



[12] 发明专利申请公开说明书

[21] 申请号 01810544.0

[43] 公开日 2003 年 7 月 23 日

[11] 公开号 CN 1432243A

[22] 申请日 2001.5.1 [21] 申请号 01810544.0

[30] 优先权

[32] 2000. 5. 1 [33] US [31] 60/201,010

[32] 2001. 4. 30 [33] US [31] 09/844,216

[86] 国际申请 PCT/US01/14045 2001. 5. 1

[87] 国际公布 WO01/84788 英 2001. 11. 8

[85] 进入国家阶段日期 2002. 12. 2

[71] 申请人 河流三角洲网络工程有限公司

地址 美国马萨诸塞州

[72] 发明人 乔恩·C·R·班纳特

[74] 专利代理机构 永新专利商标代理有限公司

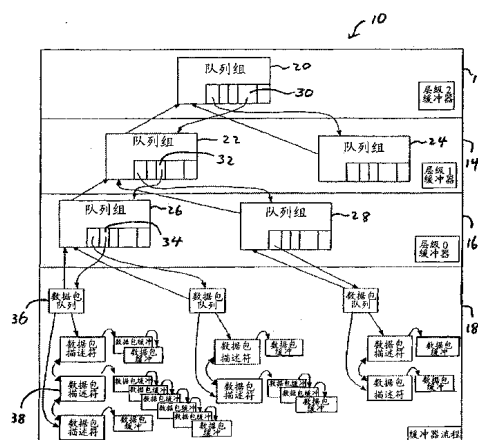
代理人 过晓东

权利要求书 10 页 说明书 23 页 附图 4 页

[54] 发明名称 数据包丢弃

[57] 摘要

一种用于分级地存储打包的数据并传送该数据包的系统，包括用于接收数据包的一个输入端，耦合到该输入端、用于存储数据包的存储器，耦合到该存储器、用于传送存储器中的数据包的输出端和耦合到存储器、用于控制该存储器结合一个分级结构成队列地存储数据包的控制单元，在所述分级结构中，该分级结构的第一层级包括一组队列组组件，其中该组中第一层级的至少一个队列组组件包括关联第二层级的一组队列组组件，所述第二层级不同于该分级结构的第一层级。



1. 一种用于分级地存储打包的数据并传送该打包的数据的系统，其包括：

用于接收数据包的输入端；

耦合到该输入端、用于存储数据包的存储器；

耦合到该存储器、用于传送存储器中的数据包的输出端；和

耦合到存储器、用于控制该存储器结合一个分级结构成队列地存储数据包的控制器，在所述分级结构中，该分级结构的第一层级包括一组队列组组件，其中该组中第一层级的至少一个队列组组件包括关联第二层级的一组队列组组件，所述第二层级不同于该分级结构的第一层级。

2. 如权利要求 1 所述的系统，其中每个队列组组件至少关联逻辑低队列和逻辑低队列组中的一个，并且，其中所述控制器用于控制存储器将数据包存储在分级结构的 N 层级中，其中 N 大于 2 且除了两个逻辑最低的层级以外，每个层级的至少一个队列组组件包括至少两个队列组组件关联一个逻辑低层级。
3. 如权利要求 2 所述的系统，其中所述的关联一个逻辑低层级的至少两个队列组组件关联一个分级结构中直接逻辑较低的层级。
4. 如权利要求 1 所述的系统，其中所述控制器用于控制存储器存储队列组组件的多个组，该组关联分级结构的第一层级。

5. 如权利要求 1 所述的系统，其中每个队列组组件至少关联逻辑低队列和逻辑低队列组中的一个，其中所述控制器用于确定第一层级中队列组组件的组中至少一个队列组组件的丢弃优先权，该优先权用于丢弃数据包的选择。
6. 如权利要求 5 所述的系统，其中所述控制器用于确定丢弃优先权，根据与队列有关的队列尺寸、源类型、源尺寸和服务请求中的至少一个。
7. 如权利要求 5 所述的系统，其中所述控制器用于确定一个丢弃值。
8. 如权利要求 7 所述的系统，其中所述丢弃值直接随着队列尺寸而变化。
9. 如权利要求 5 所述的系统，其中所述控制器用于确定第二层级的至少一个队列组组件的丢弃优先权，该优先权用于丢弃数据包的选择。
10. 如权利要求 9 所述的系统，其中所述控制器用于确定丢弃优先权，该优先权用于丢弃数据包的选择，用于第一层级的至少一个队列组组件和不同地第二层级的至少一个队列组组件。
11. 如权利要求 5 所述的系统，其中所述控制器用于丢弃队列中的至少一个数据包，该队列与具有一个第一层级的组中队列组组件的最高丢弃优先权的队列组组件相关。

12. 如权利要求 11 所述的系统，其中第一层级的队列组组件的组是第一组，且其中如果具有第一组的队列组组件的最高丢弃优先权的队列组组件包括一个第二层级的队列组组件的第二组，那么该控制器确定一个丢弃优先权用于第二队列组组件的组的每个队列组组件，并且丢弃至少一个队列的数据包，该队列与具有第二组中的最高丢弃优先权的第二组中的队列组组件相关。
13. 如权利要求 11 所述的系统，其中所述至少一个被控制器丢弃的数据包在队列的头、队列的尾和队列的随机位置的任何一处处理，所述队列与具有第一组队列组组件的最高丢弃优先权的队列组组件相关。
14. 如权利要求 11 所述的系统，其中所述控制器用于丢弃队列的多个数据包，该队列与具有最高丢弃优先权的队列组组件相关，无论该丢弃优先权在所有的数据包丢弃之前是否改变。
15. 如权利要求 7 所述的系统，其中所述控制器用于确定丢弃值，该丢弃值用于至少一个第一层级的组中的队列组组件，从不同的意义上说，而不是用于至少一个第一层级的组中的其他队列组组件。
16. 如权利要求 7 所述的系统，其中所述控制器用于确定丢弃值，该丢弃值用于分级结构的第一层级的队列组组件，该分级结构包括分级结构的第二层级的队列组组件的组，基于该主要组件的有效尺寸和队列的有效尺寸的至少一个，所述队列在分级结构的一个逻辑低层级而不是第二层级，所述层级与相关于该主要组件的第二层级的至少一个组件相关。

17. 如权利要求 5 所述的系统，其中所述控制器用于丢弃队列的至少一个数据包，该队列与具有大约是第一层级的组的队列组组件的最高丢弃优先权的队列组组件相关。
18. 如权利要求 5 所述的系统，其中所述控制器用于丢弃队列的至少一个数据包，该队列与具有丢弃优先权的队列组组件相关，该优先权在一个丢弃值的最高优先权的范围内，该丢弃值与第一层级的队列组组件的组中至少一个队列组组件相关。
19. 如权利要求 5 所述的系统，其中所述控制器用于丢弃队列的至少一个数据包，该队列与响应于存储器的队列组组件相关，该存储器尝试存储将超出该存储器数据范围的一些数据。
20. 如权利要求 5 所述的系统，其中所述控制器用于丢弃队列的至少一个数据包，该队列与响应于存储器的队列组组件相关，该存储器尝试存储将超出队列组的数据数量范围的一些数据。
21. 如权利要求 20 所述的系统，其中所述的至少一个数据包从不同于下述队列的队列中丢弃，所述的队列与存储器尝试超出其数据范围的组相关。
22. 如权利要求 20 所述的系统，其中所述的至少一个数据包被丢弃，无论存储器是否尝试存储一些数据，该数据将超出存储器数据量的界限。

