长度；以及

比较所述量化的拥塞值和另一阈值。

5.如权利要求1所述的方法，还包括以下步骤：

使用报告量化的拥塞值的拥塞通知过程，检查与所述选择的出口端口相关联的队列；以及

在拥塞数据库中记录所述量化的拥塞值。

6.如权利要求1所述的方法，还包括以下步骤：

检查与所述选择的出口端口相关联的队列的队列长度；以及

在拥塞数据库中记录所述队列长度。

7.如权利要求1所述的方法，还包括以下步骤：

检索所述数据流的所述前一数据分组的时戳；以及

比较所述前一数据分组的所述时戳和所述数据分组的时戳以确定自收到所述前一数据分组以来的所述时间。

8.如权利要求1所述的方法，其中识别所述端口集中所述更不拥塞的出口端口包括以下步骤：

选择具有最低拥塞测量值或最短队列长度的替换出口端口。

9.如权利要求1所述的方法，还包括以下步骤：

检查使用拥塞通知过程报告的、用于与所述选择的出口端口相关联的路由器的量化拥塞值；以及

在拥塞数据库中记录所述量化的拥塞值。

10.一种由网络元件通过将数据流重新分布到与链路聚合群组相关联的端口集中更不拥塞的端口，为所述链路聚合群组改进负载分担而实现的方法，所述方法包括以下步骤：

在所述网络元件的入口端口接收(301)数据流中的数据分组；

执行(305)负载分担过程以识别所述链路聚合群组的出口端口；

使用拥塞监视数据库，检查(307)所述识别的出口端口是否拥塞；

如果所述识别的出口端口拥塞，则检查(311)当前时间与所述数据流中前一数据分组的时戳之间的差是否超过阈值；

如果所述差超过所述阈值，则使用拥塞监视数据，选择(315)所述链路聚合群组的所述端口集中的新出口端口；

更新(319)流表以将所述数据流绑定到所述新出口端口；以及

将所述数据分组转发(321)到所述链路聚合群组的新出口端口。

11.一种通过将数据流重新分布到与链路聚合群组相关联的端口集中更不拥塞的端口，为所述链路聚合群组改进负载分担的网络元件，所述网络元件包括：

第一线路卡(205)，包括配置为所述链路聚合群组的一部分的端口集(225)；

耦合到所述第一线路卡的交换结构(223)，所述交换结构配置成允许在所述网络元件的线路卡之间的通信；以及

耦合到所述交换结构的第二线路卡(203)，所述第二线路卡包括端口集(227)、L2和L3处理器(207)及结构接入处理器(219)，

所述端口集(227)配置成接收数据流中的数据分组，

所述L2和L3处理器(207)配置成：执行负载分担过程以识别所述链路聚合群组的出口端口；使用拥塞监视数据库检查所述识别的出口端口是否拥塞；如果所述识别的出口端口拥塞，则检查当前时间与所述数据流中前一数据分组的时戳之间的差是否超过阈值；如果所述差超过所述阈值，则使用拥塞监视数据选择所述链路聚合群组的所述端口集中的新出口端口；并且更新流表以将所述数据流绑定到所述新出口端口，以及

所述结构接入处理器(219)用于将所述数据分组转发到所述链路聚合群组的新出口端口。

12.如权利要求11所述的网络元件，还包括：

耦合到所述L2和L3处理器的所述流表(217)，所述流表存储所述数据分组的时戳以便与随后的数据分组到达时间进行比较。

13.如权利要求11所述的网络元件，其中所述L2和L3处理器还配置成对所述数据分组的报头数据进行散列处理以生成标识符，并且在所述流表中查找所述标识符以获得所述识别的出口端口。

14.如权利要求11所述的网络元件，其中所述L2和L3处理器还配置成通过检查拥塞数据库以获得量化的拥塞值或队列长度，并且比较所述量化的拥塞值和另一阈值，检查所述识别的出口端口是否拥塞。

15.如权利要求11所述的网络元件，其中所述L2和L3处理器还配置成使用报告量化的拥塞值的拥塞通知过程，检查与所述识别的出口端口相关联的队列，并且在拥塞数据库中记录所述量化的拥塞值。

16.如权利要求11所述的网络元件，其中所述L2和L3处理器还配置成检查与所述识别的出口端口相关联的队列的队列长度，并且在拥塞数据库中记录所述队列长度。

17.如权利要求11所述的网络元件，其中所述L2和L3处理器还配置成检索所述数据流

的所述前一数据分组的时戳，并且比较所述前一数据分组的所述时戳和所述数据分组的时戳以确定自收到所述前一数据分组以来的所述时间。

18.如权利要求11所述的网络元件，其中所述L2和L3处理器还配置成通过选择具有最低拥塞测量值或最短队列长度的替换出口端口，识别所述端口集中所述更不拥塞的出口端口。

19.如权利要求11所述的网络元件，其中所述L2和L3处理器还配置成检查使用拥塞通知过程报告的、用于与所述识别的出口端口相关联的路由器的量化拥塞值，并且在拥塞数据库中记录所述量化的拥塞值。

用于LAG接口上网络流的动态负载平衡的方法

技术领域

[0001] 本发明的实施例涉及用于负载平衡的方法和系统。具体地说，实施例涉及用于通过链路聚合接口重新分布数据流以改进吞吐量的方法和系统。

背景技术

[0002] 链路聚合是在计算机连网领域中用于描述平行(即，以聚合方式)使用多个连接(即，链路)增大网络元件的吞吐量的过程。使用多个网络连接以替代单个网络连接，这提供了比单个连接能够保持的更高的吞吐量。链路聚合的使用也提供了冗余以防止链路之一失效。就单链路失效而言，可减少吞吐量，但将不丢失连接性，这是因为其它链路继续服务于在其中支持链路聚合的来源与目的地节点之间的通信。

[0003] 如电气和电子工程师协会(IEEE)标准802.3ad中所述的链路聚合将网络元件的多个物理端口绑定成称为链路聚合群组(LAG)的单个更大容量逻辑端口。在进入数据流与特定网络元件中LAG之间的接口能够称为LAG接口。LAG能够服务于任何数量的数据流(即，一般在特定来源节点与目的地节点之间的有关数据分组集)。通常，通过散列函数跨LAG接口的构成链路分布数据流。到散列函数的输入是N元组，N元组从像第2级和第3级(L2/L3)报头字段等分组的一些固定属性推导。其中，第2级和第3级指开放系统互连(OSI)模型。一旦数据流绑定到LAG的输出端口，直到它处于活动状态，它都保持与该端口相关联。散列方法和固定数据流输出端口绑定确保不存在由于通过聚合接口的传送造成的分组重新排序。

