

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04L 12/56 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510120409.8

[43] 公开日 2006 年 5 月 24 日

[11] 公开号 CN 1777145A

[22] 申请日 2005.11.10

[21] 申请号 200510120409.8

[30] 优先权

[32] 2004.11.16 [33] KR [31] 10-2004-0093741

[71] 申请人 三星电子株式会社

地址 韩国京畿道

[72] 发明人 俞兑浚

[74] 专利代理机构 中科专利商标代理有限公司
代理人 戎志敏

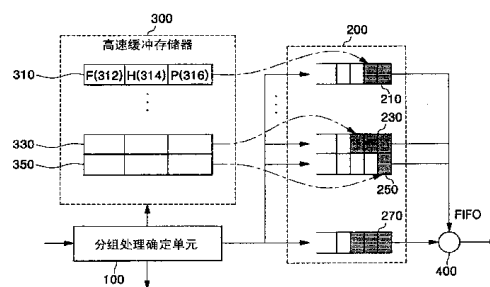
权利要求书 4 页 说明书 12 页 附图 7 页

[54] 发明名称

路由器及使用其管理分组队列的方法

[57] 摘要

提供了一种用于对分组传输执行队列管理的路由器，其包括：第一存储单元，用于存储并输出所输入的分组，其中所述分组请求从源设备发送到目的地设备；第二存储单元，用于存储关于所述第一存储单元中所存储的分组的信息；和分组处理确定单元，用于基于所述第一存储单元中是否存在可用存储容量，来确定是否将输入分组存储在所述第一存储单元中，并且将关于分组的信息更新到所述第二存储单元中。



1、 一种用于对分组传输执行队列管理的路由器，包括：

5 第一存储单元，用于存储并输出所输入的分组，其中所述分组请求从源设备发送到目的地设备；

 第二存储单元，用于存储关于所述第一存储单元中所存储的分组的
 信息；和

 分组处理确定单元，用于基于所述第一存储单元中是否存在可用
10 存储容量，来确定是否将输入分组存储在所述第一存储单元中，并且
 基于确定结果将关于分组的信息更新到所述第二存储单元中。

 2、 根据权利要求1所述的路由器，其中所述第二存储单元中存储的所述信息包括：

 流 ID (F)，这是关于请求发送分组的源设备的信息；

15 命中计数 (H)，指示相同源设备请求发送的次数；以及

 p_pos_queue (P)，指示关于所述第一存储单元中存储的分组位于何处的信息。

 3、 根据权利要求2所述的路由器，其中，当在所述第一存储单元中没有存储空间，并且所述命中计数的总和小于设置阈值时，所述
20 分组处理确定单元丢弃输入分组，并且根据最近最少使用 (LRU) 算法更新所丢弃的分组的流 ID (F)。

 4、 根据权利要求2所述的路由器，其中，当在所述第一存储单元中存在存储空间来存储分组时，所述分组处理确定单元将输入分组存储在所述第一存储单元中，并且根据最近最少使用 (LRU) 算法更新
25 包含所述分组的流 ID (F) 的信息。

 5、 根据权利要求2所述的路由器，其中，当在所述第一存储单元中没有存储空间，并且所述命中计数的总和大于设置阈值时，所述分组处理确定单元从所述第一存储单元中检测在所述第二存储单元中具有最大命中计数 (H) 的流 ID (F) 的分组，将输入分组存储到所述
30 第一存储单元的空余空间中，并且将关于所存储的分组的的信息更新到

所述第二存储单元中。

6、 根据权利要求 5 所述的路由器，其中，当输入分组的流 ID (F) 存储在所述第二存储单元中，并且所述命中计数 (H) 为最大值时，所述分组处理确定单元根据最近最少使用 (LRU) 算法更新所述流 ID (F)。

7、 根据权利要求 6 所述的路由器，其中，当输入分组的流 ID (F) 存储在所述第二存储单元中，并且所述命中计数 (H) 不是最大值时，所述分组处理确定单元将所述第二存储单元中存储的命中计数 (H) 的最大值减“1”，并且将输入分组的命中计数 (H) 加“1”，同时根据最近最少使用 (LRU) 算法更新所述流 ID (F)。

8、 根据权利要求 7 所述的路由器，其中，当输入分组的流 ID 没有存储在所述第二存储单元中，并且存在空间来更新时，所述分组处理确定单元将与输入分组相对应的条目存储到所述第二存储单元中。

9、 根据权利要求 8 所述的路由器，其中，当输入分组的流 ID (F) 没有存储在所述第二存储单元中，并且没有空间来更新时，所述分组处理确定单元根据最近最少使用 (LRU) 算法删除最近最少使用的条目，并且在删除后的空间处更新输入分组的相应条目。

10、 一种使用路由器来管理分组传输的队列的方法，包括：

从源设备接收请求发送的分组；

确定第一存储单元中是否存在可用存储空间来存储所述分组；

当所述第一存储单元中没有存储空间时，基于所述源设备请求发送的次数与设置阈值的比较结果，确定是存储还是丢弃所述分组；以及

将关于根据确定结果所处理的分组的信息更新到第二存储单元中，所述第二存储单元存储关于分组的信息。

11、 根据权利要求 10 所述的方法，其中所述第二存储单元中存储的关于分组的信息包括：

流 ID (F)，这是关于请求发送分组的源设备的信息；

命中计数 (H)，指示相同源设备请求发送的次数；以及

p_pos_queue (P), 指示关于所述第一存储单元中存储的分组位于何处的信息。

12、 根据权利要求 10 所述的方法, 其中更新信息的步骤包括:
当在所述第一存储单元中没有存储空间, 并且所述命中计数的总和小于设置阈值时, 丢弃输入分组, 并且根据最近最少使用 (LRU) 算法更新所丢弃的分组的流 ID (F)。

13、 根据权利要求 11 所述的方法, 其中更新信息的步骤包括:
当在所述第一存储单元中存在存储空间来存储分组时, 将输入分组存储在所述第一存储单元中, 并且根据最近最少使用 (LRU) 算法更新包含所述分组的流 ID (F) 的信息。

14、 根据权利要求 11 所述的方法, 其中更新信息的步骤包括:
当在所述第一存储单元中没有存储空间, 并且所述命中计数的总和大于设置阈值时, 从所述第一存储单元中检测在所述第二存储单元中具有最大命中计数 (H) 的流 ID (F) 的分组; 将输入分组存储到所述第一存储单元的空余空间中, 并且将关于所存储的分组的更新到所述第二存储单元中。

