

[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 200380106643.4

[43] 公开日 2006 年 2 月 1 日

[11] 公开号 CN 1729661A

[22] 申请日 2003.11.18

[21] 申请号 200380106643.4

[30] 优先权

[32] 2002.12.18 [33] EP [31] 02080356.5

[86] 国际申请 PCT/IB2003/005261 2003.11.18

[87] 国际公布 WO2004/056051 英 2004.7.1

[85] 进入国家阶段日期 2005.6.17

[71] 申请人 皇家飞利浦电子股份有限公司

地址 荷兰艾恩德霍芬

[72] 发明人 K·G·W·戈斯森斯

E·里普科马 P·维拉格

[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 李亚非 刘杰

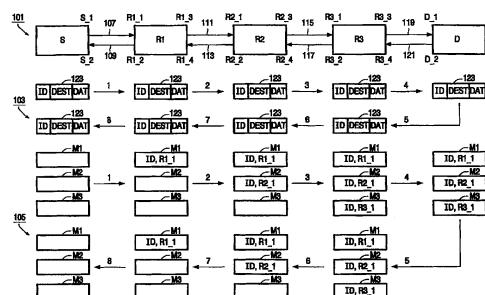
权利要求书 1 页 说明书 12 页 附图 2 页

[54] 发明名称

分组交换网络中返回路径的推导

[57] 摘要

用于传输数据的网络由链接在一起的一组两个和多个节点组成，例如交换机，路由器，或计算机系统。通过网络从源节点传输数据到目的节点。在分组交换网络中，称为分组的小数据单元被路由通过网络从源节点到达目的节点。这些分组也被用来编程网络。在一些情况下要求分组沿返回路径传播到源节点。在本发明中，从存储在网络节点中的信息推导返回路径。



1. 一种确定网络中分组返回路径的方法，该网络包含多个节点和节点间的多个链路，并且其中对于具有至少一个到第二节点的链路的每一个第一节点，在第二节点和第一节点之间存在一个链路，

5 当从源节点经由至少一个中间节点向目的节点发送分组时，使用本方法，

其特征在于，该方法包含在中间节点存储信息用来推导返回路径的步骤。

10 2. 根据权利要求 1 的确定网络中分组返回路径的方法，其特征在于，进一步包含当从源节点发送分组到目的节点时，在分组访问过的每一节点中存储信息用来推导返回路径的步骤。

15 3. 根据权利要求 1 的确定网络中分组返回路径的方法，其特征在于，存储在中间节点中的信息包含分组的标识符和编码中间节点将用来返回分组的输出端口的信息。

4. 一种集成电路，包含网络，该网络具有多个节点和节点间的多个链路，并且其中对于具有至少一个到第二节点的链路的每一个第一节点，在第二节点和第一节点之间存在一个链路，安排该网络当从源节点经由至少一个中间节点向目的节点发送分组时确定分组的返回路径，其特征在于，安排中间节点来存储用于推导返回路径的信息。

20 5. 根据权利要求 4 的集成电路，其特征在于，多个节点中的每一节点被安排用来存储用于推导返回路径的信息。

6. 根据权利要求 4 的集成电路，其特征在于，中间节点被安排用来存储分组的标识符和编码中间节点用来返回分组的输出端口的信息。

分组交换网络中返回路径的推导

技术领域

5 本发明涉及一种确定网络中分组返回路径的方法，该网络包含多个节点和节点间的多个链路，并且其中对于每一个具有至少一个到第二节点的链路的第一节点，在第二节点和第一节点之间存在链路，当从源节点发送分组经由至少一个中间节点到达目的节点时使用本方法。

10 本发明进一步涉及一种集成电路，包含一个网络，该网络具有多个节点和节点间的多个链路，并且其中对于具有至少一个到第二节点链路的每一个第一节点，在第二节点和第一节点之间存在链路，安排该网络当从源节点发送分组经由至少一个中间节点到达目的节点时确定分组的返回路径。

15 背景技术

通常，传送数据的网络包含链接在一起的一组两个或多个设备，该设备被称作节点。网络中的节点可以包含交换机，路由器，或计算机系统。这些计算机系统也可以具有外围设备，这些外围设备对于使计算机系统能够工作是必需的。在网络中两个相邻节点之间的通信路径被称作链路。通过单个传输信道可以实现一个链路。可替换地，在两个节点之间的两个链路可以被组合在单个传输信道中。在不同的网络中，为了提高通信的带宽，两个邻接的节点可以有三个或多个链路用于在这两个节点之间通信。所有这些链路同样可以在单个传输信道中实现。通过网络数据从源节点被传送到目的节点。网络可以被用来，例如在装配在集成电路上的几个元件之间通信，或者在几个计算机系统之间通信。通过网络数据以消息或分组的形式被传送。消息是用户定义的数据单元而分组是网络定义的数据单元。在所谓的消息交换网络中，消息被路由通过网络到达它们的目的地，而在分组交换网络中，分组被路由通过网络到达它们的目的地。在分组交换网络的情况下，应当被发送到指定目的地的消息被划分为几个分组送往目的地。在目的地，搜集消息中的分组并且重新装配为初始消息。分组交换网络的一个优点是，通过把源点和终点之间的通信拆为相对较小的数据单元，它允许以更精细的粒度共享网络