发明内容

[0004] 在一个实施例中，一种方法由网络元件实现以通过将数据流重新分布到与链路聚合群组相关联的端口集中更不拥塞的端口，为链路聚合群组改进负载分担。网络元件在网络元件的入口端口接收数据流中的数据分组。执行负载分担过程以选择网络元件的出口端口。检查选择的出口端口是否拥塞。检查自收到数据流中前一数据分组以来的时间是否超过阈值。识别端口集中更不拥塞的出口端口。更新流表以将数据流绑定到更不拥塞的出口端口，并且将数据分组转发到更不拥塞的出口端口。

[0005] 在另一实施例中，另一方法由网络元件实现以通过将数据流重新分布到与链路聚合群组相关联的端口集中更不拥塞的端口，为链路聚合群组改进负载分担。在此方法中，在网络元件的入口端口接收数据流中的数据分组。执行负载分担过程以选择链路聚合群组的出口端口。使用拥塞监视数据库，检查识别的出口端口是否拥塞。检查在数据流中当前时间与前一数据分组的时戳之间的差是否超过阈值。使用拥塞监视数据，选择链路聚合群组的端口集中的新出口端口。更新流表以将数据流绑定到新出口端口，并且将数据分组转发到链路聚合群组的新出口端口。

[0006] 在一个实施例中，一种网络元件通过将数据流重新分布到与链路聚合群组相关联的端口集中更不拥塞的端口，为链路聚合群组改进负载分担。网络元件包括：第一线路卡，包括配置为链路聚合群组的一部分的端口集；耦合到第一线路卡的交换结构(switch

fabric)，交换结构配置成允许在网络元件的线路卡之间的通信；以及耦合到交换结构的第二线路卡，第二线路卡包括端口集、L2和L3处理器及结构接入处理器(fabric access processor)。端口集配置成接收数据流中的数据分组。L2和L3处理器配置成执行负载分担过程以选择链路聚合群组的出口端口，使用拥塞监视数据库检查识别的出口端口是否拥塞，检查在数据流中当前时间与前一数据分组的时戳之间的差是否超过阈值，使用拥塞监视数据选择链路聚合群组的端口集中的新出口端口，并且更新流表以将数据流绑定到新出口端口。结构接入处理器配置成将数据分组转发到链路聚合群组的新出口端口。

附图说明

[0007] 本发明通过示例方式而不是限制的方式在附图的图形中示出，图中相似的标号表示类似的元件。应注意的是，在此公开内容中对“一”或“一个”实施例的不同引用不一定是指相同的实施例，并且此类引用是指至少一个。此外，在结合实某个实施例描述某个特定特征、结构或特性时，认为结合无论是否明确描述的其它实施例来实现此类特征、结构或特性是在本领域技术人员的认知之内。

[0008] 图1是用于为链路聚合群组实现负载平衡的过程的一个实施例的流程图。

[0009] 图2是实现负载平衡和链路聚合群组的网络元件的一个实施例的图。

[0010] 图3是用于为链路聚合群组实现负载平衡的过程的另一实施例的流程图。

[0011] 图4是分布式链路聚合群组的一个示例实施例的图。

具体实施方式

[0012] 在下面的描述中，陈述了许多特定细节。然而，要理解的是，实践本发明的实施例可无需这些特定细节。在其它情况下，公知的电路、结构和技术未详细显示以免混淆对此描述的理解。其它情况下，控制结构、门级电路和全软件指令序列未详细示出以免混淆本发明。通过包括的描述，本领域技术人员将能够在不进行不当实验的情况下实现适当的功能性。

[0013] 在下面的说明和权利要求中，可使用术语“耦合”和“连接”及其衍生词。应理解，这些术语无意作为彼此的同义词。“耦合”用于指示可相互直接物理或电接触或不直接物理或电接触的两个或更多个元件相互协作或交互。“连接”用于指示在相互耦合的两个或更多个元件之间通信的建立。

[0014] 为便于理解实施例，虚线在图中用于表示某些项目的可选性质(例如，本发明的给定实施例不支持的特征；给定实施例支持但在一些情况下使用并且在其它情况下不使用的特征)。

[0015] 图中所示技术能使用在一个或更多个电子装置上存储和执行的代码和数据实现。此类电子装置使用非暂时性计算机可读存储介质(例如，磁盘、光盘、只读存储器、闪存存储器装置、相变存储器)和暂时性计算机可读通信介质(例如，电气、光学、声学或其它形式传播信号 - 如载波、红外信号、数字信号)存储和传递(在内部和/或通过网络与其它电子装置)代码和数据。另外，此类电子装置一般情况下包括与诸如存储装置、一个或更多个输入/输出装置(例如，键盘、触摸屏和/或显示器)和网络连接等一个或更多个其它组件耦合的一个或更多个处理器的集合。处理器的集合与其它组件的耦合一般情况下是通过一个或更多

个总线或桥接器(也称为总线控制器)。存储装置和携带网络业务的信号分别表示一个或更多个非暂时性有形计算机可读存储介质和暂时性计算机可读通信介质。因此,给定电子装置的存储装置一般情况下存储代码和/或数据以便在该电子装置的一个或更多个处理器的集合上执行。当然,本发明的实施例的一个或更多个部分可使用软件、固件和/或硬件的不同组合来实现。

[0016] 以下缩略词在本文中经常使用,并且在此提供以便参考:链路聚合群组(LAG)、虚拟输出排队(VOQ)、量化的拥塞通知(QCN)、拥塞点(CP)、拥塞通知标记(CNTAG)及拥塞通知消息(CNM)。

[0017] 本发明的实施例提供用于在交换机和路由器上的链路聚合接口的更佳带宽利用级别、用于交换机和路由器的增大吞吐量、更佳的网络性能及更高的用户满意度。实施例提供与新兴连网趋势一致的这些改进,包括更高的带宽要求、更多分组化、增大的机器到机器(M2M)业务、频繁的视频业务爆发、扩展的基于因特网协议(IP)的服务和增大在服务器与最终用户节点之间突发业务的频率的类似连网趋势。