15、 根据权利要求 14 所述的方法, 其中更新信息的步骤还包括:
当输入分组的流 ID (F) 存储在所述第二存储单元中, 并且所述命中计数 (H) 为最大值时, 根据最近最少使用 (LRU) 算法更新所述流 ID (F)。

16、 根据权利要求 15 所述的方法, 其中更新信息的步骤还包括:
当输入分组的流 ID (F) 存储在所述第二存储单元中, 并且所述命中计数 (H) 不是最大值时, 将所述第二存储单元中存储的命中计数 (H) 的最大值减 “1”, 并且将输入分组的命中计数 (H) 加 “1”, 同时根据最近最少使用 (LRU) 算法更新所述流 ID (F)。

17、 根据权利要求 16 所述的方法, 其中更新信息的步骤还包括:
当输入分组的流 ID (F) 没有存储在所述第二存储单元中, 并且存在空间来更新时, 将与输入分组相对应的条目存储到所述第二存储单元中。

18、 根据权利要求 17 所述的方法, 其中更新信息的步骤还包括:

当输入分组的流 ID (F) 没有存储在所述第二存储单元中，并且没有空间来更新时，根据最近最少使用 (LRU) 算法删除最近最少使用的条目，并且在删除后的空间处更新输入分组的相应条目。

路由器及使用其管理分组队列的方法

5

技术领域

本发明涉及一种路由器及使用其管理分组队列的方法，更具体地，涉及一种能够在维持公平缓冲区占用同时控制分组传输的路由器及使用其管理分组队列的方法。

10

背景技术

一般来说，在因特网上流动的业务量的大小和传输速率不同。使用队列管理以及调度方案来最小化当业务量在因特网上流动时可能发生的诸如拥塞之类的问题。

15 一个这样的问题是业务量的不公平。术语“不公平”是指这样一种现象：少数特定业务量占用路由器中大部分的缓冲区容量，而不考虑公平性。

图 1 图示了因特网上的业务量的不公平现象的示例。

如图 1 所示，在路由器 30 与路由器 40 之间的网络的链路带宽为
20 10Mbps。

在图 1 中，当分别使用传输控制协议（TCP）和用户数据报协议（UDP）的应用 A 10 和应用 B 20 处于对路由器 30 的缓冲区的竞争条件时，使用 UDP 的应用 B 20 最终占用路由器 30 中大部分缓冲区容量。

也就是说，当使用 UDP 的应用 B 20 向 B 的接收器（sink）60 发送大于 10Mbps 的分组时，随着时间流逝，其将占用大部分链路带宽。因此，即使使用 TCP 的应用 A 10 希望向 A 的接收器 50 发送分组，也没有剩余带宽用来进行传输，这导致了传输的不公平。

已经提出了队列管理以及调度方案来解决前述问题。

一个这样的示例是基于去尾队列管理的 FIFO（先进先出）或先来
30 先服务（FCFS）调度。其优点在于，简单分组转发使分组处理的开销

最小化，并且易于实现。然而，基于去尾的 FIFO 调度仅支持尽力而为的服务。换言之，该方案具有这样的结构缺陷：不保证服务质量，并且路由器的缓冲区可能被生成许多业务量的某些流占用。

已经提出了多种队列管理算法和分组调度机制来改进基于去尾的 FIFO 调度的缺点。这当中，IntServ（集成服务）模型除了尽力而为的服务之外还加入了新的服务类。为了加入这些服务，路由器应该获取必要的资源来保证流的服务质量。所获取的资源包括带宽、存储器等。诸如 RSVP（资源预留协议）之类的协议用来获取资源。

然而，IntServ 模型的问题在于，没有足够的可扩展性，并且需要许多资源，因为其预先为服务获取资源，并且保留所有流的信息。

引入了 DiffServ（区分服务）模型来解决 IntServ 模型的问题。在 DiffServ 模型中，将各种流分类为数个服务类，并且在中介路由器中对各个服务类进行处理。DiffServ 模型不需要对所有路由器进行流状态管理和通告。DiffServ 模型在分组头部的特定位中指定所需的服务类。这种方案基于所需的 QoS（服务质量）来对所有业务量进行分类，并且相应地聚集相关业务量，以解决调度问题。

另外，提出了使用局部状态方案的 RED（LRU-RED）（最近最少使用-随机早期检测）和 FQ（LRU-FQ）（最近最少使用-公平排队）作为 IntServ 模型与 DiffServ 模型之间的中间模型。术语“局部状态”是指，与 IntServ 模型不同，路由器不保留所有流的信息，而是仅使用有限的存储器来保留特定流的信息。这些流的信息的存储器管理遵循 LRU 算法。

由于 LRU 算法的特性，在相对长的时间中频繁发送分组的流，该流中的信息存储在存储器中的概率较高。这里，其信息存储在存储器中的流被定义为比其信息没有存储在存储器中的流发送相对较多分组的流，即，破坏公平性的流。

路由器分析输入分组。当分组对应于存储器中所包含的流时，这些分组经历指定调整。这里，在 LRU-RED 的情形中，应用这样的 RED 算法：将存储器中所存储的流丢弃的概率高。LRU-RED 使用两个存储器。对存储在存储器中的流，分组被存储在一个队列中，而对没有存

储在存储器中的流，分组存储在一个队列中，然后应用同等调度，来抑制不公平性。

然而，IntServ 模型需要每一个路由器存储流的状态信息。这又在路由器中需要大容量的存储装置，并且大大影响大多数流中的处理速度。另外，对控制相关功能（例如，接入管理和批准/许可）的处理带来了高开销。最后，中途的所有路由器应该支持 IntServ 模型，这样降低了可扩展性。

同时，DiffServ 模型中所使用的业务量聚集模型具有很差的预测性。因此，对于 DiffServ 模型而言，非常难以保证一定水平的服务。因此，DiffServ 模型基于每个聚集的规则来提供相对服务，而不是保证一定水平的服务。换言之，某些聚集接收数据要好于或差于其他聚集。

LRU-RED 同样具有 RED 队列管理的缺点。其降低了缓冲区的整体使用率。另外，难以建立可靠的调整策略，因为调整是基于概率的。LRU-FQ 在存在许多流交换少数分组的网络中引起分组记录的问题以及公平性问题。