23. 如权利要求 5 所述的系统，其中所述的控制器用于丢弃与队列组组件相关的队列的至少一个数据包，根据下面至少一个：响应于至少尝试存储将超出数据量界限（相关于该系统）的一些数据的存储器，和提前随机地丢弃。
24. 如权利要求 23 所述的系统，其中所述的数据量界限是队列尺寸界限。
25. 一种用于分级地存储打包的数据并传送该打包数据的系统，其包括：
用于接收数据包的输入端；
耦合到存储器的输出端，用于传送存储器的数据包；和
耦合到该输入端、输出端的控制装置，用于以队列方式存储输入端接收到的数据包，该队列与分级结构相关，该分级结构中每个层级都包括队列组组件的至少一个组，且该分级结构中不同于分级结构中最低层级的每个层级上的至少一个队列组组件包括一个队列组组件的组，所述组与分级结构的逻辑低层级相关，每个队列组组件与逻辑低队列和一个逻辑低队列组中的至少一个相关，所述控制装置还用于传送存储的数据包给输出端。
26. 如权利要求 25 所述的系统，其中所述的逻辑低层级是在分级结构中直接逻辑较低。
27. 如权利要求 25 所述的系统，其中所述控制装置还用于确定队列组组件的有效长度。

28. 如权利要求 27 所述的系统，其中所述控制装置用于根据与队列相关的队列尺寸、源类型、源尺寸和服务请求中的至少一个来确定有效长度。
29. 如权利要求 28 所述的系统，其中有效长度的值直接地随着队列尺寸而变化。
30. 如权利要求 29 所述的系统，其中所述控制装置还用于丢弃所选队列的至少一个数据包，所述队列与队列组组件相关，该组件具有研究用于丢弃的分级结构的组中最长的有效长度。
31. 如权利要求 30 所述的系统，其中研究的分级结构的组与分级结构的较高层级的队列组组件相关，该组件具有与之相关的组的最长有效长度。
32. 如权利要求 31 所述的系统，其中所述控制装置用于丢弃所选队列的多个数据包，无论与所选队列相关的该队列组组件在所有的数据包被丢弃之前，是否中断具有它的组中最长的有效长度。
33. 如权利要求 27 所述的系统，其中所述控制装置用于确定至少两个不同的分级机构组件的有效长度。
34. 如权利要求 27 所述的系统，其中所述控制装置用于丢弃队列的至少一个数据包，该队列与具有一有效长度的队列组组件相关，该有效长度在有效长度的最大长度的范围内，所述范围与研究用于丢弃的分级结构组相关。

35. 一种用于分级地存储打包的数据并传送该打包数据的系统，其包括：

用于接收数据包的输入端；

耦合到该输入端的存储器，用于存储数据包；

耦合到存储器的输出端，用于传送存储器的数据包；和

耦合到存储器的控制器，用于控制存储器以队列的方式存储数据包，所述队列与分级结构相关，在该分级结构中，第一层级包括一个队列组组件的组，其中第一层级的该组中至少一个队列组组件包括一个队列组组件的组，该组与分级结构的第二层级相关，所述第二层级不同于第一层级，每个队列组组件与逻辑低队列和逻辑低队列组中的至少一个相关；

其中所述控制器用于控制存储器在分级结构的 N 层级上存储数据包，其中的 N 大于 2 并且不同于两个逻辑最低层级的每个层级的至少一个队列组组件包括至少一个队列组和一个其他的与逻辑低层级相关的队列组组件；和

其中，所述控制器用于确定一个丢弃优先权，该优先权是第一层级队列组组件的组中每个队列组组件的优先权，用于丢弃数据包的选择，该丢弃优先权基于丢弃值，所述丢弃值直接随着与队列组组件有关的队列的队列尺寸而变化。

36. 如权利要求 35 所述的系统，其中所述控制器用于根据与队列有关的源类型、源尺寸和服务请求中的至少一个确定丢弃优先权。

37. 如权利要求 35 所述的系统，其中所述控制器用于确定用于不同的分级结构队列组组件的丢弃数据包的选择的丢弃优先权。
38. 如权利要求 35 所述的系统，其中所述控制器用于丢弃所选队列的至少一个数据包，该数据包与具有第一层级的组中队列组组件的最高丢弃优先权的队列组组件相关。
39. 如权利要求 38 所述的系统，其中所述控制器用于丢弃所选队列的多个数据包，在所有数据包被丢弃之前，无论该与所选队列相关的队列组组件的丢弃优先权是否改变成不同于该组的最高丢弃优先权。
40. 如权利要求 35 所述的系统，其中所述控制器用于丢弃所选队列的至少一个数据包，该数据包与具有第一层级的组中队列组组件的大约最高丢弃优先权的队列组组件相关。
41. 如权利要求 40 所述的系统，其中所述具有大约最高丢弃优先权的队列组组件具有一个丢弃优先权，该优先权在丢弃值的最高优先权范围内，所述丢弃值与第一层级队列组组件的组中的队列组组件的至少一个相关。

42. 一种用于分级地存储打包的数据并传送该打包数据的方法，其包括：

在输入端接收数据包；和

根据分级结构以队列方式存储接收到的数据包，在该分级结构中，每个层级都包括至少一个队列组组件，并且在该分级结构中，不同于最低层级的其他每个层级的至少一个队列组组件都包括队列组组件的组，该组与分级结构的逻辑低层级有关，每个队列组组件与逻辑低队列和逻辑低队列组中的至少一个相关。

43. 如权利要求 42 所述的方法，其中所述较低层级在分级结构中直接逻辑地较低。

44. 如权利要求 42 所述的方法，进一步包括确定队列组组件的有效长度。

45. 如权利要求 44 所述的方法，其中确定有效长度是基于与队列有关的队列尺寸、源类型、源尺寸和服务请求中的至少一个。

46. 如权利要求 45 所述的方法，其中有效长度的值直接地随着队列尺寸而变化。

47. 如权利要求 46 所述的方法，进一步包括丢弃所选队列的至少一个数据包，该队列与队列组组件相关，所述队列组组件具有研究用于丢弃的分级结构的组中最长的有效长度。

48. 如权利要求 47 所述的方法，其中研究的分级结构队列组与分级结构中较高层级的队列组组件相关，该组件具有与之相关的队列的最长有效长度。

-
- 49.** 如权利要求 48 所述的方法，其中至少一个数据包的丢弃包括丢弃所选队列的多个数据包，在所有数据包被丢弃之前，无论与所选队列相关的队列组组件是否中断具有该组中最长有效长度。
- 50.** 如权利要求 44 所述的方法，其中用于至少两个不同的分级结构组件的有效长度的确定被不同地确定。
- 51.** 如权利要求 44 所述的方法，进一步包括队列的至少一个数据包的丢弃，该数据包与具有一个有效长度的队列组组件相关，所述有效长度在有效长度的最大长度范围内，该范围与研究的用于丢弃的分级结构组相关。

数据包丢弃

背景技术

本发明涉及电子通信，尤其涉及包交换数据网络。

在数据包交换网络中，数字信息包由信号源发出并传送到接收站，根据其间不同链路的承载力和可用性，常常通过不同的路由。数据由连接节点的链路发送。数据可以通过枝发于节点的不同链路发送。

不同于电话网络中所有的通信量都通过固定带宽的线路，其中在任一点的所有线路带宽的总和不超过连接网络节点的速度，在数据包交换网络中，经常有这样的情况，希望通过网络组件之间的通信量的带宽总和超过连接速度。当更多的数据包到达节点而不能被传送时，这种情况被视为过载或拥塞。不能立即传送的数据包可以被丢弃掉或存储到缓冲器中以备随后传送。由于存储空间限制，如果过载的情况持续下去，那么最终节点缓冲器的容量将被耗尽，那么所有引入的数据包都将被丢弃，直到数据包的传输释放出一些缓冲器空间。