[0018] 现有技术的缺点包括虽然链路聚合组合了端口,但情况不一定是吞吐量将与端口计数呈线性缩放。视数据业务的性质而定,数据流可未跨链路聚合群组的所有链路均匀分布。由于LAG中的一些链路拥塞,而其它链路利用不足,结果是跨LAG的不平衡负载分布,造成次佳的性能。

[0019] 以前的负载平衡过程通过只考虑数据流的报头字段(N元组),跨LAG的成员链路分布数据流。一般情况下,用于每个数据流的LAG中的出口端口由N元组所输入到的散列函数确定。此方法确保流的连续分组在流的持续时间内绑定到相同成员链路。因此,它防止数据流的分组失序到达目的地。

[0020] 以前的过程的缺陷是数据流分布未考虑LAG中链路的利用级别。可能的情况是,以前的过程中多个数据流绑定在相同链路上,而其它链路相对利用不足。这种不均匀的分布造成LAG的次佳性能,这是因为绑定到拥塞链路的数据流具有比它们如果被指派到LAG的利用不足的链路则将可能具有的吞吐量更低的吞吐量。这造成了LAG的效率和吞吐量的总体降低。

[0021] 本发明的实施例通过检测在LAG的成员链路上不平衡的负载条件,并且将映射到过载链路上的数据流移到LAG的相对利用更不足的链路,克服了现有技术发明内容的缺点。在过载的链路将拥塞时,通过使用量化的拥塞通知算法(例如,IEEE 802.1Qau中定义的QCN),通过队列长度监视或类似过程,能够检测到链路上的拥塞。响应检测到与数据流绑定到的链路有关的拥塞,实施例识别更不拥塞的链路并且将数据流迁移到该链路。为确保有序到达,为通过每个链路发送的最后数据分组保持时戳或类似机制。如果在当前时间前超过指定时间量发送了数据流的最后数据分组,则假设它已到达,并且数据流能够在另一链路上发送下一分组而不顾及数据分组的失序到达。

[0022] 在诸如VOQ的系统等特定实施例中,外部端口队列拥塞转移到调度的结构系统中入口线路卡的VOQ端口。就变得拥塞的输出端口而言,拥塞将同样在网络元件的相应线路卡上每个结构接入处理器(FAP)上的所有VOQ上显现。FAP能够实现诸如CP监视等QCN功能性,以生成QCN拥塞通知消息,指示跨不同端口的相对拥塞。在其它实施例中,通过检查队列长度或者通过类似度量,也能够轮询VOQ以感应拥塞。在带有端口绑定数据的入口线路卡上能

够保持聚合的流状态。

[0023] 图1是用于为链路聚合群组实现负载平衡的过程的一个实施例的流程图。在一个实施例中，响应在入口端口接收数据分组的流中的数据分组，启动过程(框101)。入口端口是网络元件的物理或虚拟端口。入口端口能够是离散线路卡的一部分，或者能够是由网络元件的网络处理器或其它组件提供的端口服务。入口端口能够服务于与另一网络元件或其它计算装置的任何链路。数据流的来源能够通过链路直接或间接与网络元件连接。

[0024] 能够执行负载分担过程以选择网络元件的出口端口(框103)。能够结合在流表中的查找来执行负载分担过程以确定分组的目的地。能够使用在收到数据分组的报头中的L2和/或L3数据，确定目的地。流表包含用于数据流的绑定，从而使每个数据流和与链路相关联的特定出口端口有联系，而通过该链路能够到达目的地。就LAG已建立以服务于特定目的地或下一跳的情况而言，能够选择LAG的链路集的任何链路，并且负载分担过程通过链路集分布数据流。在一个实施例中，数据分组能够是数据流的第一、最后或任何数据分组，并且过程保持相同。在其它实施例中，在第一数据分组上建立并且在最后数据分组上删除绑定，同时在第一数据分组上选择并且之后在流表中查找出口端口。

[0025] 负载分担过程能够在诸如L2和L3报头数据等数据分组的报头信息或在数据流的每个数据分组中保持不变的任何数据内执行散列。散列值被映射或编排索引，从而选择LAG的一致对应出口端口及其相关联的链路。响应收到的前一数据分组，负载分担的过程也能够进行检查以确定与数据分组相关联的数据流是否已经被重新指派到更不拥塞的出口端口和链路。此信息能够存储在流表中，并且在此情况下，不需要散列或类似过程。

[0026] 能够检查选择的出口端口及其相关联的链路是否受拥塞影响(框105)。出口端口拥塞的检查能够基于相对于负载分担过程异步或同步收集的量化的拥塞数据。拥塞数据能够通过QCN过程，通过队列长度监视或类似拥塞监视提供。被监视的拥塞点能够是与LAG的每个出口端口相关联的线路卡的队列或虚拟队列。这些线路卡队列能够是在入口线路卡或出口线路卡上。入口线路卡拥塞度量的使用降低了在聚集拥塞数据方面的开销。下面描述又一实施例，其中，从网络元件外部的来源收集拥塞数据。

[0027] 如果选择的出口端口不拥塞，则分组能够被指派到入口线路卡的相关联队列，以便通过网络元件的交换结构，转发到出口端口的线路卡(框107)。如本文中下面进一步所述，数据分组的时戳能够存储在流表、单独的数据结构或数据库中或类似的位置中(框113)。在此情况下，数据流保持绑定到当前出口端口和链路。这能够是原来选择的出口端口或者响应拥塞的以前检测而选择的出口端口。

[0028] 如果当前选择的出口端口拥塞，则检查自数据流中前一数据分组以来的时间是否超过指定的阈值(框109)。在入口端口或入口线路卡收到的每个数据分组能够具有时戳或记录的其到达时间的类似指示。时戳能够存储在流表或单独的数据结构或数据库中，并且与数据流相关联，替代前一数据分组的时戳。这样，保持了由入口线路卡转发的最后数据分组的时戳。阈值能够是指示自最后转发的分组以来经过了足够的时间以确保其到达的任何值。数据流可要求有序分组到达，并且时间的经过能够确保当前数据分组未在备选链路上发送并且在前一数据分组前到达目的地节点。

[0029] 如果阈值尚未超过，则数据分组和数据流不能安全地重新路由到LAG的另一链路上。在此情况下，尽管存在拥塞，仍将数据分组转发到当前选择的出口端口而不是将其重新