发明内容

做出本发明是为了解决上述问题。本发明的目的是通过使用局部状态提供能够对缓冲区保持公平而不会使特定流占用路由器中所有缓冲区的路由器及使用其管理队列的方法。

本发明的另一目的是通过使用局部状态方案提供能够减轻对使用存储空间的需求的路由器及使用其管理队列的方法。

根据本发明的一个方面，提供了一种用于对分组传输执行队列管理的路由器，包括：第一存储单元，用于存储并输出所输入的分组，其中所述分组请求从源设备发送到目的地设备；第二存储单元，用于存储关于所述第一存储单元中所存储的分组的信息；和分组处理确定单元，用于基于所述第一存储单元中是否存在可用存储容量，来确定是否将输入分组存储在所述第一存储单元中，并且将关于分组的信息更新到所述第二存储单元中。

优选地，所述第二存储单元可以包括：流 ID (F)，这是关于请求发送分组的源设备的信息；命中计数 (H)，指示相同源设备请求发送的次数；以及 p_pos_queue (P)，指示关于所述第一存储单元中存储的分组位于何处的信息。

- 5 当在所述第一存储单元中没有存储空间，并且所述命中计数的总和小于设置阈值时，所述分组处理确定单元可用丢弃输入分组，并且根据最近最少使用 (LRU) 算法更新所丢弃的分组的流 ID。

- 当在所述第一存储单元中存在存储空间来存储分组时，所述分组处理确定单元可以将输入分组存储在所述第一存储单元中，并且根据
10 LRU 算法更新包含所述分组的流 ID 的信息。

- 当在所述第一存储单元中没有存储空间，并且所述命中计数的总和大于设置阈值时，所述分组处理确定单元可以从所述第一存储单元中检测在所述第二存储单元中具有最大命中计数的流 ID 的分组，将输入分组存储到所述第一存储单元的空余空间中，并且将关于所存储的
15 分组的信息更新到所述第二存储单元中。

 当输入分组的流 ID 存储在所述第二存储单元中，并且所述命中计数为最大值时，所述分组处理确定单元可以根据 LRU 算法更新所述流 ID。

- 当输入分组的流 ID 存储在所述第二存储单元中，并且所述命中
20 计数不是最大值时，所述分组处理确定单元可以将所述第二存储单元中存储的命中计数的最大值减“1”，并且将输入分组的命中计数加“1”，同时根据 LRU 算法更新所述流 ID。

- 当输入分组的流 ID 没有存储在所述第二存储单元中，并且存在空间来更新时，所述分组处理确定单元可以将与输入分组相对应的条
25 目存储到所述第二存储单元中。

 当输入分组的流 ID 没有存储在所述第二存储单元中，并且没有空间来更新时，所述分组处理确定单元可以根据 LRU 算法删除最近最少使用的条目，并且在删除后的空间处更新输入分组的相应条目。

- 根据本发明的另一方面，提供了一种使用路由器来管理分组传输
30 的队列的方法，该方法包括如下步骤：从源设备接收请求发送的分组；

确定第一存储单元中是否存在可用存储空间来存储所述分组；当所述第一存储单元中没有存储空间时，基于所述源设备发送分组的重复次数与设置阈值的比较结果，确定是存储还是丢弃所述分组；以及将关于根据确定结果所处理的分组的信息更新到第二存储单元中，所述第
5 二存储单元存储关于分组的信息。

当在所述第一存储单元中没有存储空间，并且所述命中计数的总和小于设置阈值时，更新信息的步骤包括：丢弃输入分组，并且根据最近最少使用（LRU）算法更新所丢弃的分组的流 ID。

当在所述第一存储单元中存在存储空间来存储分组时，更新信息的
10 的步骤包括：将输入分组存储在所述第一存储单元中，并且根据 LRU 算法更新包含所述分组的流 ID 的信息。

当在所述第一存储单元中没有存储空间，并且所述命中计数的总和大于设置阈值时，更新信息的步骤包括：从所述第一存储单元中检测在所述第二存储单元中具有最大命中计数的流 ID 的分组；将输入分
15 组存储到所述第一存储单元的空余空间中，并且将关于所存储的分组的信息更新到所述第二存储单元中。

根据本发明，使用局部状态方案，减轻了对存储空间的需求，并且基于这种方案，可以将缓冲区占用的不公平性控制在通过阈值定义的水平上，因此，可以更公平地控制分组的缓冲区占用。另外，当缓
20 冲区中没有存储空间时，对相关分组执行队列管理，由此最大化缓冲区利用。另外，如果必要的话，通过基于阈值来调节对缓冲区的调整，易于改变缓冲区管理策略。

附图说明

25 结合附图，参考下面的详细描述，对本发明的更彻底的认识及本发明所具有的许多优点将变得更好理解，因此将变得非常清楚，在附图中相似的标号标识相同或相似的部件，其中：

图 1 图示了因特网上业务量的不公平现象；

图 2 示出了根据本发明实施例的具有关于从相应源设备输入的流
30 的信息的高速缓冲存储器的结构；

图 3 示出了根据本发明实施例的用于最近最少使用-最长队列丢弃 (LRU-LQD) 队列管理的路由器;

图 4 是示出了使用根据本发明的路由器的示例性队列管理方法的流程图;

5 图 5 是示出了在从高速缓冲存储器中检测具有最大命中计数的流 ID 的过程之后处理输入分组的流程图;

图 6 是具体示出将分组相关信息以及缓冲区中存储的分组的流 ID 更新到高速缓冲存储器中的过程的流程图; 以及

图 7 示出了使用根据本发明实施例的路由器的 LRU-LQD 队列管理
10 方法的伪代码的示例。

具体实施方式

下面, 将参考附图更详细地描述本发明实施例的配置和操作。在附图中, 相似的标号表示相似的元件。另外, 当对已知的相关功能或
15 配置的详细描述将会使本发明的主旨变得含糊时, 将省略这样的描述。

根据本发明, 提出并公开了最近最少使用-最长队列丢弃 (LRU-QD) 队列管理方法, 该方法通过使用局部状态 (仅使用特定的有限信息, 而不是所有流的信息), 能够维持路由器中缓冲区利用的公平性。

20 图 2 示出了根据本发明实施例的具有关于从相应源设备输入的流的信息的高速缓冲存储器的结构。

如图 2 所示, 高速缓冲存储器 100 包括: 流 ID (F) 120, 这是关于请求发送相关分组的源设备的信息; 命中计数(hit count) (H) 140, 指示相同源设备请求发送该分组的次数; 以及 p_pos_queue (P) 160, 指示关于该队列中相关分组位于何处的信息。这里, 术语“命中”表示在分组输入到路由器时, 发送该输入分组的源设备于缓冲 100 中设置的流 ID (F) 120 匹配。换言之, 命中计数 (H) 140 是指从相同源将请求发送的分组输入的次数。
25