数据包在过载情况下，交换网络通常由多用户共享使用，这些用户可能具有不同的服务请求。该数据包交换网络能够在过载的情况下有选择地丢弃数据包以尝试满足这些不同的用户需求。用户基本上是可以使得数据包在网络中传输的任何事物，如：人类、计算机的运行程序、与网络连接的硬件设备或它们

的组合。尽管用户可能具有不只一个的信息流，但信息流的项可以被用户交互使用。

用来选择数据包丢弃的具有 FIFO 时序调度/缓冲机构的系统通常包括根据其重要性和当数据包到达时队列的长度，丢弃信息量。在这些机构中，当缓冲器部分地满时，低优先权的通信量被丢弃，当缓冲器再满时，高优先权的通信量被丢弃。应用这样的机构，很难“调协”丢弃阈值使得该系统高效运作且满足所有服务请求。在某些情况下，得到一个数据而不丢弃其他数据是不可能的。

发明概述

本发明的实施方式提供了用于丢弃基于优先权的分级结构的数据包交换网络的数据包。

通常，在一方面，本发明提供了一种用于分级地存储打包数据并传送该打包数据的系统。该装置包括一个用于接收数据包的输入端，一个耦合到该输入端并用于存储数据包的存储器，一个耦合到该存储器并用于传送存储器中的数据包的输出端，和一个耦合到该存储器并用于控制存储器以队列形式存储数据包的控制器，所述队列与一分级结构相关，在该分级结构中，一第一层级包括一队列组组件的组，其中第一层级的该组中的至少一个队列组组件包括一与该分级结构中一第二层级相关的队列组组件的组，所述第二层级不同于所述第一层级。

本发明的实施方式可以包括一个或更多的下面的特征。每个队列组组件与一逻辑低队列和一逻辑低队列组中的至少一个相关，所述控制器用于控制存储器在分级结构的 N 层级上存储数据包，其中 N 大于 2 且不同于所述两个逻辑最低层级的其他

每个层级的至少一个队列组组件都包括与一逻辑低层级相关的至少两个队列组组件。该与一逻辑低层级相关的至少两个队列组组件与该分级结构的直接逻辑低层级相关。所述控制器用于控制存储器存储与该分级结构中第一层级相关的队列组组件组成的多个组。

本发明的实施方式还可以包括一个或更多的下面的特征。每个队列组组件与一逻辑低队列和一逻辑低队列组中的至少一个相关，所述控制器用于确定第一层级队列组组件组成的组中至少一个队列组组件的丢弃优先权，该优先权用于丢弃数据包的选择。所述控制器用于根据与一队列有关的队列尺寸、源类型、源尺寸和服务请求中的至少一个确定丢弃优先权。该控制器用于确定丢弃值。所述丢弃值直接地随着队列尺寸而变化。所述控制器用于确定第二层级的至少一个队列组组件的丢弃优先权，该优先权用于丢弃数据包的选择。所述控制器用于确定用于丢弃数据包的选择的丢弃优先权，该选择用于第一层级的至少一个队列组组件和不同地第二层级的至少一个队列组组件。该控制器用于丢弃队列的至少一个数据包，该数据包与具有一个第一层级的组中队列组组件的最高丢弃优先权的队列组组件相关。在第一层级的队列组组件组成的组是第一组，其中如果具有第一组的队列组组件的最高丢弃优先权的队列组组件包括一个第二层级的队列组组件的第二组，那么该控制器确定一个丢弃优先权用于第二队列组组件的组的每个队列组组件，并且丢弃至少一个队列的数据包，该队列与具有第二组中的最高丢弃优先权的第二组中的队列组组件相关。至少一个被控制器丢弃的数据包在队列的头、队列的尾和队列的随机位置的任何一处处理，所述队列与具有第一组队列组组件的最高丢弃优先权的队列组组件相关。所述控制器用于丢弃队列的多个数据包，该队列与具有最高丢弃优先权的队列组组件相关，无论该

丢弃优先权在所有的数据包丢弃之前是否改变。所述控制器用于确定丢弃值，该丢弃值用于至少一个第一层级的组中的队列组组件，从不同的意义上说，而不是用于至少一个第一层级的组中的其他队列组组件。所述控制器用于确定丢弃值，该丢弃值用于分级结构的第一层级的队列组组件，该分级结构包括分级结构的第二层级的队列组组件的组，基于该主要组件的有效尺寸和队列的有效尺寸的至少一个，所述队列在分级结构的一个逻辑低层级而不是第二层级，所述层级与相关于该主要组件的第二层级的至少一个组件相关。所述控制器用于丢弃队列的至少一个数据包，该队列与具有大约是第一层级的组的队列组组件的最高丢弃优先权的队列组组件相关。所述控制器用于丢弃队列的至少一个数据包，该队列与具有丢弃优先权的队列组组件相关，该优先权在一个丢弃值的最高优先权的范围内，该丢弃值与第一层级的队列组组件的组中至少一个队列组组件相关。所述控制器用于丢弃队列的至少一个数据包，该队列与响应于存储器的队列组组件相关，该存储器尝试存储将超出该存储器数据范围的一些数据。所述控制器用于丢弃队列的至少一个数据包，该队列与响应于存储器的队列组组件相关，该存储器尝试存储将超出队列组的数据数量范围的一些数据。所述的至少一个数据包从不同于下述队列的队列中丢弃，所述的队列与存储器尝试超出其数据范围的组相关。所述的至少一个数据包被丢弃，无论存储器是否尝试存储一些数据，该数据将超出存储器数据量的界限。所述的控制器用于丢弃与队列组组件相关的队列的至少一个数据包，根据下面的至少一个：响应于至少尝试存储将超出数据量界限（相关于该系统）的一些数据的存储器，和提前随机地丢弃。所述的数据量界限是队列尺寸界限。

通常，在另一方面，本发明提供了一个用于分级地存储打包的数据并传送该打包数据的系统，该装置包括用于接收数据包的一个输入端，耦合到存储器的一个输出端，用于传送存储器的数据包，和耦合到该输入端、输出端的控制装置，用于以队列方式存储输入端接收到的数据包，该队列与分级结构相关，该分级结构中每个层级都包括队列组组件的至少一个组，且该分级结构中不同于分级结构中最低层级的每个层级上的至少一个队列组组件包括一个队列组组件的组，所述组与分级结构的逻辑低层级相关，每个队列组组件与逻辑低队列和一个逻辑低队列组中的至少一个相关，所述控制装置还用于传送存储的数据包给输出端。

本发明的实施方式可以包括下面的一个或更多的特征。所述的逻辑低层级是在分级结构中直接逻辑较低。所述控制装置还用于确定队列组组件的有效长度。所述控制装置用于根据与队列相关的队列尺寸、源类型、源尺寸和服务请求中的至少一个来确定有效长度。有效长度的值直接地随着队列尺寸而变化。所述控制装置还用于丢弃所选队列的至少一个数据包，所述队列与队列组组件相关，该组件具有研究用于丢弃的分级结构的组中最长的有效长度。研究的分级结构的组与分级结构的较高层级的队列组组件相关，该组件具有与之相关的组的最长有效长度。所述控制装置用于丢弃所选队列的多个数据包，无论与所选队列相关的该队列组组件在所有的数据包被丢弃之前，是否中断具有它的组中最长的有效长度。所述控制装置用于确定至少两个不同的分级机构组件的有效长度。所述控制装置用于丢弃队列的至少一个数据包，该队列与具有一有效长度的队列组组件相关，该有效长度在有效长度的最大长度的范围内，所述范围与研究用于丢弃的分级结构组相关。