中许多用户之间的同一数据路径。在这个文献的剩余部分中，为了效率的原因，使用词“分组”和“分组交换”，也可以把这些词当作“消息”和“消息交换”。

在分组交换网络中，除发送数据之外，也可以使用分组来对网络编程，⁵ 例如预定(reserve)或释放资源，或建立或消除连接。资源的例子是在路由器中的缓冲器容量或连接的带宽。建立连接的例子是在网络中设置一系列路由器以便经由那个连接从源节点发送一个或多个分组到目的节点。当在许多用户之间共享网络时，仲裁方案组合在单个传输信道上的分组传输。例如，可以使用时分复用(TDM)，其通过给每一个流指定¹⁰ 集合中不同的时隙组合数据流。TDM在时隙的固定时序中通过单个传输信道重复发送数据。

在一些资源预定和连接建立的情况下，例如由于这个动作不能在路径上的节点之一中执行，其中该路径是路由分组所经过的路径，资源预定和连接建立失败。一个例子是由于缺乏资源而失败，例如缺乏在沿着¹⁵ 路径的节点中的缓冲器容量。因此，期望的连接不能被建立。随后，直到路径的那个点撤销才能进行资源预定和建立连接。因此分组重新访问以前访问过的路径节点是必须的，也就是说，它沿返回路径传播到源节点。

US2002/0031095 描述了一种当通过网络发送分组时建立返回路径说明的方法。²⁰ 该网络包含通过在任意网络拓扑的物理点到点连接中的至少两个双向连接接口灵活组网的模块。当模块转发分组到另一个模块时，那个模块的接收接口号被存储在分组中。以这种方式，从存储在分组中并且与分组访问过的模块对应的接收接口列表中推导出返回路径。

现有技术的处理器的一个缺点是，在分组中存储关于返回路径的信息，²⁵ 这会增加分组的大小，特别是在分组只包含目的地址而不是通过网络的路径的完整说明的情况下。

发明的公开内容

本发明的一个目标是提供一种确定网络中分组返回路径的改进的方法，³⁰ 该方法允许减小分组的大小。

通过一种在所陈述的那种网络中确定分组返回路径的方法实现该目标，其特征在于该方法包含在中间节点存储信息用于推导返回路径的步

步骤。在节点中储存关于返回路径的信息，这些节点是分组在它到目的节点的路径上访问过的节点。一旦出现故障，例如由于不能够在特定节点中预定资源，分组从保存在一个或多个节点中的信息中推导它的返回路径，这些节点是分组在它到目的节点的路径上访问过的节点。不要求分组中的额外空间来存储关于返回路径的信息，这允许减小分组的大小。
5

本发明的一个有利实施例的特征在于该方法进一步包含步骤：当从源节点发送分组到目的节点时，在分组访问过的每一节点中存储信息用于推导返回路径，而不是仅仅在分组访问过的有限数量的节点中存储信息或者甚至仅仅在一个中心节点存储信息。关于返回路径的信息被分布10保存在节点中并且当分组沿返回路径传播时，分组从一个节点传播到另一个节点，从每一节点推导关于返回路径的信息。通过在分组访问过的所有节点中存储关于返回路径的信息，对每一个体节点来说可以减小确定返回路径的开销。

本发明一个实施例的特征在于，在中间节点存储的信息包含分组的15标识符和编码中间节点用来返回分组的输出端口的信息。一个优点是这个信息很容易在节点中推导并且唯一标识每一分组的返回路径。

根据本发明，在引言段落中定义的集成电路的特征在于，安排中间节点存储信息用于推导返回路径。因此，可以减小在芯片级通信网络中使用的分组的大小，减小通信开销。
20

根据本发明的集成电路的优选实施例在从属权利要求中定义。

附图简述

图 1 展示了当使用目的地路由从源节点发送分组到目的节点时，使用根据本发明的确定网络中分组返回路径的方法的网络的实施例。

图 2 展示了当使用源路由从源节点发送分组到目的节点时，使用根据本发明的确定网络中分组返回路径的方法的网络的实施例。
25

实施例的描述

图 1 展示了当使用目的地路由从源节点发送分组到目的节点时，使用根据本发明的确定网络中分组返回路径的方法的网络的实施例，也就是说，只有关于最后目的地的信息被存储在分组中。参考 101，展示了30网络，包含节点 S, R1, R2, R3 和 D，这些节点通过它们的输入端口和输出