图 3 示出了根据本发明优选实施例的用于最近最少使用-最长队列丢弃 (LRU-LQD) 队列管理的路由器。
30

如图 3 所示，路由器包括分组处理确定单元 101、队列 200、高速缓冲存储器 300、以及先进先出（FIFO）单元 400。

分组处理确定单元 101 基于队列 200 的缓冲区是否可用，确定是存储还是丢弃输入分组。

- 5 队列 200 存储并输出其缓冲区中从分组处理确定单元 101 接收到的分组。这里，在将输入分组存储到缓冲区中之后，队列 200 基于是否可以输出分组，顺序将所存储的分组输出到 FIFO 单元 400。

高速缓冲存储器 300 也在分组处理确定单元 101 的控制下存储关于队列中所存储的分组的信息 310、330 和 350。高速缓冲存储器 300
10 以分组为基础，分别存储与队列中相应缓冲区 210、230 和 250 中所存储的每个分组相对应的信息 310、330 和 350。例如，在分组处理确定单元 101 的控制下，高速缓冲存储器 300 存储队列 200 的第一缓冲区 210 中所存储的分组的信息 310，包括流 ID (F) 312、命中计数 (H) 314、以及分组存储位置 (P) 316。

- 15 队列 200 在逻辑上具有由多个缓冲区构成的单个缓冲单元。图 3 图示了作为缓冲区 210、230 和 250 的这些缓冲区中的一些，其中缓冲区 210、230 和 250 分别对应于高速缓冲存储器 300 的信息 310、330 和 350，并且对应于请求发送分组的各自的源设备。队列 200 还包括下面将要讨论的缓冲区 270 作为其缓冲单元的一部分。如上所述，缓
20 冲区被视为单个缓冲单元。

FIFO 单元 400 以先进先出的方式示出从队列 200 的各个缓冲区 210、230、250 和 270 接收到的分组。输出分组通过传输线分别发送到目的地设备。

- 同时，在确定是丢弃还是存储输入分组时，分组处理确定单元 101
25 基于队列 200 中是否存在可用存储空间，并且基于将高速缓冲存储器 300 中存储的各个分组的命中计数信息 H 的总和与设置阈值进行比较的结果，来做出确定。

换言之，在接收分组时，如果在路由器的队列 200 中的缓冲单元中没有可用存储空间，并且高速缓冲存储器的命中计数 H 的总和小于
30 阈值，则分组处理确定单元 101 丢弃输入分组，并根据最近最少使用

(LRU) 算法将丢弃分组的流 ID (F) 更新到高速缓冲存储器 300。这种情形意味着分组对缓冲区占用的不公平性不会超过用户定义的范围。

当路由器的队列 200 中的缓冲单元中没有可用存储空间，并且高速缓冲存储器的命中计数 H 的总和大于设置阈值时，分组处理确定单元 101 在队列 200 中的分组中检测流 ID (F) 与最大命中计数 (H) 相对应的分组。丢弃所检测到的分组，并且将当前输入分组存储在队列 200 的缓冲单元中。当高速缓冲存储器 300 中与当前输入分组相对应的命中计数 (H) 是最大命中计数 (H) 时，分组处理确定单元 101 不执行上述过程，相反，丢弃该流。

缓冲 300 中的相应变化包括下列三种情形：

第一种情形是输入分组的流 ID (F) 已经存储在高速缓冲存储器 300 中并且其命中计数 (H) 最大。在这种情形中，只执行这样的过程：根据 LRU 算法更新高速缓冲存储器 300 中关于分组的信息，例如，信息 310。

第二种情形是输入分组的流 ID 已经存储在缓冲 300 中但是其命中计数 (H) 不是最大。在这种情形中，高速缓冲存储器 300 中存储的另一分组的最大命中计数 (H) 减“1”，而与输入分组的流 ID 相对应的命中计数 (H) 加“1”。接着，根据 LRU 算法更新高速缓冲存储器 300。

第三中情形是输入分组的流 ID (F) 不在高速缓冲存储器 300 中。此时，分组处理确定单元 101 确定高速缓冲存储器 300 中是否有可用存储空间来存储输入分组及其信息。

当存在可用存储空间时，分组处理确定单元 101 在高速缓冲存储器 300 中存储相应条目。当高速缓冲存储器 300 中没有存储空间时，分组处理确定单元 101 根据 LRU 算法删除最近最少使用的条目，然后存储关于当前输入分组的信息。

图 4 是示出了使用根据本发明的路由器来管理队列的示例性方法的流程图。

首先，当分组处理确定单元 101 从源设备接收到分组时 (S110)，

其确定队列 200 的缓冲单元中是否有空余空间来存储分组 (S120)。如果确定存在存储接收分组的空间, 分组处理确定单元 101 将该分组存储到队列 200 的缓冲区 270 中 (S210)。分组处理确定单元 101 将分组相关信息以及缓冲区 270 中存储的该流的流 ID 更新到高速缓冲存储器 300 中 (S220)。

同时, 当在步骤 S120 中确定队列 200 中没有缓冲区空间来存储分组时, 分组处理确定单元 101 确定高速缓冲存储器 300 的命中计数 (H) 的总和是否大于阈值 (S130)。当确定命中计数的总和小于阈值时, 分组处理确定单元 101 丢弃输入分组 (S140), 并且将相关信息以及该分组的流 ID 更新到高速缓冲存储器 300 中 (S150)。

另一方面, 当在步骤 S130 中确定命中计数的总和大于阈值时, 分组处理确定单元 101 检测高速缓冲存储器 300 中命中计数最大的分组的流 ID (F) (S160)。

分组处理确定单元 101 然后从队列 200 的缓冲单元中检测并丢弃所存储的与所检测到的流 ID (F) 相对应的分组 (S170), 除非当前接收到的分组对应于已存储的具有最大命中计数 (H) 的分组 (这种情形见图 5 的过程)。

此时, 分组处理确定单元 101 将当前接收到的分组存储到丢弃后的队列 200 中腾出的缓冲区中 (S180)。另外, 分组处理确定单元 101 将包括该分组的流 ID (F) 在内的信息更新到高速缓冲存储器 300 中 (S190)。