通常，在另一方面，本发明提供了一个用于分级地存储打包的数据并传送该打包数据的系统，该系统包括一个用于接收数据包的输入端，一个耦合到该输入端的存储器，用于存储数据包，一个耦合到存储器的输出端，用于传送存储器的数据包，和一个耦合到存储器的控制器，用于控制存储器以队列的方式存储数据包，所述队列与分级结构相关，在该分级结构中，第一层级包括一个队列组组件的组，其中第一层级的该组中至少一个队列组组件包括一个队列组组件的组，该组与分级结构的第二层级相关，所述第二层级不同于第一层级，每个队列组组件与逻辑低队列和逻辑低队列组中的至少一个相关，其中所述控制器用于控制存储器在分级结构的 N 层级上存储数据包，其中的 N 大于 2 并且不同于两个逻辑最低层级的每个层级的至少一个队列组组件包括至少一个队列组和一个其他的与逻辑低层级相关的队列组组件，其中所述控制器用于确定一个丢弃优先权，该优先权是第一层级队列组组件的组中每个队列组组件的优先权，用于丢弃数据包的选择，基于丢弃值的该丢弃优先权，所述丢弃值直接随着与队列组组件有关的队列的队列尺寸而变化。

本发明的实施方式可以包括下面的一个或更多的特征。所述控制器用于根据与队列有关的源类型、源尺寸和服务请求中的至少一个确定丢弃优先权。所述控制器用于确定用于不同的分级结构队列组组件的丢弃数据包的选择的丢弃优先权。所述控制器用于丢弃所选队列的至少一个数据包，该数据包与具有第一层级的组中队列组组件的最高丢弃优先权的队列组组件相关。所述控制器用于丢弃所选队列的多个数据包，在所有数据包被丢弃之前，无论该与所选队列相关的队列组组件的丢弃优先权是否改变成不同于该组的最高丢弃优先权。所述控制器用于丢弃所选队列的至少一个数据包，该数据包与具有第一层级

的组中队列组组件的大约最高丢弃优先权的队列组组件相关。所述具有大约最高丢弃优先权的队列组组件具有一个丢弃优先权，该优先权在丢弃值的最高优先权范围内，所述丢弃值与第一层级队列组组件的组中的队列组组件的至少一个相关。

通常，在另一方面，本发明的特征在于一个用于分级地存储打包的数据并传送该打包数据的方法，该方法包括在一个输入端接收数据包，和根据分级结构以队列方式存储接收到的数据包，在该分级结构中，每个层级都包括至少一个队列组组件，并且在该分级结构中，不同于最低层级的其他每个层级的至少一个队列组组件都包括队列组组件的组，该组与分级结构的逻辑低层级有关，每个队列组组件与逻辑低队列和逻辑低队列组中的至少一个相关。

本发明的实施方式可以包括下面一个或更多的特征。所述较低层级在分级结构中直接逻辑地较低。该方法可以进一步包括确定队列组组件的有效长度。确定有效长度是基于与队列有关的队列尺寸、源类型、源尺寸和服务请求中的至少一个。有效长度的值直接地随着队列尺寸而变化。该方法还可以进一步包括丢弃所选队列的至少一个数据包，该队列与队列组组件相关，所述队列组组件具有研究用于丢弃的分级结构的组中最长的有效长度。研究的分级结构队列组与分级结构中较高层级的队列组组件相关，该组件具有与之相关的队列的最长有效长度。至少一个数据包的丢弃包括丢弃所选队列的多个数据包，在所有数据包被丢弃之前，无论与所选队列相关的队列组组件是否中断具有该组中最长有效长度。用于至少两个不同的分级结构组件的有效长度的确定被不同地确定。该方法可以进一步包括队列的至少一个数据包的丢弃，该数据包与具有一个有效长度

的队列组组件相关，所述有效长度在有效长度的最大长度范围内，该范围与研究的用于丢弃的分级结构组相关。

本发明的多个实施方式可以提供下面所说的一个或更多的优点。可以实现数据网络的高效利用。通过路由选择数据，不同的服务请求可以被满足。可以决定关于数据包到达节点之后的数据包的丢弃。

这些和其他优点及本发明自身将在后面的附图、实施方式的具体描述和权利要求中体现。

附图说明

附图 1 为通信系统的示意图。

附图 2 为如附图 1 所示的路由器的示意性框图。

附图 3 为队列分级结构的示意性框图。

附图 4 为处理丢弃数据包的流程框图。

优选实施方式

参考附图 1，通信系统 100 包括信号源 102，104，106 和如因特网的数据交换网络 108，以及接收站 110，112，114，116。通过网络 108，路由器或节点 118，120，122，124，126 将信号源 102，104，106 的数据发送到接收站 110，112，114，116。所示的每一个连接都设置为可双向通信的。信号源的实施例可以包括 Internet 服务提供者 (ISPs)，单位，单位的部门，各等级用户，和独立用户。

参考附图 2，为如附图 1 所示的典型的路由器 118，包括处理器 130 和缓冲存储器 132。所述处理器 130 与所述缓冲器 132 耦合可双向通信。数据包可以存储在缓冲器 132 中，该数据包与已存储在缓冲器 132 中的数据包结合（参见附图 3 和下面与之相关的描述），所述数据包从缓冲器 132 传输出时由处理器或控制器 130 控制。处理器可以为如个人电脑的计算机中央处理单元（CPU），所述处理器执行软件代码指令，以执行不同的如上所述以及如下所述的功能。或者，处理器 130 可以为基本上独立于存储于内存中的软件的硬件。处理器 130 的硬件设备执行可以比软件控制处理器速度更快，并且硬件设备优选地用来执行网络 108 以外的数据连接，基于软件控制的设备用来执行网络 108 内部的数据连接。处理器的功能可以由软件，硬件，固件，和/或硬布线或它们的组合，也可以由安置在特定区域的不同于以上描述的设备来完成，包括在系统 100 中分布。

参考附图 3，与数据包交换网络节点（存储于如附图 2 的路由器 118 中的缓冲器 132 中）相关的数据包分级结构 10 包括三个队列组层级 12，14，16 和一个数据包队列层级 18。分级结构中的各层级相对其它层级逻辑低或高。尽管示出了三层队列组层级，但仍可接受其它数量的队列组如一个，两个或多于三个。如图所示，层级 12 包括队列组 20，层级 14 包括队列组 22，24，层级 16 包括队列组 26，28。

在每个队列组 20，22，24，26，28 中具有队列组组件，例如，组 20 的队列组组件 30，组 22 的队列组组件 32 和组 26 的队列组组件 34。所述队列组组件包括/安置有队列（如队列中的存储器）和/或分级结构 10 中低层级的队列组（作为队列组的参考）。所述队列组件表示了接收到并缓冲的数据量，该数据量由节点处发送。队列组中高于最低层级的层级 12、14，这里指层

级 12、14，队列组 20、22、24 中所示的队列组组件定位低层级队列组，即队列组 22、24（在组 20 中）和 26、28（在组 22 中）。最低层级队列组中示出的队列组组件，这里指组 26、28，定位数据包的组（虽然示出了这样的组件和队列之间的一对一通信）。每一个数据包可显示地或隐式地根据程控规则关联组。数据可以根据不同的规则与组关联，例如端口号码和/或终端等级。例如，数据包可以根据端口号码与层级 12 的队列组组件相关联，并且可以根据终端等级与层级 14 的队列组组件相关联，或者反之亦然。

队列组中还具有用于确定分级结构 10 中的组以指定用于丢弃的机构。该机构指示作为丢弃目标的队列或估计确定队列的在分级结构中进一步向下的队列组。例如，该机构可以是共享公共属性的队列和/或队列组，例如，有效长度范围或丢弃值（将在下面论述），与队列存储段相关联的形式。