路由(框107)。在一个实施例中,在转发数据分组前更新数据流的时戳(框113)。因此,负载分担过程能够表征为它通过只重新路由其中不需要另外的延迟以确保有序分组到达的那些数据流,寻求最小化重新路由的影响的机会性。

[0030] 如果超过阈值,则能够将数据分组和数据流指派到另一链路而不考虑失序到达。过程随后通过识别在LAG上数据分组通过其能够到达相同目的地的更不拥塞的出口端口而继续(框111)。更不拥塞的出口端口能够通过任何方法选择。在一个实施例中,通过比较与出口端口相关联的每个队列的量化的拥塞,并且选择最不拥塞的出口端口,能够选择更不拥塞的出口端口。在其它实施例中,队列拥塞或类似度量的加权平均值指示长期负载、负载趋势或轨迹。能够记录用于数据分组的时戳并且将其与用于数据流的新选择的出口端口相关联(框115)。与时戳记录平行,在其之后或在其之前,更新用于数据流的流表绑定以反映用于数据流的新选择的出口端口(框117)。随后,能够为数据流的随后数据分组利用或引用此数据。随后,数据流将保持与此出口端口和链路相关联,直至它变成拥塞,在此情况下,能够重新指派数据流。

[0031] 随后,将数据分组转发到已选择的更不拥塞的出口端口(框119)。在用于出口端口的队列中能够将数据分组排队,其中,为服务类和出口端口或目的地的每个组合建立单独的队列。这能够是线路卡内的队列,其中,数据分组等待通过网络元件的交换结构转发到适当的出口线路卡。

[0032] 图2是实现负载平衡和链路聚合群组的网络元件的一个实施例的图。网络元件包括管理数据分组跨网络的处理和转发以便路由到网络元件操作所处的网络中的目的地的线路卡203,205集。网络元件能够服务于与其它计算装置或网络元件的任何数量的链路或物理连接。每个线路卡203、205能够包括服务于这些链路的多个物理或虚拟端口。线路卡203、205能够用于处理入口和出口业务。网络元件能够支持任何数量的线路卡203、205。线路卡203、205能够通过交换结构223互连。

[0033] 交换结构223是管理线路卡之间相互通信的交换装置或交换卡集。在其它实施例中,利用总线或类似通信介质。交换结构223允许任何数量的线路卡相互进行通信,以便每个线路卡能够将数据分组转发到任何其它线路卡,从而允许数据分组到达其目的地。

[0034] 每个线路卡包括L2/L3处理器207、结构接入处理器(FAP) 221和流表217。L2/L3处理器207接收来自端口227的数据分组,或者将数据分组发送到端口227。端口处理通过链路与对应网络元件或计算装置的第1层数据通信。除其它L2和L3处理要求外,L2/L3处理器207管理用于进入数据分组的出口端口的识别。此L2/L3处理器207能够包括拥塞监视模块209和负载分担模块211。

[0035] 负载分担模块211使用散列或类似负载分布过程确定出口端口。出口端口的选择能够是在逐数据流的基础的数据流上。负载分担模块211能够应用到路径的任何等价集或任何LAG。负载分担模块211能够包括执行此初始出口端口选择过程的基础选择模块213。负载分担模块211也能够包括在指派的出口端口或链路拥塞时将数据流重新指派到其它出口端口和链路的重新分布模块215。负载分担模块211能够保持与每个出口端口相关联的拥塞的数据库或数据结构,或者能够与单独的拥塞监视模块209组合工作以保持当前拥塞信息。拥塞信息的收集能够相对于数据流负载分担异步进行。

[0036] 重新分布模块215利用拥塞信息确定出口端口是否具有拥塞,从而能够选择更不

拥塞的出口端口以服务于数据流。重新分布模块如上所述运行以通过将拥塞的出口端口上的数据流重新指派到其它更不拥塞的出口点，识别拥塞并且重新平衡负载。重新分布模块215通过保持用于转发到用于每个数据流的出口端口的最后数据分组的时戳，管理此类数据流重新指派的定时。这允许重新分布模块215识别何时能够迁移数据流而不考虑失序分组到达，这是因为数据流的前一分组在足够长之前已发送，使得最后数据分组将已到达目的地，或者将在备选出口端口上发送的数据分组之前到达。这些时戳能够存储在离散数据结构中，在流表中或者类似地存储。

[0037] 流表217是保持在数据流与出口端口或链路之间的绑定。流表217由L2/L3处理器207用于跟踪哪些数据流绑定到每个出口端口或链路。流表217也能够包括其它数据流有关信息，如时戳。

[0038] FAP 219包括保持要跨交换结构223转发到其它线路卡205和出口端口225的数据分组的队列或虚拟队列集。FAP 219能够保持用于数据流的每个出口端口、线路卡或类似组织的单独队列221。数据流也能够具有服务类或服务质量。FAP 219能够在每出口点的基础上保持用于每个服务质量或服务类的单独队列221。因此，每个出口端口225能够具有与其相关联的多个队列或虚拟队列。除管理用于出口端口的队列外，FAP也能够管理与交换结构223的接口，允许线路卡与其它线路卡交换数据分组。FAP 219保持的队列221能够具有任何数量、大小或组织。FAP 219也能够与拥塞监视模块209组合工作，以在FAP 219管理的每个队列221上提供拥塞报告。FAP 219能够通过发送拥塞通知消息或量化的拥塞通知消息，支持拥塞监视模块209。在一个实施例中，FAP 219将用于队列221的拥塞信息提供到线路卡203内的L2/L3模块207。在其它实施例中，FAP 219也能够将拥塞通知信息提供到其它线路卡，并且接收来自其它线路卡的拥塞信息。

[0039] 图3是用于为链路聚合群组实现负载平衡的过程的另一实施例的流程图。图形显示一特定示例实施例，总过程能够表征为涉及两个步骤。步骤一，检测到在聚合接口的构成链路上负载条件的不平衡。步骤二，将拥塞的链路上的数据流重新映射到聚合接口的负载相对更低的成员链路。在处理这些步骤前，描述支持步骤的拥塞监视。

[0040] 在上述系统上下文中，如果数据流分布跨LAG的成员链路不均一，则队列和相关联出口端口(例如，VOQ/虚拟端口)能够遇到拥塞。在一个实施例中，在IEEE标准802.1Qau中定义的QCN机制能够通过将QCN拥塞点(CP)附连到该出口端口监视出口端口来识别变得拥塞的端口。备选，通过定期的队列长度(例如，VOQ长度)的简单轮询，能够检测到拥塞。由此获得的队列长度能够通过EWMA滤波器传递以消除瞬间噪声和理解稳定趋势。大的队列长度是持久拥塞的指示。