图 5 是示出了作为图 4 的步骤 S160 的子进程的、处理输入分组的过程的流程图。

首先, 分组处理确定单元 101 确定当前接收到的分组是否对应于具有最大命中计数的流 ID (F) 的分组 (S310)。当确定当前输入分组不对应于具有最大命中计数 (H) 的流 ID (F) 时, 该进程前进到图 4 的步骤 S170。

在当前输入分组对应于具有最大命中计数 (H) 的流 ID (F) 时, 分组处理确定单元 101 丢弃当前输入分组 (S320)。然后, 分组处理确定单元 101 确定高速缓冲存储器中是否存在与当前丢弃分组相对应的

流 ID (F) (S330)。

当确定高速缓冲存储器中存在当前丢弃分组的流 ID (F) 时，分组处理确定单元 101 确定命中计数 (H) 是否是高速缓冲存储器 300 中存储的命中计数中最大的一个 (S340)。当确定与当前丢弃分组相对

5 应的相关流 ID (F) 的命中计数 (H) 不是最大时，分组处理确定单元 101 将最大命中计数值减“1” (S350)。即，另一分组的命中计数 (H) 被确定为最大命中计数，并且将该计数值减“1”。

另外，分组处理确定单元 101 将与当前丢弃分组相对应的相关流 ID (F) 的命中计数 (H) 加“1” (S360)。此时，分组处理确定单元

10 101 根据 LRU 算法更新流 ID (F) (S370)。

在 S340 中，当确定与当前丢弃分组相对应的相关流 ID 的命中计数 (H) 最大时，分组处理确定单元 101 根据 LRU 算法将包括流 ID (F) 在内的分组相关信息更新到高速缓冲存储器 300 中 (S380)。

另一方面，当在步骤 S330 中确定当前丢弃分组的流 ID 不在高速

15 缓冲存储器中时，分组处理确定单元 101 确定是否存在空间来将关于当前输入分组的信息更新到高速缓冲存储器 300 中 (S410)。当确定存在空间来更新分组信息时，分组处理确定单元 101 将与该分组相对应的条目存储到高速缓冲存储器 300 中 (S420)。

在 S410 中，当确定在高速缓冲存储器 300 中没有空间来更新分

20 组信息时，分组处理确定单元 101 根据 LRU 算法，从高速缓冲存储器 300 中删除最近最少使用的条目 (S430)。此时，分组处理确定单元 101 将与该分组相对应的条目存储到删除后的空间中 (S440)。

图 6 是具体示出图 4 的步骤 S220 的流程图。

首先，分组处理确定单元 101 确定输入分组的流 ID (F) 是否已

25 经在高速缓冲存储器 300 中 (S221)。如果确定流 ID (F) 在高速缓冲存储器 300 中，则分组处理确定单元 101 将高速缓冲存储器 300 中存储的相应命中计数 (H) 加“1” (S222)。此时，分组处理确定单元 101 根据 LRU 算法将该分组的流 ID (F) 更新到高速缓冲存储器 300 中 (S223)。

30 同时，当在 S221 中确定输入分组的流 ID (F) 不在高速缓冲存储

器 300 中时，分组处理确定单元 101 确定在高速缓冲存储器 300 中是否存在空间来更新或存储关于输入分组的信息（S224）。当确定在高速缓冲存储器 300 中存在空间来更新关于分组的信息时，分组处理确定单元 101 将与输入分组相对应的条目存储到高速缓冲存储器 300 中（S225）。

5 当在 S224 中确定在高速缓冲存储器 300 中没有空间来更新关于分组的信息时，分组处理确定单元 101 根据 LRU 算法从高速缓冲存储器 300 中删除最近最少使用的条目。此时，分组处理确定单元 101 将与输入分组相对应的条目存储到高速缓冲存储器 300 中现在空出的空间中（S227）。

10 在图 4 的步骤 S150 或 S190，或者图 6 的步骤 S223、S225 或 S227 之后，分组处理确定单元 101 确定是否输出队列 200 的缓冲单元中存储的分组之一用于传输（S228）。如果从队列 200 输出了分组，则分组处理确定单元 101 将高速缓冲存储器 300 中存储的相应分组的命中计数（H）减“1”（S229）。

15 图 7 示出了使用根据本发明实施例的路由器的 LRU-LQD 队列管理方法的伪代码的示例。

20 这里，“阈值”和“条目概率”是由管理者设置的因子。较高的阈值指示减少对高速缓冲存储器 300 中存储的流的调整。较高的条目概率指示减轻了可以在高速缓冲存储器 300 中存储流的条件。所示的少许代码包括在高速缓冲存储器 300 中存储或更新信息的进程以及管理队列的进程。

25 根据本发明，使用局部状态方案，减轻了对存储空间的需求，并且基于这种方案，可以将缓冲区占用的不公平性控制在通过阈值定义的水平上。因此，可以更公平地控制分组的缓冲区占用。