队列组中的组件可以被指定优先丢弃。除非队列是加权的，否则最长的（请求最大的缓冲空间）队列组组件（即，与请求最大缓冲空间的队列相关联）将被指定一个丢弃值，与其他组件的组中的值相比较，建立队列或队列组，其中组件最可能地具有最高优先级用于选择丢弃数据包，即为了队列或队列组进一步估计确定丢弃的队列。于是，队列的队列尺寸或作为队列组尺寸的一部分用于确定丢弃值。该丢弃值可以基于一个或更多的系数进行调整/加权。例如，队列组组件可以指定加权，该加权基于队列源的类型、队列源的尺寸、或队列源提出的和/或与队列相关的服务请求。在最后一个例子中，一个队列源能够获得一个首选的加权通过提出一个首选的服务请求。例如，一个人可以得到比通过 ISP 得到的更多的带宽。

加权队列调整队列的有效长度，至少为了达到最长队列丢弃（LQD）的目的。加权使队列丢弃的可能性客观地不同，如，较低的，尽管它没有经过加权可能不会被丢弃，又如，如果加权不调整队列丢弃的可能性以低于队列组中另一个队列丢弃的可能性，或者调整另一个队列丢弃的可能性高于所述的队列丢弃的可能性。队列的最适宜的有效长度，或其他丢弃值直接地随着队列的尺寸（请求的缓冲空间）而改变。该改变可以是线性的，但不是必须的。该改变还可以是与队列尺寸成反比例的。有效长度可以用作用来确定队列的丢弃优先权的丢弃值。同样地，队列组组件的加权调整一个与每一个存储段相关的有效的缓冲空间请求。丢弃的可能性和/或量值可以由于分级结构的部分被不同地确定，这些部分例如：不同的队列组组件、不同的层级、和/或不同的队列。

可以使用多个公式来确定加权系数用于队列或队列组组件。例如，根据已经（或将要）用于处理队列或结合组件的队列的程度，更多的权被提供给队列或队列组组件。在又一个例子中，如果队列 1（或组件 1）的有效长度会变的长一点以处理队列 2 的数据包（或结合组件 2 的数据包）的话，队列 1（或组件 1）的有效长度可以被确定的比队列 2（或组件 2）的长度小，这样将在最后处理的数据包将被首先丢弃。然而首选地，无论选择什么公式都将引起队列或队列组组件客观上更可能地成为选择丢弃的目标，或者成为有关队列的组件的目标，那么就需要更多的带宽和/或缓冲空间。

使用具有加权的有效长度的 LQD 是被丢弃的最长的有效长度队列（如，指标队列的一个或更多的数据包将被丢弃）。数据包被存储丢弃的数据包的去分配和/或再分配存储器丢弃，这样其他数据可以写入去分配和/或再分配存储器。对于队列组组件，

选择丢弃的优先权（有效长度）被同样地确定，且其是具有最高有效缓冲空间请求的服从于丢弃选择的队列组组件。

然而，由于与组中其他队列或组件相关的队列或队列组组件，队列或队列组组件的最高丢弃优先权不需要最高的丢弃值。例如，该最高丢弃优先权可以与具有最低丢弃值或最接近某个数的丢弃值的队列或队列组组件相关，在组中任意确定或者用不同的方法。同样地，多个组件可以具有相同的丢弃值。

在分级结构 10 的每个层级 12、14、16 上的队列组组件和在层级 18 上的队列具有最高丢弃可能性，他们服从于丢弃选择。这个丢弃可能性依赖于队列或队列组组件的缓冲空间请求和队列或队列组组件的任何加权。

参考附图 4，并进一步参考附图 1-3，丢弃数据包的过程操作 50 包括阶段 52，该阶段决定分级结构 10 中的节点缓冲器容量是否被超出。所述容量可以是物理超出（没有更多的空间），或管理性的超出（即规则禁止进一步存储或至少启动数据包丢弃）。如果容量被超出，在下面的阶段 54 中，数据包（与一个队列并且可能地与一个或多个队列组结合的）将作为被丢弃的目标。分级结构 10 的层级进行分析确定丢弃的数据包对象，优选的为从高层级到低层级，更优选的为从最高层级到最低层级。由此，在层级 12 上，根据其在队列组中与其他丢弃值相关联的丢弃值，所述队列组 20 经过分析确定具有最高丢弃优先权的队列组组件。这里，组件 30 与队列组 22 比较，确定组件 30 具有最高优先权。因此，在层级 14 的队列组 22 对其具有最高丢弃优先权的队列组组件进行分析。假设队列组组件 32 在层级 14 中被选为丢弃对象，经过分析层级 16 中的队列组 26 确定组件 34 具有最高丢弃优先权。经过分析层级 18，特别是分析任何与组件 34 相应的队列，确定具有最高丢弃优先权的队列。该队列，

在这里为队列 36，是存储段 34 中的唯一的队列，成为被选择的丢弃对象，并且一个或更多个该队列的数据包在阶段 54 被丢弃。所述被选择丢弃的队列/队列组组件不必是最长的，但大约是最长的队列/队列组组件。所述队列/队列组组件可以结合长度范围（存储段），并且在此范围内，具有最长长度范围并具有至少一个数据包的数据包/队列组组件可能选为要被丢弃的队列/队列组组件（最高丢弃优先权的队列/队列组组件）。

选定丢弃的对象是根据丢弃值/可能性持续直到分级结构 10 的节点缓冲器空间容量没有被等待节点发送的队列总和超出，或直到预定的数据包的数量被丢弃。

过程 50 可具有重新调整的阶段，添加，或在不脱离本发明的情况下删除。例如，可由低层级到高层级分析分级结构 10 的各层级。同样，通常随着过程 50 发生的数据包排队和出列会影响到被丢弃的队列和缓冲器容量是否被超出。

下文提供了本发明实施方式的进一步描述。

本发明的实施方式提供了数据包推出或最长队列丢弃技术（LQD）。本发明的实施方式提供了一种系统，该系统具有“分流”等待使得当数据包到达队列时，不必对数据包作出丢弃决定。在 LQD 系统中，直到缓冲器完全占满时，才对数据包进行丢弃，但可能（即，出于如安放特定数据流最大队列尺寸的策略的原因）在缓冲器完全占满之前就进行数据包丢弃。当缓冲器完全满时，或由于其他任何原因数据包将被丢弃时，队列中的数据包被选择丢弃。

下面的部分描述了一套以硬件形式执行的分级结构推出的结构和方法。在该部分，假设存储段队列在层级结构 10 中使用。

数据包队列

所述数据包队列存储了接收到的和等待传送的数据包。

名称	尺寸(bits)	备注
Queue group	18	指向队列的成员的队列组
Prev Bqueue	18	指向在缓冲器列表中的在前队列
Next Bqueue	18	指向在缓冲器列表中的下一个队列
Next Squeue	18	指向调度程序列表中的下一个队列
Head Packet	18	指向队列中的第一数据包
Tail Packet	18	指向队列中的最后数据包
Head Length	12	队列头数据包的长度

表 1 数据包队列类型

如果该组是由如数据包 ID 推断，队列组指针可能不会得以使用。

数据包队列组

数据包队列组用以存放队列列表以执行最长队列推出策略。

名称	尺寸 (bits)	备注
Queue group	18	指向队列成员的队列组
Length	25	队列的字节长度
Prev Gqueue	18	指向列表中的在先队列

Next Squeue	18	指向列表中的下一队列
Bucket Mask	15	用于说明具有入口的存储段的位屏蔽，用于确定最大的队列
Size N Queue Head	18	指向长度大小为 $(N*2K) - ((N-1)*2K)$ 队列列表中的头队列
Size n Queue Tail	18	指向长度大小为 $(N*2K) - ((N-1)*2K)$ 队列列表中的尾队列
Size 2 Queue head	18	
Size 2 Queue Tail	18	
.....		
Size 256 Queue Head	18	指向长度大小为 $128*2K$ 或更大的队列列表中的头队列
Size 256 Queue Tail	18	指向长度大小为 $128*2K$ 或更大的队列列表中的尾队列