端口 S_1, S_2, R1_1, R1_2, R1_3, R1_4, R2_1, R2_2, R2_3, R2_4, R3_1, R3_2, R3_3, R3_4, D_1 和 D_2 与链路 107, 109, 111, 113, 115, 117, 119 和 121 耦合。网络 101 可以是网络，或者网络的一部分，或者集成电路的一部分。节点 S, R1, R2, R3 和 D 可以包含发送一数据单元到下一目的地的路由器或交换机。节点 S, R1, R2, R3 和 D 也可以包含用于耦合到其它节点的多个输入端口和输出端口，在图 1 中没有示出。对于所有的节点，S, R1, R2, R3 和 D 保持如果第一节点有至少一个到第二节点的链路，在第二节点和第一节点之间也存在一个链路。节点 S, R1, R2, R3 和 D 已经各自存储了返回关系，将那个节点的每一输入端口与那个节点的每一输出端口关联起来以便当在上述输入端口上接收来自特定节点的分组和通过上述输出端口发送分组时，把返回关系发送到那个特定节点。节点 R1, R2, R3 分别包含存储器 M1, M2 和 M3。节点 D 和 S 也包含一个存储器，在图 1 中没有示出。从源节点 S 发送分组 123 到目的节点 D。安排分组 123 来对网络编程，例如，建立或消除连接，或者预定或释放资源。建立连接的一个例子是为了以期望的方向发送分组把某一节点的输入端口耦合到那个节点的输出端口。资源的例子是路由器中的缓冲器容量或连接带宽。如果在每一节点网络的编程是成功的，分组被路由到目的节点 D。然而，网络编程在某一节点可能失败，例如由于缺乏诸如缓冲器容量之类的资源。在那种情况下为了重新编程网络，分组从前进到源节点 S 的那个确定节点经过返回路径回到源节点 S 是必需的，例如通过释放预定的资源。在这个实施例中假定，一直到目的节点 D 网络编程是成功的。分组 123 包含标识符 ID, 目的地址 DEST 和用于编程网络的数据 DAT。为了知道使用哪个输出端口用于发送分组到希望的目的地，每一节点 S, R1, R2, R3 和 D 已经存储了目的地关系，使所有的目的地与那个节点的输出端口相关联。使用这个信息，给定由那个节点接收的分组的目的地址 DEST，节点能确定使用哪个输出端口将分组发送到邻接节点之一。例如，目的地关系和返回关系都能被编程在可编程序存储器中，该存储器存在于节点 S, R1, R2, R3 和 D 中，在图 1 中没有示出。目的地址 DEST 等于目的节点 D 的地址。参考 103，展示了当从源节点 S 发送分组到目的节点 D 时分组 123 前进的路径。参考 105，展示了当从源节点 S 发送分组 123 到目的节点 D 时存储器 M1, M2, M3 的内容。在第一步骤 1 中，通过输出端口 S_1，链路 107 和输入端口 R1_1，由源节点 S 发送分

组 123 到节点 R1。节点 R1 从分组 123 中读取标识符 ID，并且结合来自输入端口 R1_1 的标识符 R1_1 把它存储在存储器 M1 中，作为一对 ID，R1_1。使用存储在分组 123 中的目的地址 DEST 和它的目的地关系，节点 R1 确定使用哪个输出端口转发分组 123，该端口是输出端口 R1_3。在下一步骤 2 中，经由输出端口 R1_3，链路 111 和输入端口 R2_1 发送分组 123 到达节点 R2。节点 R2 从分组 123 中读取标识符 ID，并且结合来自输入端口 R2_1 的标识符 R2_1 把它存储在存储器 M2 中，作为一对 ID，R2_1。使用存储在分组 123 中的目的地址 DEST 和它的目的地关系，节点 R2 确定使用哪个输出端口转发分组，该端口是输出端口 R2_3。在下一步骤 3 中，经由输出端口 R2_3，链路 115 和输入端口 R3_1 发送分组到达节点 R3。节点 R3 从分组 123 中读取标识符 ID，并且结合来自输入端口 R3_1 的标识符 R3_1 把它存储在存储器 M3 中，作为一对 ID，R3_1。使用存储在分组 123 中的目的地址 DEST 和它的目的地关系，节点 R3 确定使用哪个输出端口转发分组，该端口是输出端口 R3_3。在下一步骤 4 中，经由输出端口 R3_3，链路 119 和输入端口 D_1 发送分组到目的节点 D。目的节点 D 读取存储在分组 123 中的目的地址 DEST，并且当与它自己的地址比较时它确定它就是目的节点。万一网络编程在节点 D 失败，使用分布保存的返回路径从目的节点 D 返回分组 123 到源节点 S 来对网络重新编程。目的节点 D 根据输入端口 D_1 的标识符 D_1 和存储在目的节点 D 中的返回关系的组合确定使用输出端口 D_2 来发送分组 123，其中经由输入端口 D_1 接收该分组。在下一步骤 5 中，经由输出端口 D_2，链路 121 和输入端口 R3_4 发送分组 123 到节点 R3。节点 R3 从分组 123 中读取标识符 ID，并且校验这个标识符以一对 ID，R3_1 的形式存储在存储器 M3 中。节点 R3 根据输入端口 R3_1 的标识符 R3_1 和存储在节点 R3 中的返回关系的组合确定使用输出端口 R3_2 发送分组。其后，消除以标识符 ID 和标识符 R3_1 成对的形式存储在存储器 M3 中的关于返回路径的信息。在下一步骤 6 中，经由输出端口 R3_2，链路 117 和输入端口 R2_4 发送分组 123 到节点 R2。节点 R2 从分组 123 中读取标识符 ID，并且检测这个标识符以一对 ID，R2_1 的形式存储在存储器 M2 中。节点 R2 根据输入端口 R2_1 的标识符 R2_1 和存储在节点 R2 中的返回关系的组合确定使用输出端口 R2_2 发送分组 123。其后，消除以标识符 ID 和标识符 R2_1 成对的形式存储在存储器 M2 中的关于返回路径的信息。在下一步骤 7 中，经