另外，当在缓冲单元中没有存储空间时，通过执行对相关分组的队列管理，可以最大化缓冲区利用。

此外，通过根据阈值来调节对缓冲区使用率的调整，易于按需改变缓冲区管理策略。

30 已经描述并图示了本发明的示例性实施例。然而，本发明并不局

限于此，本领域的技术人员应该认识到，在不脱离所附权利要求所包括的本发明的精神的前提下，可以做出许多修改。

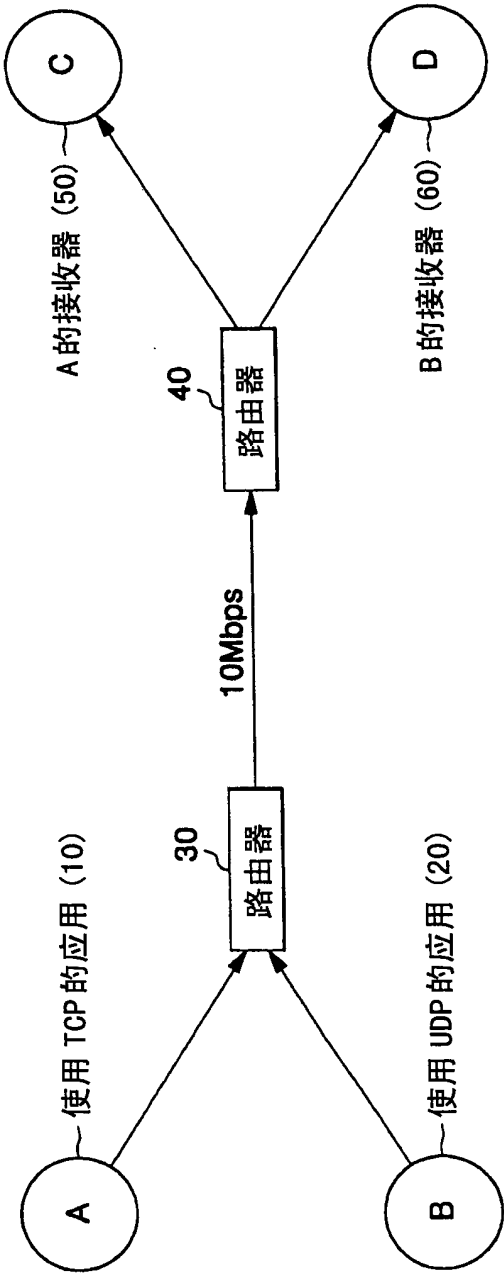


图 1

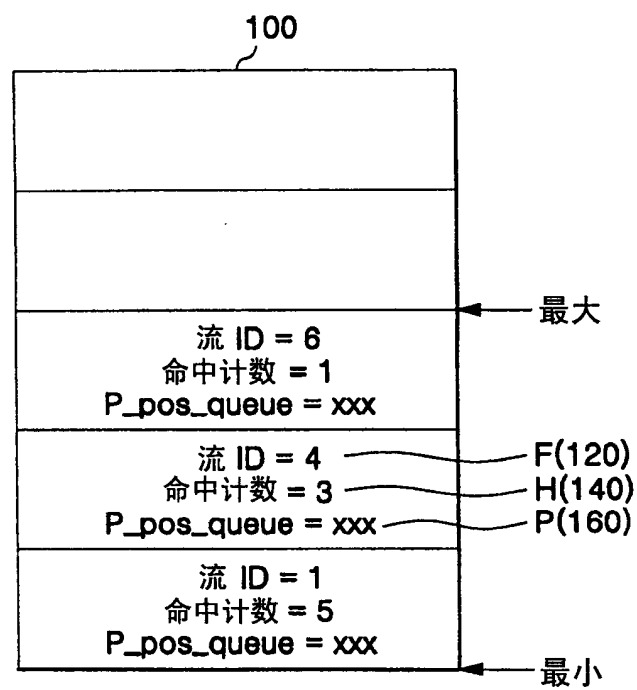


图 2

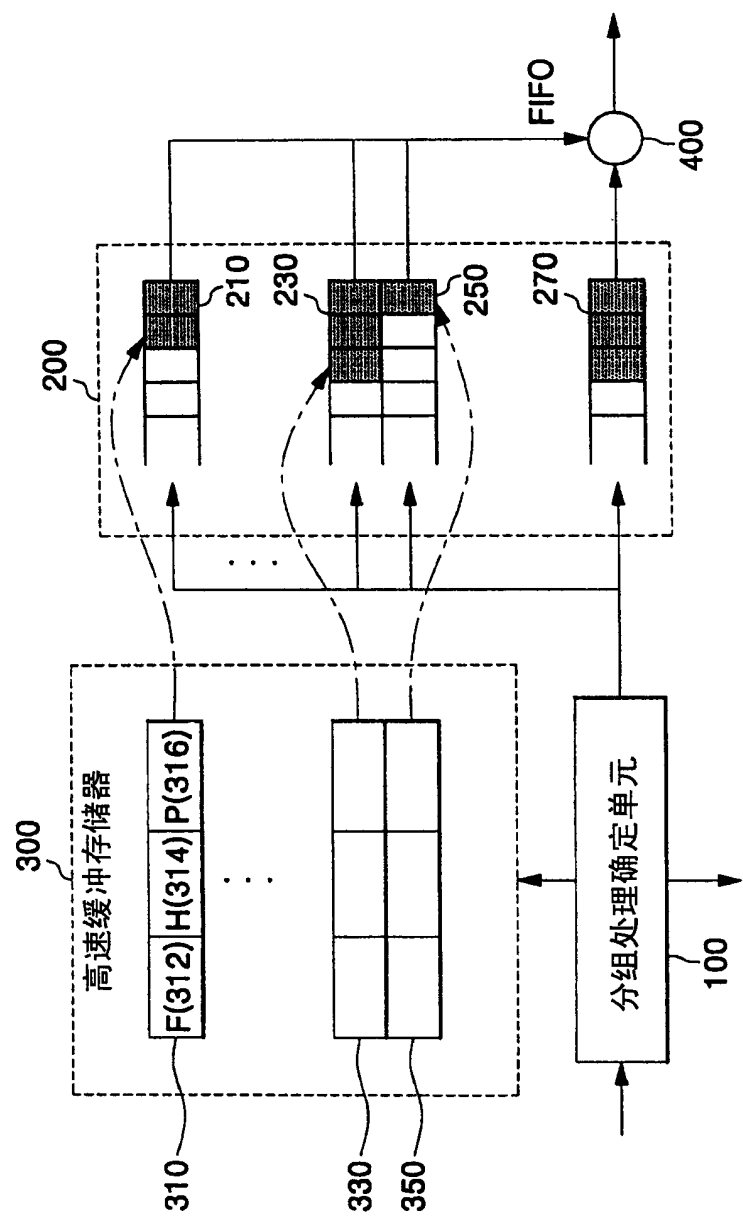


图 3

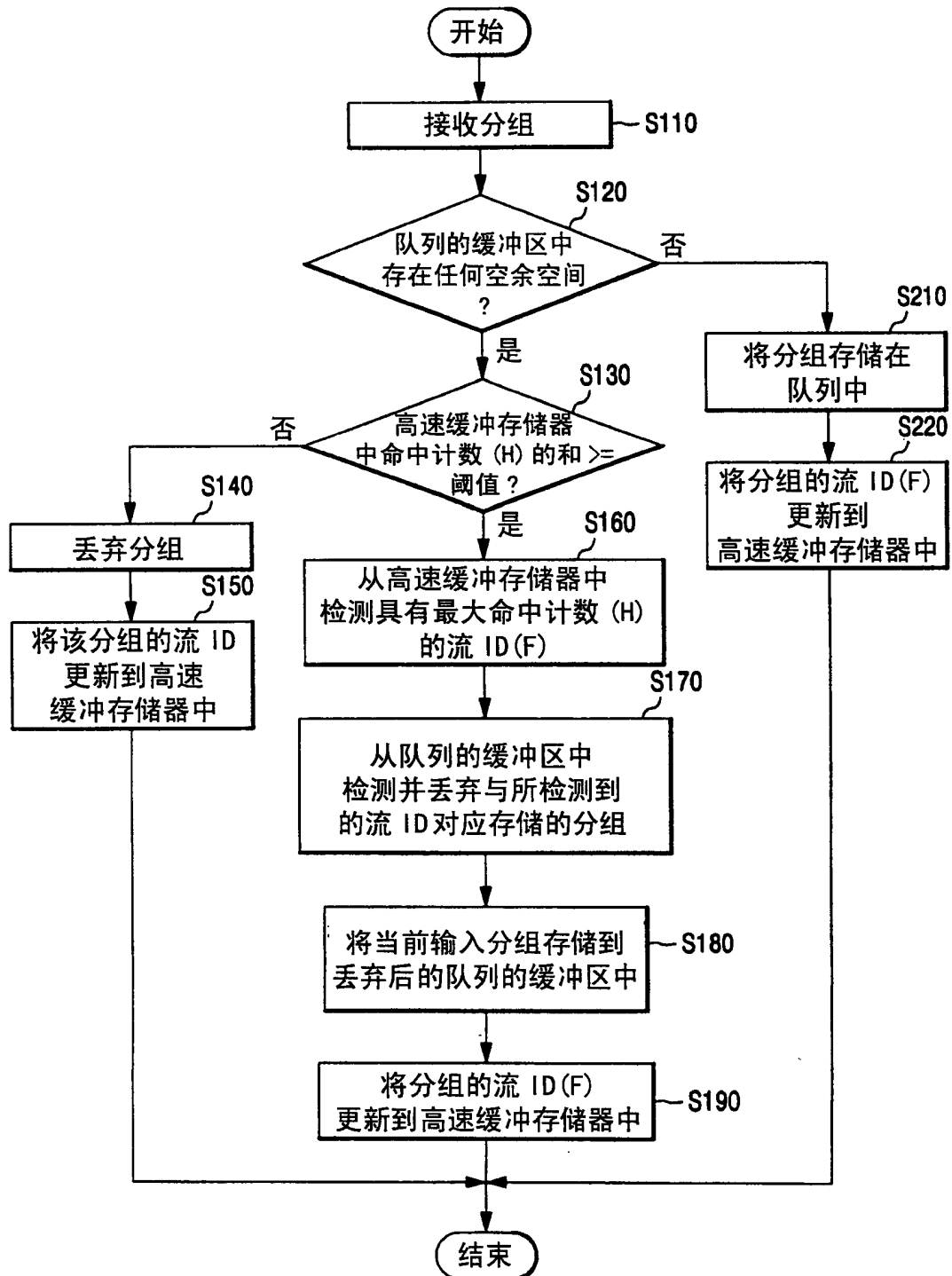


图 4

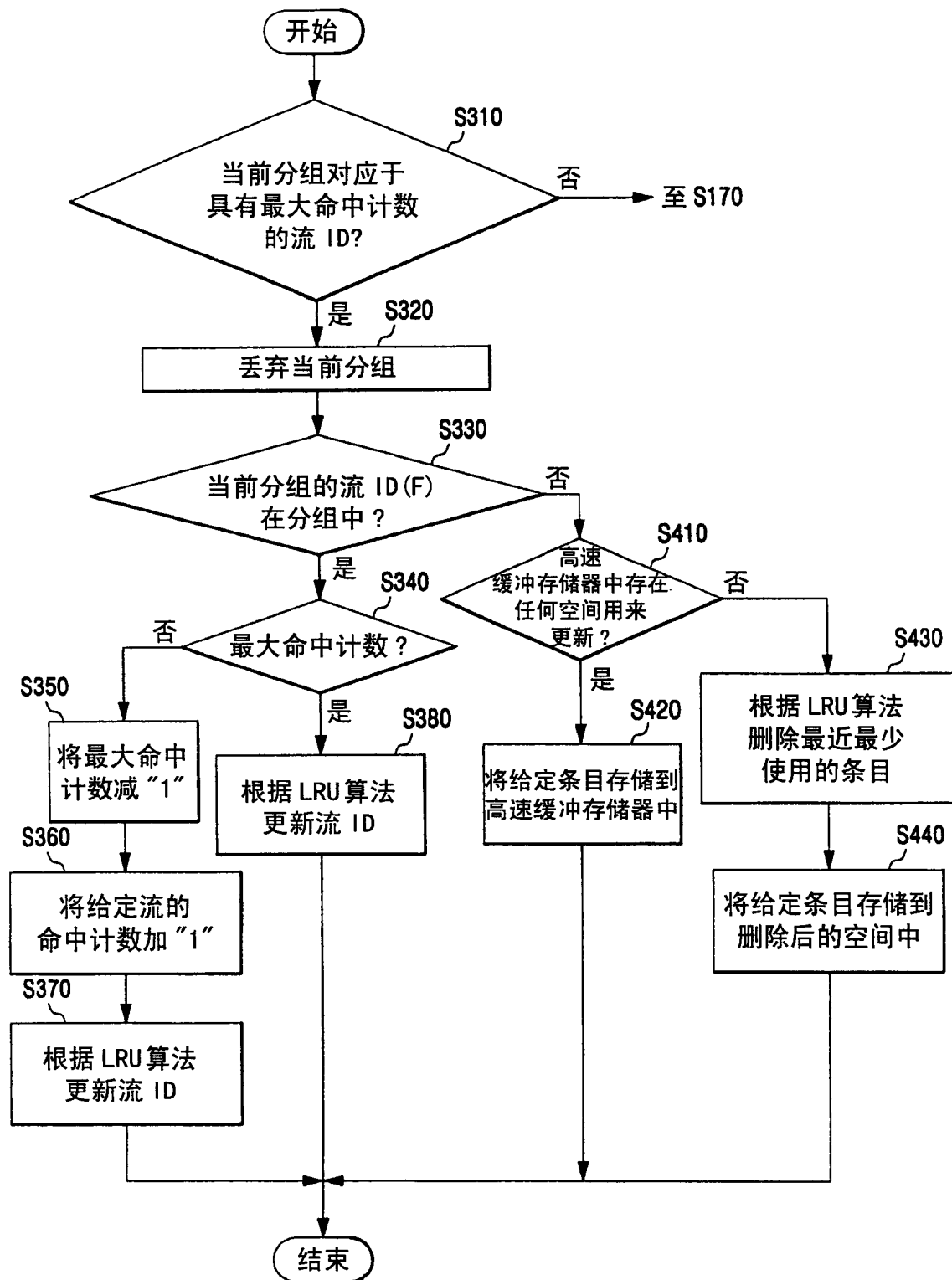


图 5

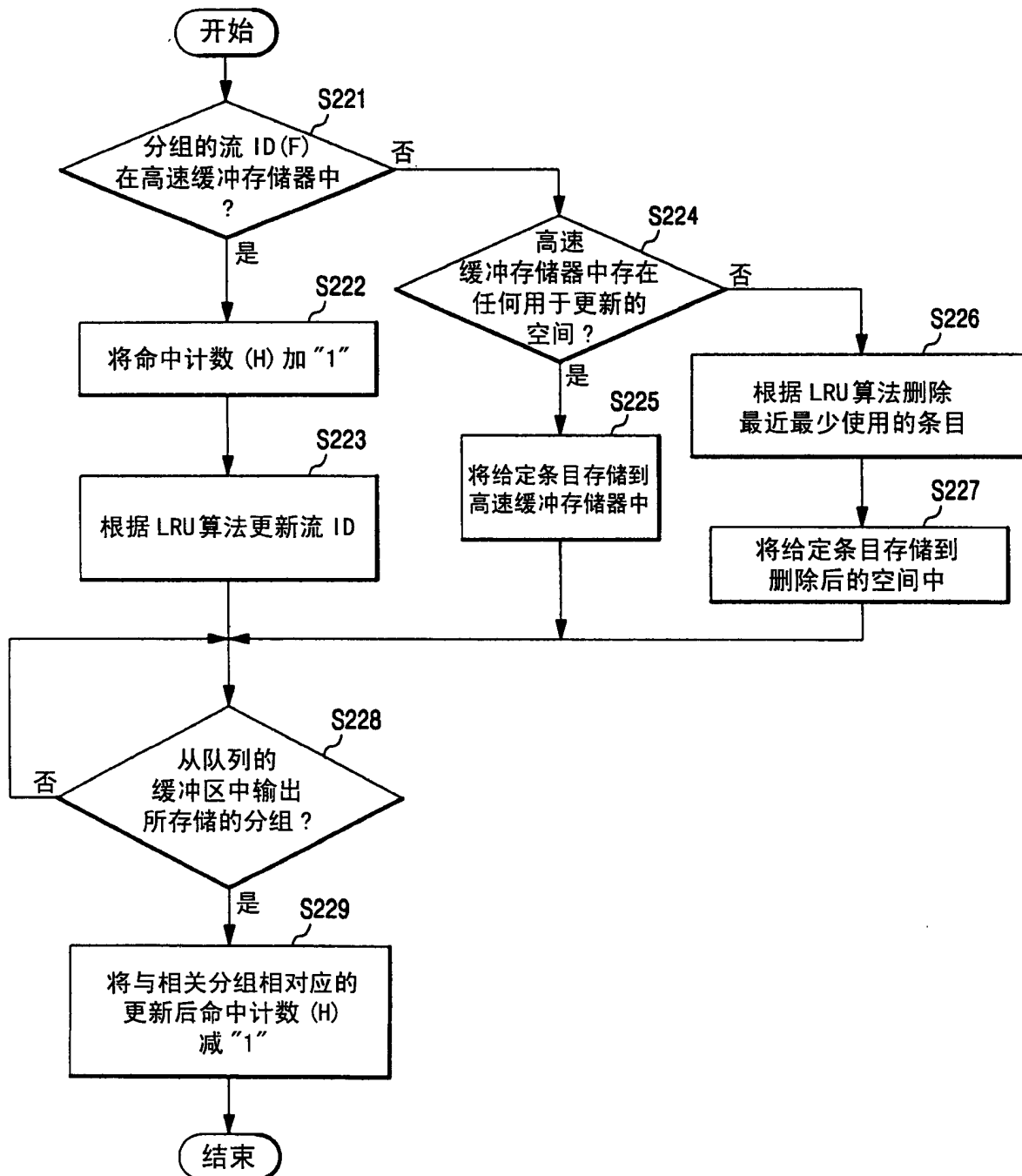


图 6

```

分组入队算法
分组 p 到达输入队列
if { 队列长度 >= 队列限制 ^ 每个高速缓存流的计数和 < 阈值 } {