[0041] 在此实施例中，QCN方法用于检测拥塞。QCN具有名为CP过程和反应点(RP)过程的两个拥塞监视组件。CP过程能够在出口端口(例如，VOQ/虚拟端口)上运行以检测拥塞。在检测到拥塞时，CP过程从进入流对帧采样，生成并且发送拥塞通知消息(CNM)到其来源或反应点(RP)。CNM消息包含传达拥塞的位置的拥塞点ID (CPID)和拥塞的量度Fb。Fb和CPID与负载分担和重新平衡过程相关。在此上下文中，CPID对应于队列(例如，VOQ)、出口端口或链路，并且Fb传达在其上拥塞的程度。因此，通过启用QCN CP，能够建立基于LAG的链路的拥塞状况的信息库。此信息能够用于将数据流动态重新映射到更不拥塞的成员链路，并且由此实现更佳的吞吐量和网络性能。

[0042] 平衡态阈值*Qeq*定义在拥塞条件下队列的操作点。换而言之，*Qeq*是目标级别，队列长度在正常拥塞条件下应在其左右振荡。CP计算拥塞度量*Fb*，并且通过取决于拥塞的严重性的概率，从进入流选择帧。如果*Fb*为负值，则它将反馈消息发送到来源。*Qlen*是给定队列的长度的测量。Fb = - (*Qoff* + *w***Qdelta*)，其中，*Qoff* = *Qlen* - *Qeq*且*Qdelta* = *Qlen*-*Qold*

[0043] *Fb*捕捉队列大小超过和速率超过的组合。因此，在*Fb*为负值时，队列预订过多，并且指示即将发生的拥塞。Fb的值越负，则拥塞的程度就越大。反馈消息包含Fb的量化的值、拥塞点id (CPID)，并且采样的帧封装在其内。

[0044] 实现所示示例过程要求将几个约束考虑在内。如果最初绑定到第一链路的数据流在数据流仍在活动状态时转移到另一链路，则存在分组重新排序的风险。为防止此情况发生，在流表中保持了时戳(也能够视为“聚合”流表，这是因为条目可对应于不止一个流)。此时戳记录数据流中最近分组的到达时间。将数据流绑定到不同出口端口的判定取决于此时戳。时间阈值定义用于创建新绑定，并且较小。计算在两个路径之间的差分延迟，带有一定的宽松以创建再次串行化分组的合并点。LAG是L2连接时，合并点能够是正好下一L2下一跳。如果自最后分组的到达时间起经过了足够的时间，则能够假设不存在输送中的数据流的分组，并且基于拥塞状态，能够将它安全地转移到不同出口端口。应注意的是，多个流能够映射到流表的一个条目。时戳将记录用于映射到条目的任何流的最近到达时间。如果经过了足够的时间，则将映射到条目的所有流转移到更不拥塞的链路。

[0045] 依赖FAP上QCN处理(或类似拥塞监视)的负载分担和重新平衡过程将发送带有*Fb*值的QCN CNM消息。*Fb*值被量化，使得其值提供由于拥塞从出口点的转移而在拥塞点拥塞的量度。基于*Fb*值，负载分担和重新平衡过程在LAG中布置不同端口，其方式使得负载最小的端口将绑定到新数据流，过程以此方式做出反应以逐渐校正不平衡。

[0046] 如上所提及的一样，负载分担和重新平衡过程依赖端口绑定操作。这些端口绑定操作依赖流表(即，聚合流表)。为便于说明，假设有如下所示NK个深度Agregate_Flow_Table[NK]，

[0047]

	时戳	绑定的端口
1		
2		
3		
.		
.		
.		
.		
.		
.		
NK		

表1

[0048] 在每个数据分组到达时，从分组报头提取关键字，并且对它进行散列处理以使索引进入上面所示的表格。这是聚合的流表，这是因为许多流能够映射到一个流桶(bucket)。负载重新平衡条件能够包括第一条件，第一条件是如果数据流绑定到出口端口，并且经过的时间不超过时间阈值，则通过现有绑定继续。第二个条件能够是如果经过的时间超过阈值，并且绑定的出口端口拥塞，或不是最小负载，则能够创建新绑定。

[0049] 这些条件尝试保护已经有负载的端口，而不是将它们进一步推入拥塞重叠。这是负载校正的继续过程，并且能够适用于命中高的数据流及命中低的数据流。作为可配置特征，如果出口端口极度拥塞，则能够以在短期间内一定的分组重新排序的代价，更改绑定。然而，这优于丢失数据分组或者在已经严重拥塞的点增大拥塞。

[0050] 假定存在上述上下文，返回到图3将负载分担和重新平衡过程示为流程图。过程响应在网络元件的线路卡的入口端口接收在数据分组的流中的数据分组而开始(框301)。随后，能够检查网络元件的输出接口是否配置为链路聚合群组(框303)。如果输出接口未配置为LAG，则过程不应用，并且数据分组能够正常转发到出口端口(框323)。

[0051] 如果输出接口为LAG，则过程通过执行初始负载分担过程以识别LAG的出口端口而继续(框305)。此初始负载分担过程使用诸如经散列处理以选择出口端口的报头信息等数据分组信息，从LAG识别出口端口。随后，使用通过拥塞监视过程建立的拥塞监视数据库，检查识别的出口端口是否拥塞(框307)。如果出口端口不拥塞，则时戳记录在流表或类似数据结构中(框313)。随后，将数据分组转发到识别的出口端口(框309)。

[0052] 随后，在当前时间与数据流的前一数据分组的时戳之间的差别超过定义的阈值(框311)。如果不超过阈值，则在流表或类似数据结构中更新时戳(框313)。随后，将数据分组转发到识别的出口端口(框309)。

[0053] 如果识别的出口端口拥塞，则使用诸如像fb等量化的值或队列长度等拥塞监视数

据，选择用于数据流的新出口端口(框315)。随后，在流表中用于新出口端口的条目中记录用于数据分组的时戳(框317)。也更新流表以将新出口端口绑定到数据流(框319)。随后，将数据分组转发到LAG的新出口端口(框321)。