由输出端口 R2_2, 链路 113 和输入端口 R1_4 发送分组 123 到节点 R1。节点 R1 从分组 123 中读取标识符 ID, 并且检测这个标识符以一对 ID, R1_1 的形式存储在存储器 M1 中。节点 R1 根据输入端口 R1_1 的标识符 R1_1 和存储在节点 R1 中的返回关系的组合确定使用输出端口 R1_2 发送分组。
5 其后, 消除以标识符 ID 和标识符 R1_1 成对的形式存储在存储器 M1 中的关于返回路径的信息。在下一步骤 8 中, 经由输出端口 R1_2, 链路 113 和输入端口 S_2 发送分组 123 到源节点 S。源节点 S 从分组 123 中读取标识符 ID 并在检测到该标识符 ID 没有存储在它的内部存储器之后确定它是分组 123 最后的目的地, 在图 1 中没有示出内部存储器。在这个实
10 施例中, 为了有效地实现“分组的标识符和输入端口的标识符”成对的存储, 存储器 M1, M2, M3 可以包含哈希表或可按内容寻址的存储器。存储器 M1, M2, M3 也可以包含关于除分组 123 之外其它分组的返回路径的信息, 没有在图 1 中示出。

从节点推导关于返回路径的信息, 该节点是当从目的节点 D 路由到源节点 S 时分组 123 访问的节点。该路径信息被分布保存在节点中并且当分组沿着返回路径传播时, 分组可以从一个节点传播到另一个时, 从每一节点推导关于返回路径的信息。因此, 不要求分组中的额外空间来存储关于返回路径的信息, 这允许减小分组的大小。
15

在其它实施例中, “分组的标识符和输出端口的标识符”成对存储在存储器 M1, M2, M3 中以确定分组 123 的返回路径。节点经由输入端口接收分组, 根据输入端口的标识符和存储在那个节点中的返回关系确定输出端口的标识符。例如, 在步骤 1 之后节点 R1 从分组 123 中读取标识符 ID 并结合标识符 R1_2 一起存储在存储器 M1 中, 作为一对 ID, R1_2。根据输入端口 R1_1 的标识符 R1_1 和存储在节点 R1 中的返回关系的组合确定标识符 R1_2。在步骤 7 中发送分组 123 到节点 R1 之后, 节点 R1 从分组 123 中读取标识符 ID 并校验这个标识符以一对 ID, R1_2 的形式存储在存储器 M1 中。使用输出端口 R1_2 的标识符 R1_2, 节点 R1 发送分组 123 经由输出端口 R1_2, 链路 109, 和输入端口 S_2 到达源节点 S。万一节点 R1, R2, R3 的输入端口数大于输出端口数, 在存储器 M1, M2, M3 中存储输出端口的标识符而不是输入端口的标识符以确定返回路径, 这要
20 求较小的存储空间。
25
30

在其它实施例中, 在到达目的节点 D 之前, 网络编程在某一节点上

可能失败。参考图 1，万一网络编程在节点 R3 失败，这个节点将把分组 123 路由到源节点 S。正如已经提到的，为了重新编程网络，分组沿返回路径传播到达源节点 S 是必需的。在这个实施例中网络的重新编程涉及取消在路径的那个点之前作的资源预定。在不同的实施例中，网络的重新编程也可以包括在沿返回路径传播期间发现到目的节点的可替换路径。节点 R3 根据输入端口 R3_1 的标识符 R3_1 和存储在节点 R3 中返回关系的组合确定使用输出端口 R3_2 发送分组，其中分组 123 是经由输入端口 R3_1 接收的。在下一步骤 6 中，经由输出端口 R3_2，链路 117 和输入端口 R2_4 发送分组到节点 R2。其后，把分组 123 路由到源节点 S，正如在先前实施例中所述。

在不同的实施例中，网络的重新编程在某一节点可能失败，例如因为某一资源正在被使用，访问该资源被拒绝。参考图 1，把分组路由到目的节点 D，但是节点 R3 的编程失败并且随后使用存储在节点 R1 和 R2 中的关于返回路径的信息把分组路由到源节点 S，正如在先前实施例中所述。网络的重新编程在节点 R2 失败，因为访问这个节点的资源被拒绝。节点 R2 读取存储在分组 123 中的目的地址 DEST，并且使用这个目的地址和目的地关系确定使用输出端口 R2_4 路由分组到目的节点 D。以 ID，R2_1 成对的形式存储在存储器 M2 中的关于返回路径的信息目前保留在存储器 M2 中。在下一步骤 3 中，经由输出端口 R2_3，链路 115 和输入端口 R3_1 发送分组到节点 R3。在节点 R3 中，作新的尝试以编程网络。如果这个尝试成功，发送分组 123 到目的节点 D，正如在先前实施例中所述。如果尝试失败，把分组路由到源节点 S，同样正如在先前实施例中所述。