    if { 发送分组 p 的流在高速缓冲存储器中 }
        { 利用 LRU 算法在高速缓冲存储器中更新该流的条目；丢弃分组 p；}
    else { 丢弃分组 p；}}
    else if { 队列长度 >= 队列限制 ^ 每个高速缓存流的计数和 >= 阈值 } {
        if { 发送分组 p 的流在高速缓冲存储器中 ^ 发送分组 p 的流具有最大计数 } {
            { 利用 LRU 算法在高速缓冲存储器中更新该流的条目；丢弃分组 p；}
        }
        else if { 发送分组 p 的流在高速缓冲存储器中 ^ 发送分组 p 的流不具有最大计数 } {
            { 选取来自队列中具有最大计数的流的分组，并且减小计数，并丢弃该分组；
              使分组 p 入队，并且增加计数，并且利用 LRU 算法在高速缓冲存储器中更新该流的条目；}
        }
    }
    else { 使分组 p 入队；
        if { 进入高速缓冲存储器测试 = YES } {
            { 在高速缓冲存储器中加入该流；}}
        }
    else { 使分组 p 入队；
        if { 发送分组 p 的流在高速缓冲存储器中 } {
            { 增加计数，并且利用 LRU 算法在高速缓冲存储器中更新该流的条目；}
        }
        else { if { 进入高速缓冲存储器测试 = YES } { 在高速缓冲存储器中加入该流；}}
        }
    }
}
进入高速缓冲存储器测试
分组 p 试图进入高速缓冲存储器
Pen: 进入高速缓冲存储器的固定概率
生成随机均匀数 n
if { n > Pen } { 返回 NO；}
else { 返回 YES；}
分组出队算法
if { 发送分组 p 的流在高速缓冲存储器中 } {
    if { 分组 p 没有入队，而发送分组 p 的流已经加入 } {
        { 使分组 p 出队；}
    }
    else { 使分组 p 出队；减小计数；}
}
else { 使分组 p 出队；}

```

图 7