表 2 数据包队列组

存储段屏蔽指示出了哪个存储段不是空的。例如，在位屏蔽中，一个二进制或逻辑的 1 指示非空，二进制或逻辑的 0 指示空的。位屏蔽可以帮助很快地确定哪个存储段是非空的，但是其他技术也可以用于确定上述的存储段。

由于分级结构中处于较高层级的数据包队列组，“尺寸为 N 的队列”存储器数量应该大一些，给上面的层级 64K 大小的数据包或更多。

所述数据包队列组用于执行数据包缓冲存储器分级结构的管理。

缓冲器管理算法

下面是对分级结构 10 中用于处理数据的技术的伪码说明。下面的伪码使用速记法，这对 C 语言程序的普通技术人员来说是很熟悉的。“old”和“new”变量用于在处理数据时，帮助调节改变加权。

入队数据包：

```
1.ENQUEUE_PKT(Q,PKT)
2.{
3.Old_Q_Length=Q->Length
4.IF(Q->Tail==NULL) {
5.Q->Tail=PKT
6.Q->Head=PKT
7.}ELSE {
8.Q->Tail->Next=PKT
9.PKT->Prev=Q->Tail
10.Q->Tail=PKT
11. }
12.
13.L_0_Buff=Q->Queue_Group;
14.L_1_Buff=L_0_Buff->Queue_Group;
15.L_2_Buff=L_1_Buff->Queue_Group;
16.UPDATE_Q(Q,L_0_Buff,PKT->Length)
```

```

17.UPDATE_Q(L_0_Buff,L_1_Buff,PKT->Length)
18.UPDATE_Q(L_1_Buff,L_2_Buff,PKT->Length)
19. }

```

所述入队队列算法处理进入的数据包。所述数据包将被安放在所述队列中，检查所述队列以确定它是否是空的（线 4）。如果是这样，就确定所述数据包为所述队列的头和尾（线 5—6）。否则，制定下一个尾指针以指向新的数据包（线 8）。确定新数据包，完成与先前数据包的尾链接和双向链接（线 9—10）。队列组链接有效地将该数据包与相应的链接列表（线 13—15）的末端相加。相应的存储段由于队列的新尺寸而被更新，所述数据包被添加进所述队列（线 16—18）。

出列数据包：

```

1.DEQUEUE_PKT(Q,C,SC)
2.{
3.PKT=Q->Head
4.IF(Q->Tail==PKT)
5.Q->Head=NULL
6.Q->Tail=NULL
7.}ELSE{
8.Q->Head=PKT->Next
9.}
10. L_0_Buff=Q->Queue_Group;
11. L_1_Buff=L_0_Buff->Queue_Group;
12. L_2_Buff=L_1_Buff->Queue_Group;
13. UPDATE_Q(Q,L_0_Buff,-1*PKT->Length)

```

14. UPDATE_Q(L_0_Buff,L_1_Buff,-1*PKT->Length)
15. UPDATE_Q(L_1_Buff,L_2_Buff,-1*PKT->Length)
16. Return PKT
17. }

所述队列数据包算法处理输出的数据包。如果将被移出的数据包是队列（线 4）中唯一的数据包，那么表示该队列通过清空头和尾（线 5—6）而变空。否则，所述头指针指向被移出数据包（线 8）后的数据包。队列组链接有效地从相应的链接列表（线 10—12）中移出该数据包。所述相应的数据包由于队列的新尺寸而被更新，所述数据包从所述该队列中移出（线 13—15）。

推出数据包：

- 1.PUSHOUT_PKT(L_N_Buff,Depth)
- 2.{
- 3.Bucket=FIND_FIRST_1(L_N_Buff->Bucket_Mask)
- 4.IF(Depth>0) {
- 5.L_(N-1)_Buff=L_N_Buff->Buckets[Bucket]->Head
- 6.Length=PUSHOUT_PKT(L_(N-1)_Buff,Depth-1)
- 7.Q=l_(N-1)_Buff;
- 8.}ELSE{
- 9.Q=L_0_Buff->Buckets[Bucket]->Head
10. PKT=Q->Tail
11. IF(PKT==Q-Head) {/*pushing out only packet in queue*/
12. Q->Head=Q->Tail=NULL
13. }ELSE {/*push out tail packet*/
14. Q->Tail=PKT->Prev

```

15. }
16. /*Add PKT to free lists*/
17. Length=PKT->Length
18. }
19. UPDATE_Q(Q,L_N_Buff,-1*Length)
20. RETURN Length
21. }

```

数据包推出算法递归处理分级结构 10 的不同层级。给出处理中的层级的数和将被处理的组（线 1）。确定从中丢弃数据（线 3）的组是非空的。推出算法访问该组，接着数据包推出算法被调用（线 5—6）。这种情况持续直到浓度为 0（线 8）。如果尾数据包也是头数据包（即，仅仅是一个数据包），那么所述头和尾数据包均是空值（线 11—12）。否则，所述尾数据包被推出（线 13—14）。该丢弃的数据包长度返回且队列和队列存储段通过分级结构 10（线 17—20）被更新。

更新队列：

```

1.UPDATE_Q(Q,C,PKT_Length)
2.{
3.Min_Q=Q->Min_Q_Length
4.Q_Weight=Q->Q_Weight
5.Old_Length=Q->Length
6.Q->Length=Q->Length+PKT_Length
7.Old_Length=(MAX((Old_Length=Min_Q,0))>>Q->Weight
8.New_Length=(MAX((Q->Length-Min_Q,0))>>Q->Weight
9.Old_Bucket=FIND_FIRST_1(Old_Length)

```

```
10. New_Bucket= FIND_FIRST_1(New_Length)
11. Old_Bucket= MIN(Old_Bucket,C_>Num_Buckets)
12. New_Bucket= MIN(New_Bucket,C_>Num_Buckets)
13.
14. IF(Old_Bucket==New_Bucket)Return
15.
16. IF(Old_Bucket>0) {
17.   BUCKET_REMOVE(Q,C,Old_Bucket)
18. }
19. IF(New_Bucket>0) {
20.   BUCKET_INSERT(Q,C,New_Bucket)
21. }
22. }
```

更新队列可以用于完成影响队列（推出、入列、出列）的任何操作。最小队列长度和相应于队列尺寸范围（即，在 2^n 和 $2^{n+1} - 1$ 之间） n 值（权）为 2^n 的值用作加权系数（线 3—4）。存储（线 5—6）具有推出和出列处理的反向 PKT-Length 值的当前的和什么时间处理的队列长度。根据与 2 的幂次（线 7—8）相关的队列尺寸范围，用二进制移位操作确定长度。使用二进制长度，确定最高存储段（线 9—10）。如果所述队列比最高尺寸的存储段大，那么就使用这样的最高存储段（线 11—12）。如果所述更新队列在同一个存储段中，那么不作任何操作（线 14）。如果所述更新队列在一个新的存储段中，那么该队列就从原来的存储段（线 16—17）中移出，并且插入该新的存储段中（线 19—20）。