在另一个实施例中，可以使用不同的方法推导分组的唯一标识符。例如，当使用时分复用仲裁方案时，为了确定在给定的时隙中路由器的那个输出端口连接到路由器的那个唯一的输入端口，使用时隙表用于路由器。因此，可以使用时隙来唯一标识分组和确定返回路径，正如下面所述。当从源节点传播到目的节点时，在分组中存储时隙，在该时隙期间节点发送分组，并且因为在两个邻近节点之间传播传播花费一个时隙，对于每一节点时隙值增加 1。万一分组不得不沿返回路径传播，它被发送到一个节点，例如经由输入端口 R2_4 发送到节点 R2。假定输入端口 R2_4 的返回关系是唯一的，通过把返回关系应用到输入端口 R2_4，输出端口

R₂₋₃ 被唯一标识。接下来，当分组从源节点传播到目的节点时，结合输出端口 R₂₋₃ 的标识符使用存储在分组中的时隙，经由它来接收分组的输入端口的标识符，也就是 R₂₋₁，可以从路由表中推导。接下来，使用返回关系和输入端口 R₂₋₁ 的标识符，可以确定输出端口 R₂₋₂ 的标识符，并且使用这个输出端口在源节点方向上发送分组，也就是说，沿返回路径传播。在发送分组之前存储在分组中的时隙值被降低 1。

图 2 展示了一个网络的实施例，当使用源路由从源节点发送分组到目的节点时，也就是说，该分组包含关于那个分组的路由的信息，该实施例使用根据本发明的确定网络中分组返回路径的方法。关于路由的信息可以以一系列后继节点输出端口的形式存储在分组中，以便每一节点从该分组检测使用哪个输出端口发送分组到下一节点。当把分组路由到目的节点时，在节点中存储关于返回路径的信息。参考 201，展示了包含节点 S₁, R₄, R₅ 和 D₁ 的一个网络，这些节点经由它们的输入端口和输出端口 S₁₋₁, S₁₋₂, R₄₋₁, R₄₋₂, R₄₋₃, R₄₋₄, R₅₋₁, R₅₋₂, R₅₋₃, R₅₋₄, D₁₋₁ 和 D₁₋₂ 用链路 207, 209, 211, 213, 215, 217 连接。网络 201 可以是一个网络，或是网络的一部分，或者集成电路的一部分。节点 S₁, R₄, R₅ 和 D₁ 可以包含用于发送数据单元到它的下一个目的地的路由器或交换机。节点 S₁, R₄, R₅ 和 D₁ 也可以包含用于耦合到其它节点的多个输入端口和输出端口，在图 2 中没有示出。从源节点 S₁ 发送分组 219 到目的节点 D₁。节点 R₄ 和 R₅ 分别包含存储器 M₄ 和 M₅。节点 S₁ 和 D₁ 同样包含一个存储器，在图 2 中没有示出。对于所有的节点 S₁, R₄, R₅ 和 D₁ 保持如果第一节点有至少一个到第二节点的链路，在第二节点和第一节点之间同样存在链路。节点 S₁, R₄, R₅ 和 D₁ 已经存储返回关系，使那个节点的每一输入端口与那个节点的一个输出端口相关联以便当在所述输入端口上接收来自特定节点的分组和经由所述输出端口发送分组时，它将被发送到那个特定节点。分组 219 被安排用来编程网络。如果在每一节点上网络编程是成功的，把分组路由到目的节点 D₁。然而，网络编程可能在某一节点失败，例如由于缺乏资源。在那种情况下为了重新编程到目前为止访问过的那部分网络，分组沿返回路径传播到源节点是必需的。在这个实施例中假定，一直到目的节点 D₁ 网络编程是成功的。参考 203，展示了当从源节点 S₁ 发送分组 219 到目的节点 D₁ 时分组 219 追随的路径。参考 205，展示了当从源节点 S₁ 发送分组到目的节点 D₁