[0054] 图4是分布式链路聚合群组的一个示例实施例的图。在一个实施例中，从下一跳或连接到LAG的每个链路的类似情况路由器接收拥塞监视数据。因此，分布式实现能够提供更稳固的拥塞信息。在此实施例中，QCN过程能够用于LAG负载拥塞数据收集。此实施例提供基于非机箱的系统。

[0055] 此实施例能够扩展到带有用于如图4所示负载分担情形的连线的离散元件的网络。路由器R1 401具有到路由器R2、R3和R4 403A-403C的拆分LAG接口411。从拆分LAG接口411外出的业务流跨成员链路分担负载，并且采用不同网络路径以到达目的地。考虑R1 401与其它路由器R2、R3和R4 403A-C建立了拥塞通知过程的情况。在一个示例实施例中，R1 401将在每个成员链路上的出口业务标记有不同的拥塞通知标记(CNTAG) 409，并且R2、R3和R4 403A-C向下游托管拥塞点。就拥塞而言，R2、R3和R4 403A-C生成的CNM 407能够以信号指示R1有关每个路径的拥塞状态。R1能够相应地更新其业务拥塞数据库，并且修改流绑定以跨LAG重新平衡负载。

[0056] 在其它备选实施例中，CP过程也能够在出口端口队列而不是入口线路卡的队列上运行(例如，VOQ)。从队列长度轮询中，并且通过使用指数加权移动平均(EWMA)引擎，也能够感应拥塞。

[0057] 要理解的是，上述描述旨在是说明性而不是限制性的。在阅读和理解上述描述后，本领域的技术人员将明白许多其它实施例。因此，本发明的范围应参照所附权利要求以及此类权利要求被授权的等同的完全范围来确定。

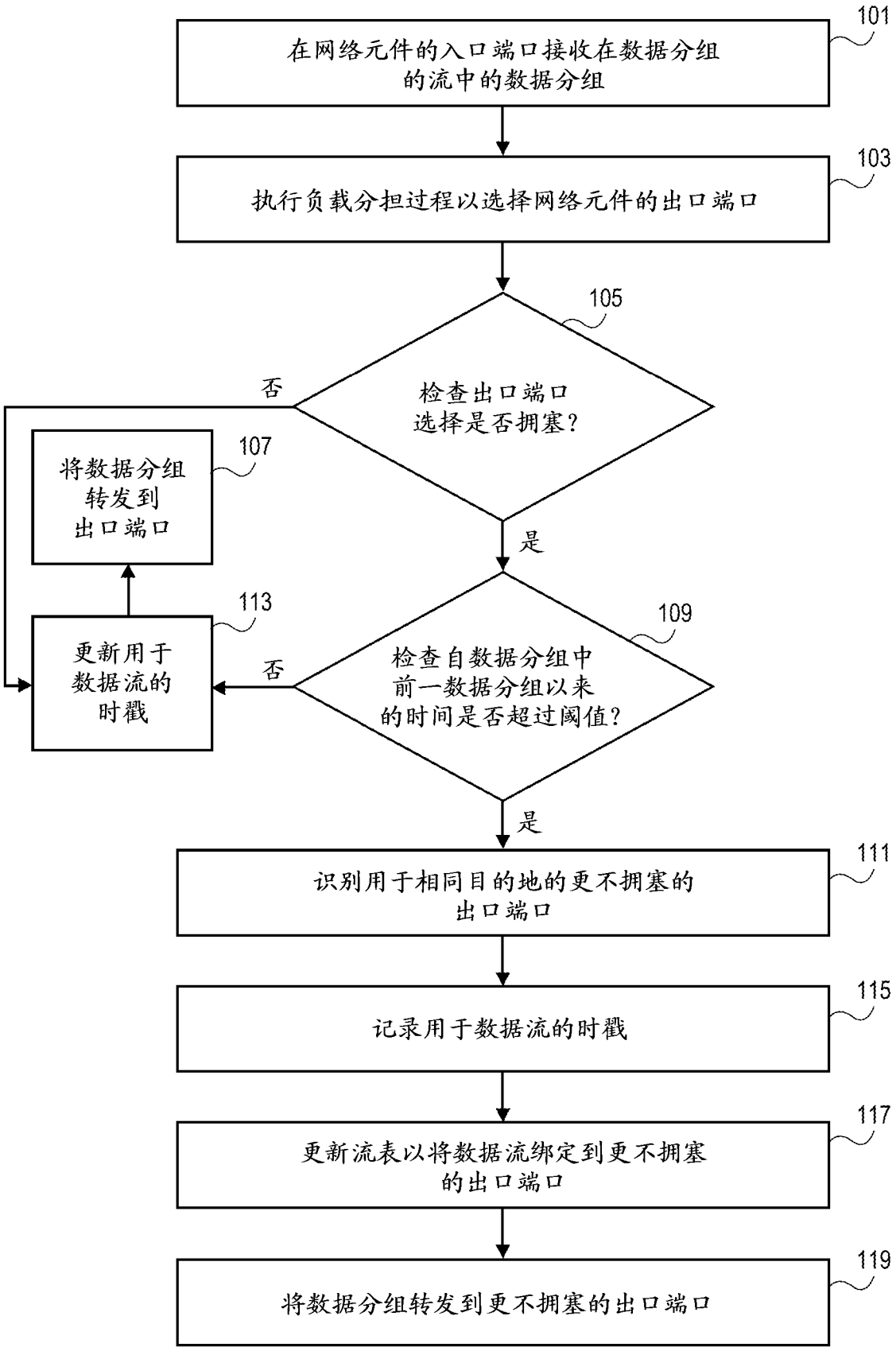

101
在网络元件的入口端口接收在数据分组的流中的数据分组
103
执行负载分担过程以选择网络元件的出口端口
105
检查出口端口选择是否拥塞?
否
是
107
将数据分组转发到出口端口
109
检查自数据分组中前一数据分组以来的时间是否超过阈值?
否
是
113
更新用于数据流的时戳
111
识别用于相同目的地的更不拥塞的出口端口
115
记录用于数据流的时戳
117
更新流表以将数据流绑定到更不拥塞的出口端口
119
将数据分组转发到更不拥塞的出口端口