存储段移出:

```
1.BUCKET_REMOVE(Q,C,Bucket_Num)
2.{
3.Buckets=C->Buckets
4.Is_Head=(Q==Buckets[Bucket_Num]->Head)
5.Is_Tail=(Q==Buckets[Bucket_Num]->Tail)
6.
7.IF(Is_Head&&Is_Tail){
8.Buckets[Bucket_Num]->Head=NULL
9.Buckets[Bucket_Num]->Tail=NULL
10. C->Bucket_Mask&=~(1<<Bucket_Num)
11. Return
12. }
13.
14. IF(Is_Head){
15. Buckets[Bucket_Num]->Head=Q->Next_Bqueue
16. Return
17. }
18.
19. IF(Is_Tail){
20. Buckets[Bucket_Num]->Tail=Q->Prev_Bqueue
21. Return
22. }
23.
24. /*Q is not at the head or tail,
```



```

25. so it has a queue before and after it */
26.
27. Q->Prev_B->Next_B=Q->Next_Bqueue
28. Q->Next_B->Prev_B=Q->Prev_Bqueue
29. Return
30. }

```

所述存储段移出算法从一个存储段中移出一个队列。提出请求以确定将被移出的队列到底是位于存储段的头部还是尾部（线 4—5）。如果该队列是该存储段中唯一的队列（线 7），那么所述头和尾为空值（线 8—9），并且使用位方法的逻辑与 AND（&）、反向（~）和左移（<<）将相应于该存储段的位屏蔽复位（线 10）。如果该队列是头队列（线 14），那么该头队列指向线的下一个队列（线 15）。如果该队列是尾队列（线 19），那么该尾队列指向移出队列的在先队列（线 20）。如果该队列在头队列和尾队列之间（线 24—25），那么相临于移出队列的队列互相指向（线 27—28）。

存储段插入：

```

1.BUCKET_INSERT(Q,C,Bucket_Num)
2. {
3. Buckets=C->Buckets
4. C->Bucket_Mask|=(1<<Bucket_Num)
5. IF(Buckets[Bucket_Num]->Head==NULL){
6. Buckets[Bucket_Num]->Head=Q
7. Buckets[Bucket_Num]->Tail=Q
8. Return

```

```
9. }ELSE{
10. Buckets[Bucket_Num]->Tail->Next_Bqueue=Q
11. Q->Prev_Bqueue=Buckets[Bucket_Num]->Tail
12. Buckets[Bucket_Num]->Tail=Q
13. Return
14. }
15. }
```

所述存储段插入算法在一个存储段中插入一个队列。所述相应于该存储段的位屏蔽使用位方法的逻辑或 OR (|=) 和上移 (<<) 置位以表明该存储段是非空的。如果所述新队列是头队列和尾队列 (线 5), 那么该头队列和尾队列均指向该新队列 (线 6—7)。否则, 相临于该新队列的队列指向该新队列 (线 10—12)。

上述这些算法在整个分级结构 10 中应用, 虽然这不是必须的。例如, 不同的技术可以在分级结构 10 的不同部分使用, 举例来说, 在同一个组中, 或在不同的组中, 也或在不同的层级。用于计算数据包推出的分级结构不必与其他源分配分级结构一致。