时存储器 M4 和 M5 的内容。分组 219 包含标识符 ID，指针 P，输出端口标识符 A1 和 A2，计数器 C 和数据 DAT。标识符 ID 为分组 219 提供唯一标识。指针 P 指向在分组 219 内存储输出端口标识符的位置，该标识符是应当用来发送分组的输出端口的标识符。输出端口标识符 A1 和 A2 唯一地标识输出端口，分组应当经由该输出端口被发送。计数器 C 确定在到达目的节点 D 之前应当通过的节点的总数。使用数据 DAT 编程网络。
5 在其它实施例中，对源路由使用不同的编码方法是可能的，正如本领域内的技术人员所已知的那样。在从源节点 S1 发送分组 219 到目的节点 D1 之前，定义指针 P 以便它指向分组 219 中的输出端口标识符 A1 的位置。
10 设置输出端口标识符 A1 等于输出端口 R4_3 的输出端口标识符 R4_3，并且设置输出端口标识符 A2 等于输出端口 R5_3 的输出端口标识符 R5_3。计数器 C 被设置为 2。在第一步骤 1 中，分组 219 经过输出端口 S1_1，链路 207 和输入端口 R4_1 由源节点 S1 发送到节点 R4。由于为了发送分组 219 而选择正确的输出端口，源节点 S1 必须具有关于它连接到的网络的信息，例如以存储在节点 S1 中的目的地关系的形式。节点 R4 读取计数器 C 的值并因为计数器 C 的值不等于 0 而检测它不是目的节点。计数器 C 的值降低 1。节点 R4 从分组 219 中读出标识符 ID，并且结合来自输入端口 R4_1 的标识符 R4_1 把它存储在存储器 M4 中，作为一对 ID, R4_1。通过读取指针 P 的值和使用那个值读取输出端口标识符 A1，节点 R4 确定使用输出端口 R4_3 发送分组 219。节点 R4 更新指针 P 以便它指向在分组 219 中存储输出端口标识符 A2 的位置。在下一步骤 2 中，经由输出端口 R4_3，链路 211 和输入端口 R5_1 发送分组 219 到节点 R5。节点 R5 读计数器 C 并确定它不是目的节点，因为计数器 C 的值不等于 0。计数器 C 的值降低 1。节点 R5 从分组 219 中读取标识符 ID，并且结合来自输入端口 R5_1 的标识符 R5_1 存储在存储器 M5 中，作为一对 ID, R5_1。通过读取指针 P 的值和使用那个值读取输出端口标识符 A2，节点 R5 确定使用输出端口 R5_3 发送分组 219。节点 R5 确定指针 P 不用更新，因为计数器 C 的值等于 0。在下一步骤 3 中，经由输出端口 R5_3，链路 213 和输入端口 D1_1 发送分组 219 到节点 D1。节点 D1 读取计数器 C 的值并确定它是目的节点，因为计数器 C 的值等于 0。因此不用更新 C 的值并且不用读取指针 P 的值。万一网络编程在节点 D1 失败，使用分布保存的返回路径，从目的节点 D1 把分组 219 路由到源节点 S1 来重新编程网络。
15
20
25
30

使用输入端口 D1_1 的标识符 D1_1 和存储在节点 D1 的返回关系，节点 D1 确定使用输出端口 D1_2 把分组 219 发送回源节点 S1，其中分组 219 是经由输入端口接收。在下一步骤 4 中，经由输出端口 D1_2，链路 217 和输入端口 R5_4 把分组发送节点 R5。节点 R5 从分组 219 中读取标识符 ID，并检测这个标识符是以一对 ID, R5_1 的形式存储在存储器 M5 中。节点 R5 根据输入端口 R5_1 的输入端口标识符 R5_1 和存储在节点 R5 的返回关系确定使用输出端口 R5_2 发送分组 219。因为计数器 C 的值等于 0，节点 R5 确定指针 P 的值不用更新。随后，节点 R5 使计数器 C 加 1。消除以标识符 ID 和标识符 R5_1 成对的形式存储在存储器 M5 中的关于返回路径的信息。在下一步骤 5 中，经由输出端口 R5_2，链路 211 和输入端口 R4_4 发送该分组到节点 R4。节点 R4 从分组 219 中读取标识符 ID，并检测这个标识符以一对 ID, R4_1 的形式存储在存储器 M4 中。节点 R4 根据输入端口 R4_1 的输入端口标识符 R4_1 和存储在节点 R4 中的返回关系的组合确定使用输出端口 R4_2 发送分组 219。节点 R4 更新指针 P 以便它指向输出端口标识符 A1 被存储的位置，并且使计数器 C 加 1。消除以标识符 ID 和标识符 R4_1 成对的形式存储在存储器 M4 中的关于返回路径的信息。在下一步骤 6 中，经由输出端口 R4_2，链路 209 和输入端口 S1_2 发送该分组到节点 S1。源节点 S1 从分组 219 中读取标识符 ID，检测这个标识符没有存储在它的内部存储器中，该存储器没有在图 2 中示出，并确定它是目的节点。

参考图 2，在不同的实施例中，在到达目的节点 D1 之前网络编程可能在某一节点失败。参考图 2，万一网络编程在节点 R5 失败，这个节点将路由分组 219 到源节点 S1。正如已经提到的，为了重新编程网络，分组 219 沿返回路径传播到源节点 S1 是必需的。节点 R5 使用输入端口 R5_1 的输入端口标识符 R5_1 和存储在节点 R5 中的返回关系的组合确定使用输出端口 R5_2 把分组 219 路由到源节点 S1。因为计数器 C 的值等于 0，节点 R5 确定指针 P 的值不用更新。节点 R5 使计数器 C 加 1。在下一步骤 5 中，经由输出端口 R5_1，链路 211 和输入端口 R4_3 发送该分组到节点 R4。随后，正如在先前实施例中所述，进一步把分组 219 路由到源节点 S1。

参考图 2，在不同的实施例中，网络的重新编程可能失败，例如因为在特定的节点访问某一资源被拒绝。分组 219 被路由到目的节点 D1，

但是网络编程在节点 R5 失败并且随后使用存储在节点中的返回信息把分组 219 路由到源节点 S，正如在较早实施例中所述。把分组 219 发送到节点 R4。节点 R4 更新指针 P 以便它指向分组 219 中输出端口标识符 A1 被存储的位置，并且使计数器 C 的值加 1。接下来，网络的重新编程在 5 节点 R4 失败并且这个节点把该分组路由到目的节点 D1。节点 R4 通过读取指针 P 的值和使用那个值读取的输出端口标识符 A1 确定使用输出端口 R4_3 发送分组 219。节点 R4 更新指针 P 以便它指向输出端口标识符 A2 被存储的位置，并且使计数器 C 的值减 1。在下一步骤 2 中，经由输出 10 端口 R4_3，链路 211 和输入端口 R5_1 发送分组 219 到节点 R5，正如在较早实施例中所述。当关于返回路径的信息被存储在节点 R4 和 R5 中时，存储在分组 219 中用于从源节点 S1 路由该分组到目的节点 D1 的信息保留在分组 219 中。因此，分组 219 可以不止一次的经由相同的路径被路由到目的节点 D1，正如在这个实施例中所述的，并每次都尝试编程该网络。