图 1

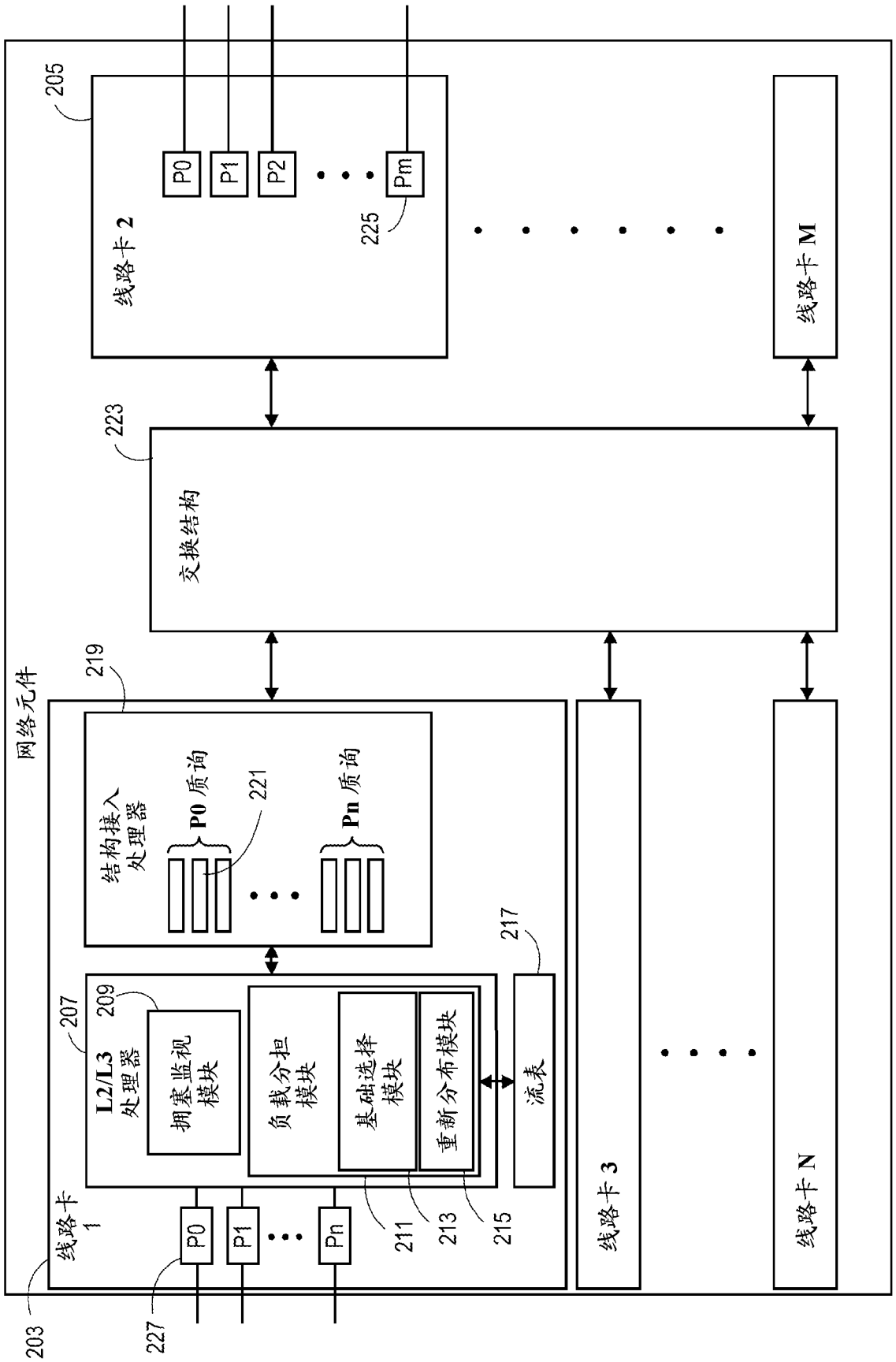

网络元件
线路卡 1
线路卡 2
线路卡 3
线路卡 M
线路卡 N
交换结构
结构接入处理器
P0 质询
Pn 质询
L2/L3 处理器
拥塞监视模块
负载分担模块
基础选择模块
重新分布模块
流表
P0
P1
P2
Pm
Pn
203
205
207
209
211
213
215
217
219
221
223
225
227