虽然这里的描述是关于网络节点的输出链路上的缓冲器的管理的, 但是这里描述的技术可以用在数据 (或 ATM 单元) 包缓存的任何点上。

其他的实施方式在后面的权利要求的范围和精神中体现。

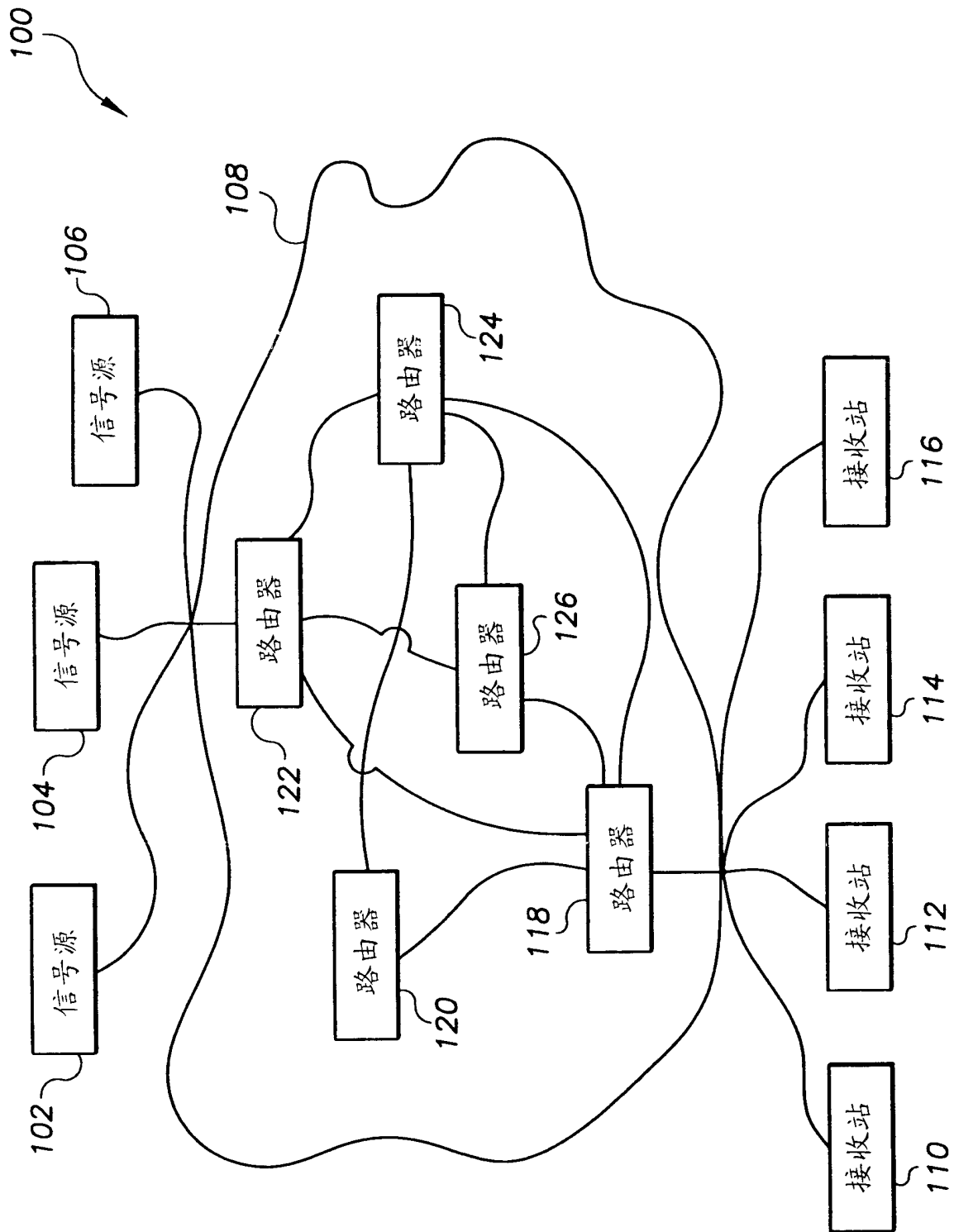


图1

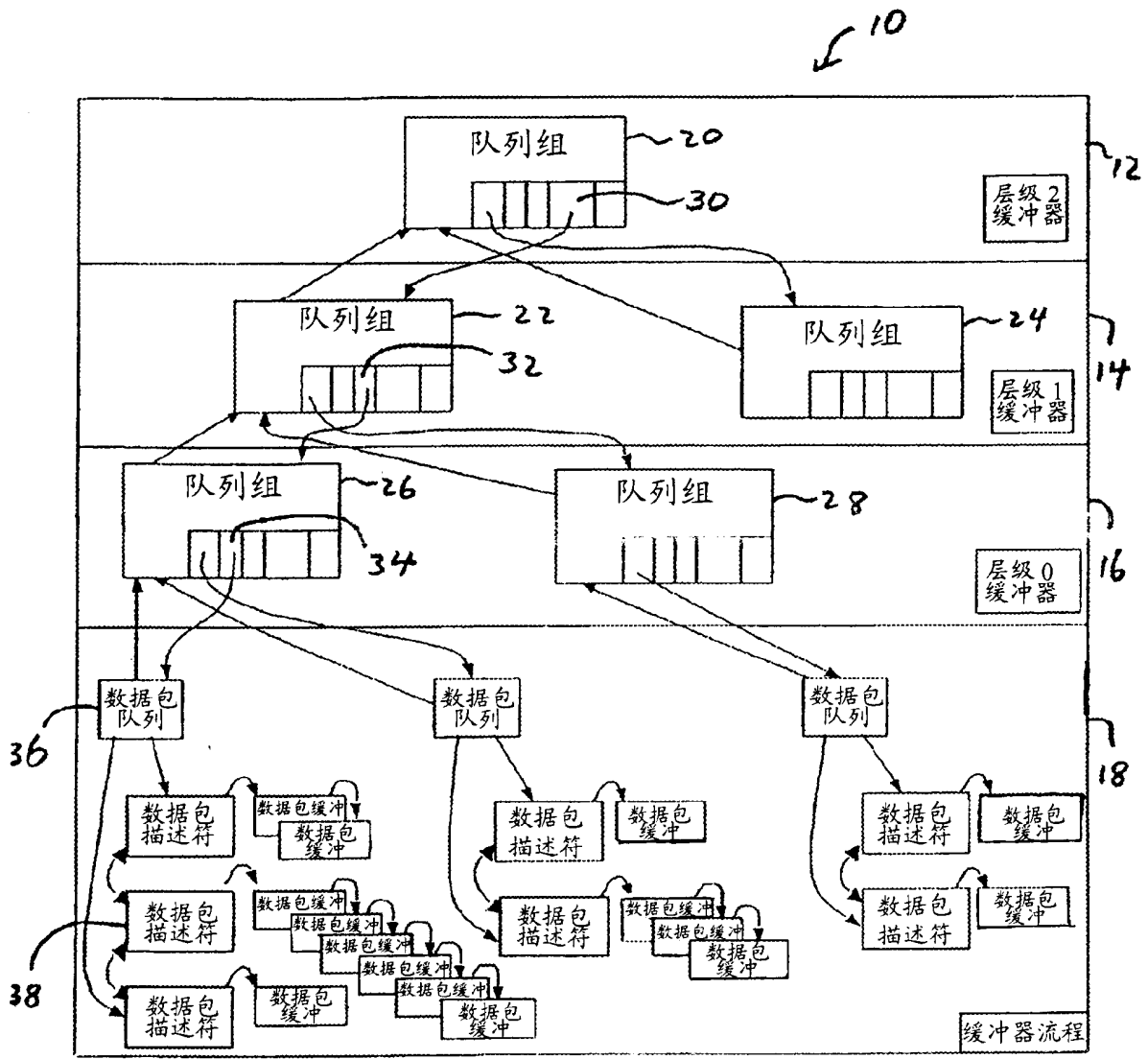
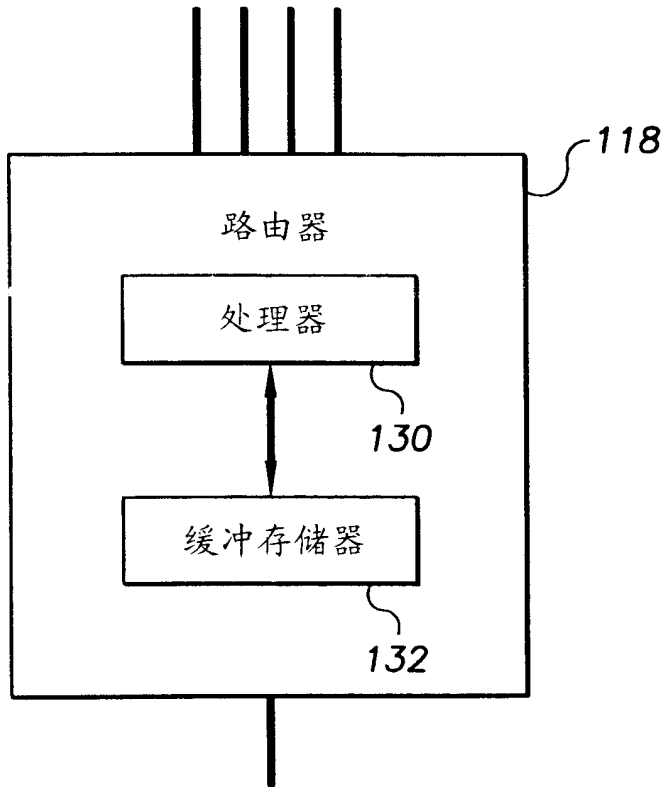


图 3

到达/来自路由器
120, 122, 124, 126



到达/来自接收站
110, 112, 114, 116

图 2

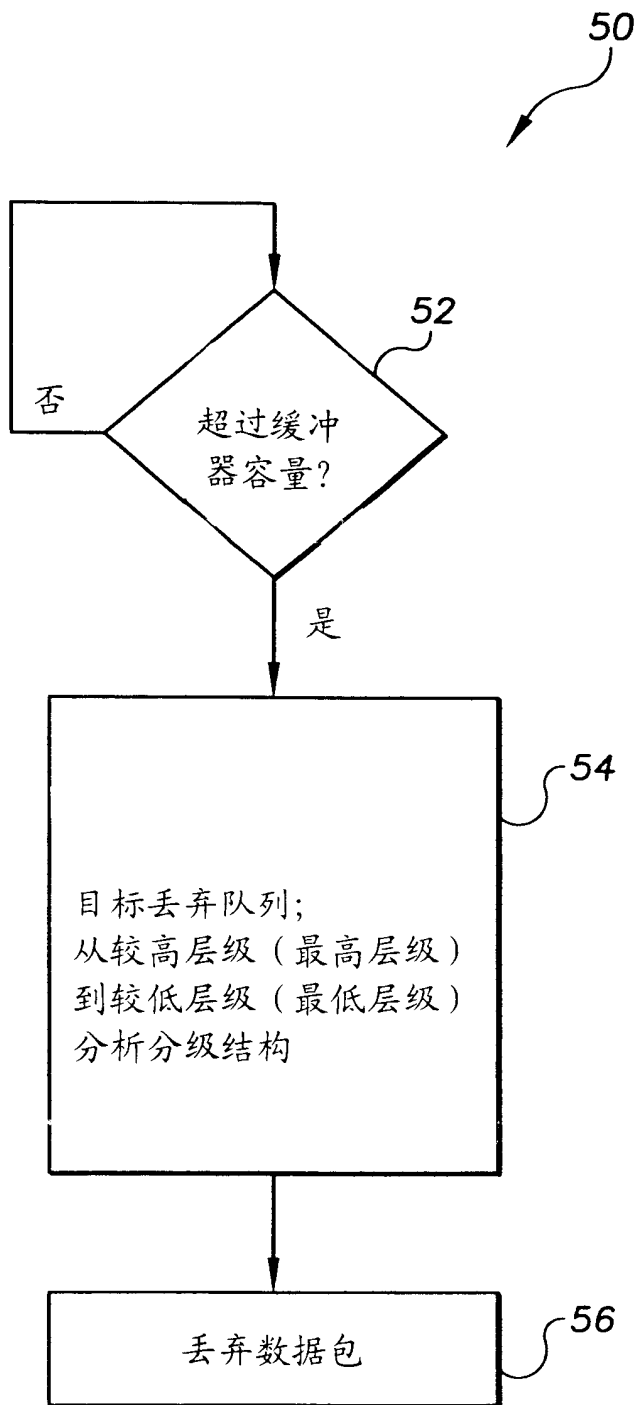


图 4