15 再次参考图 1，在不同的实施例中，在从源节点 S 到目的节点 D 的路径上不是在所有分组 123 访问过的节点中存储关于返回路径的信息。如果部分返回路径是唯一的并等于该分组从源节点 S 传播到目的节点 D 的路径，没有返回信息必须被存储在与那部分返回路径相关的节点中。例如，在一个实施例中，节点 R2 仅有两个输入端口 R2_1 和 R2_4，和 20 两个输出端口 R2_3 和 R2_2，当从源节点 S 发送分组 123 到目的节点 D 时，没有关于返回路径的信息被存储在节点 R2 的存储器 M2 中。万一网络编程在节点 R3 失败，分组 123 被路由到源节点 S，正如在先前实施例中所述。节点 R3 发送分组 123 经由输出端口 R3_2，链路 117 和输入端口 R2_4 到达节点 R2。因为可以从它的目的地关系确定，节点 R2 仅仅使 25 用输出端口 R2_2 把分组 123 路由到源节点 S，并且发送分组 123 经由输出端口 R2_2，链路 113 和输入端口 R1_4 到达节点 R1。随后，节点 R1 把分组 123 发送源节点 S，正如在先前实施例中所述。在这个实施例中，假定当节点 R2 已经从节点 R3 接收到该分组并且在节点 R2 中的网络重新编程是成功时，不允许节点 R2 把该分组发送回节点 R3。

30 应该注意到上述实施例是说明而不是限制本发明，并且本领域内的技术人员在不脱离附加权利要求范围的情况下能够设计许多可以替换的实施例。在权利要求书中，在括弧内放置的任何参考标记都不应该被解

释为限制权利要求。词“包含”不排除除了权利要求书中所列的那些之外的单元或步骤的出现。在单元之前的词“一”或“一个”不排除多个这样单元的出现。在列举了几个装置的设备权利要求中，这些装置中的几个可以由同一项硬件实现。仅仅在相互不同的从属权利要求中描述某一措施这个事实不表示不可以有利地使用这些措施的组合。
5