图 2

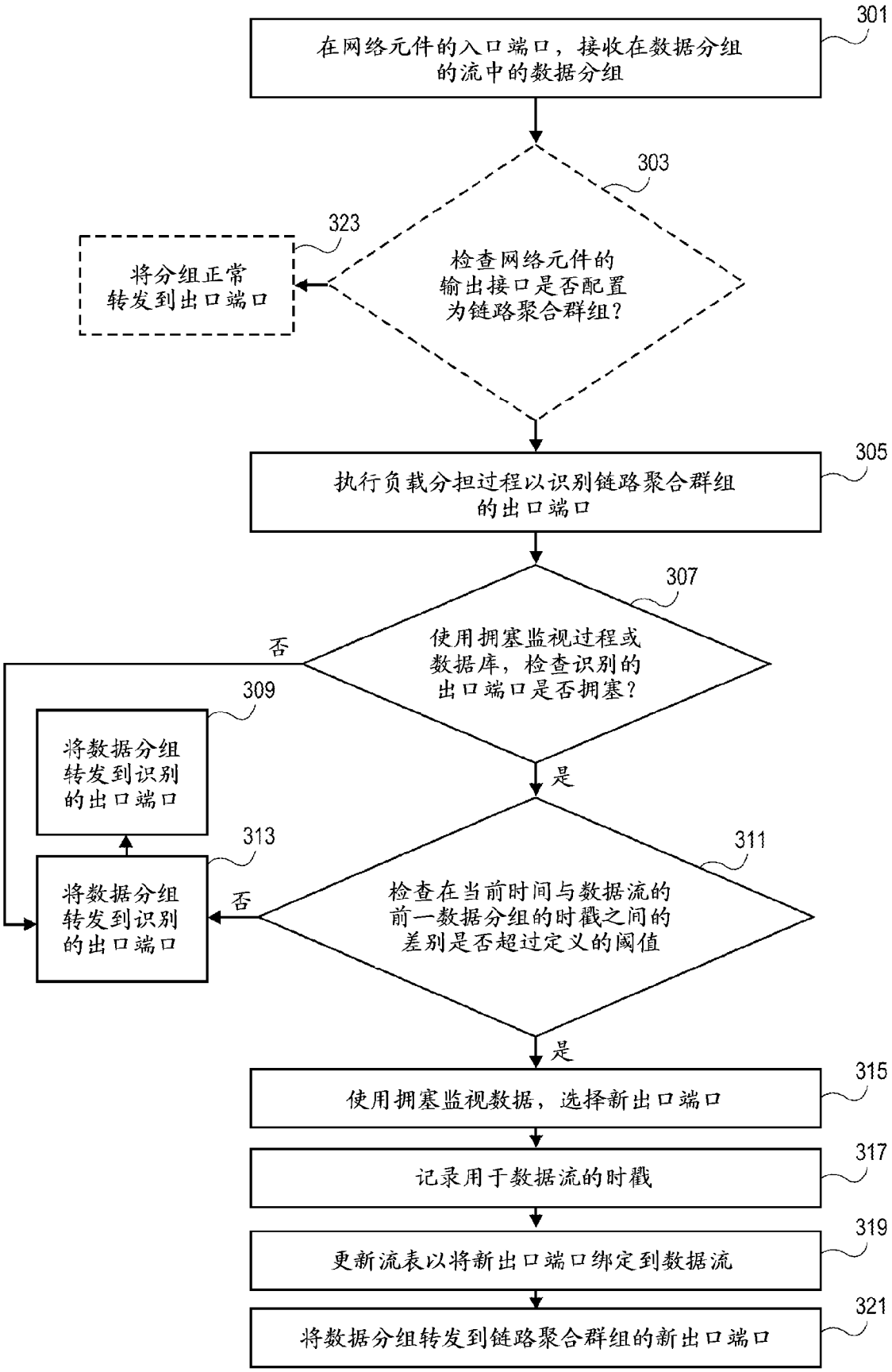

301
在网络元件的入口端口，接收在数据分组的流中的数据分组
303
检查网络元件的输出接口是否配置为链路聚合群组？
323
将分组正常转发到出口端口
305
执行负载分担过程以识别链路聚合群组的出口端口
307
使用拥塞监视过程或数据库，检查识别的出口端口是否拥塞？
否
是
309
将数据分组转发到识别的出口端口
311
检查在当前时间与数据流的前一数据分组的时戳之间的差别是否超过定义的阈值
否
是
313
将数据分组转发到识别的出口端口
315
使用拥塞监视数据，选择新出口端口
317
记录用于数据流的时戳
319
更新流表以将新出口端口绑定到数据流
321
将数据分组转发到链路聚合群组的新出口端口

图 3

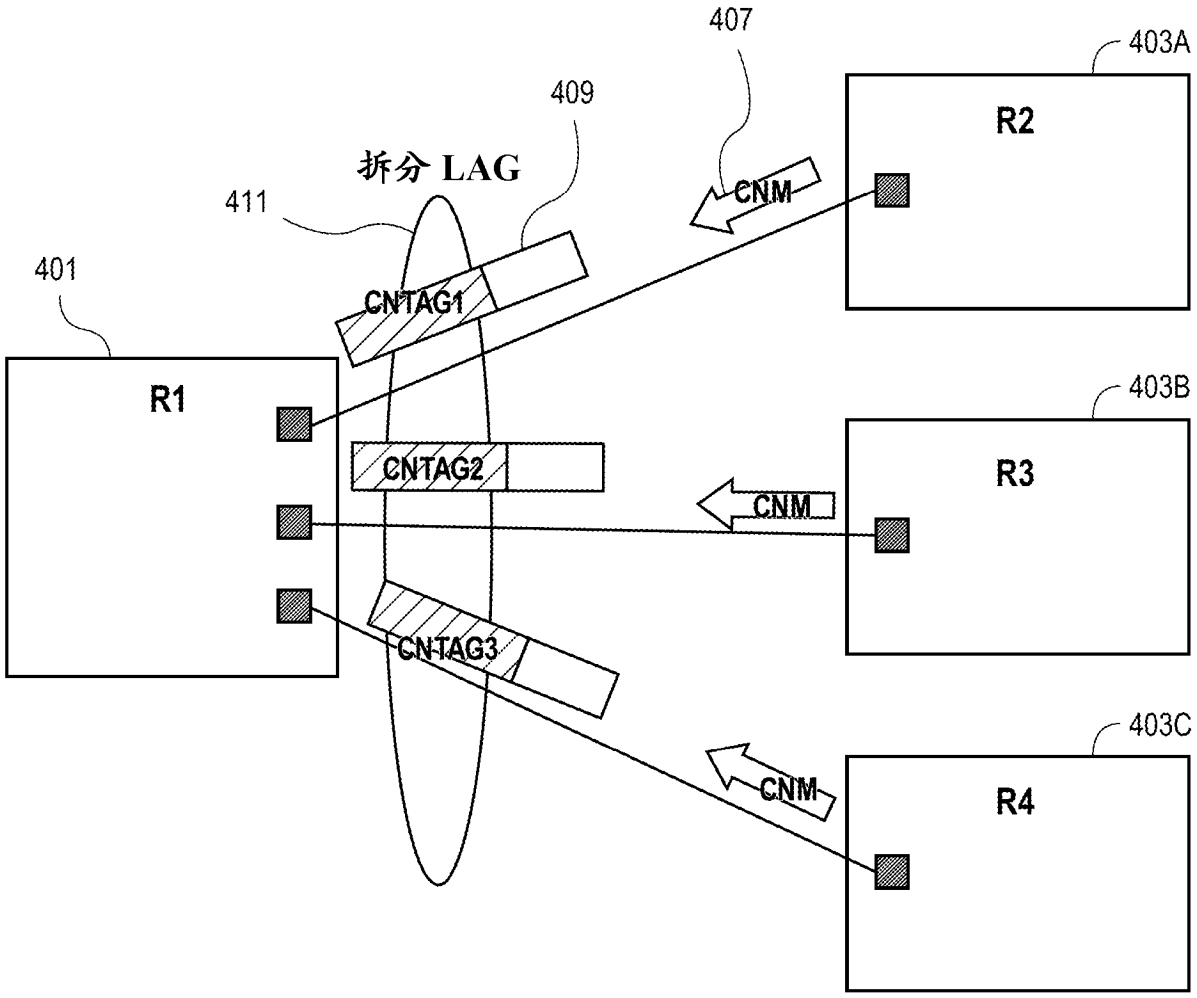

407
403A
409
R2
拆分 LAG
411
CNM
401
CNTAG1
R1
403B
CNTAG2
R3
CNM
CNTAG3
403C
CNM
R4

图 4