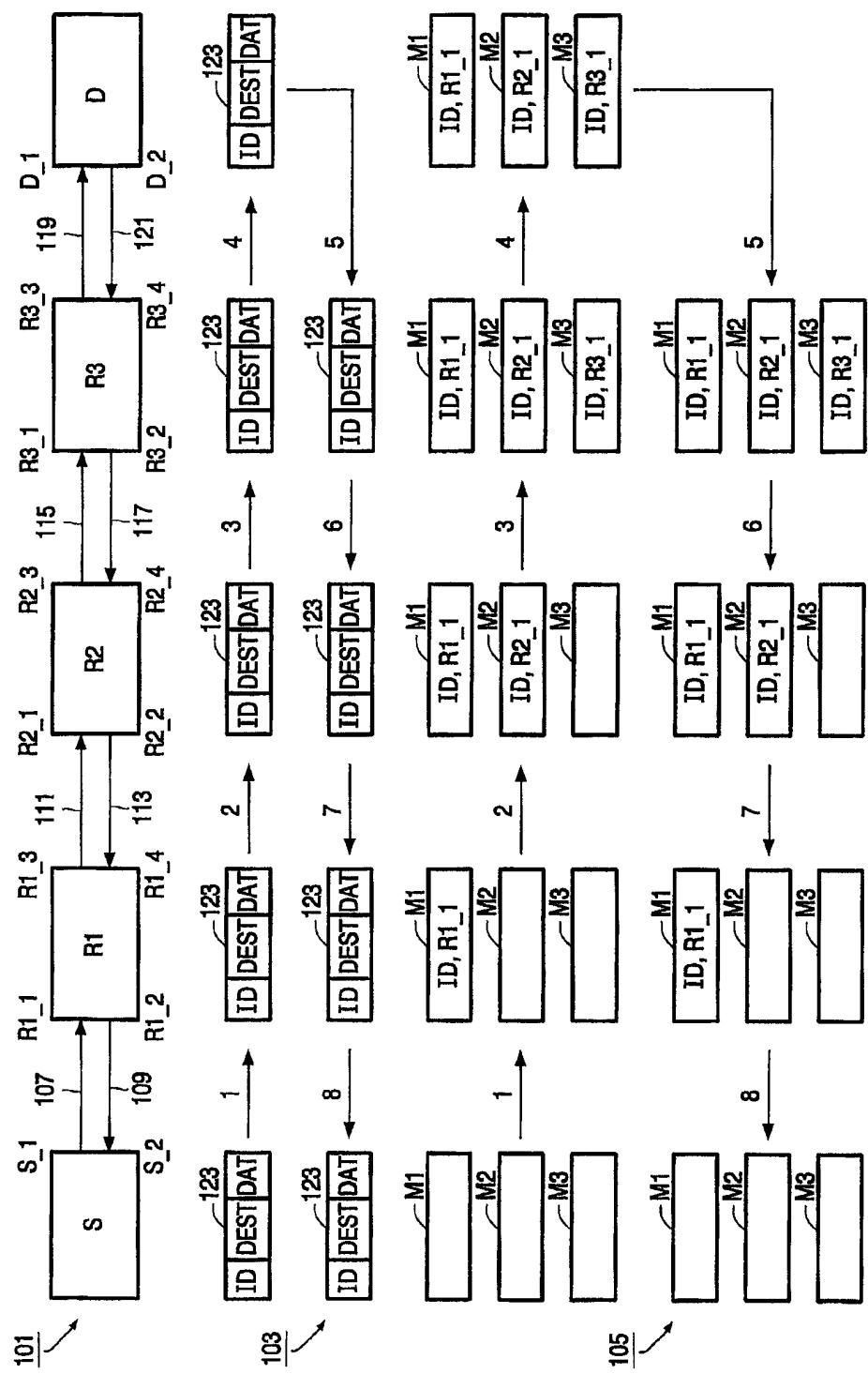


图 1

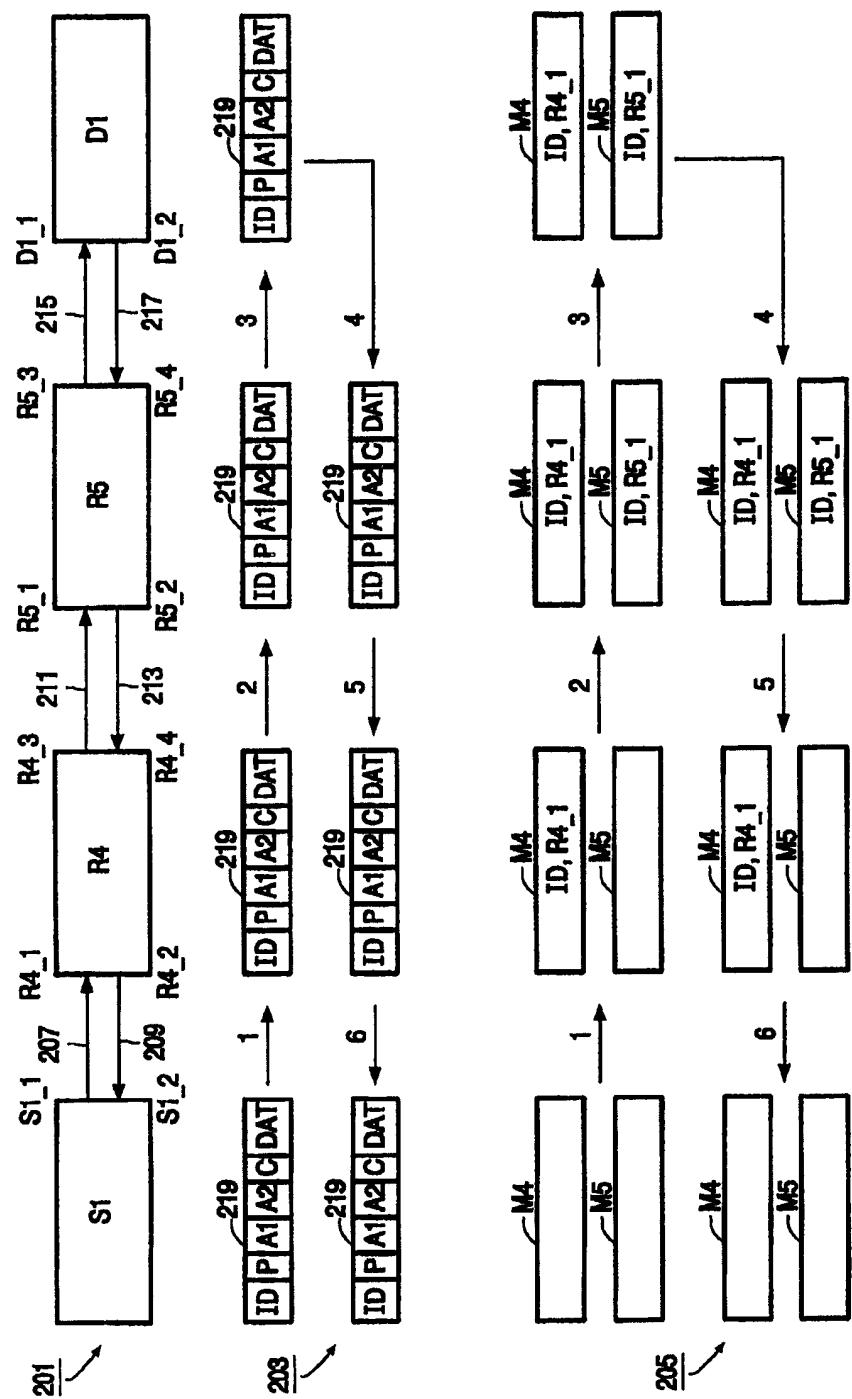


图 2