(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 1820537 B
(45)授权公告日 2010.11.03

(21)申请号 200480017784.3

(22)申请日 2004.06.22

(30)优先权数据
MI2003A001309 2003.06.26 IT

(85)PCT申请进入国家阶段日
2005.12.26

(86)PCT申请的申请数据
PCT/EP2004/051191 2004.06.22

(87)PCT申请的公布数据
WO2005/002276 EN 2005.01.06

(73)专利权人 爱立信股份有限公司
地址 瑞典斯德哥尔摩

(72)发明人 S·普雷特贾尼

(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 吴立明 刘杰

(51)Int.Cl.
H04Q 11/04(2006.01)

(56)对比文件
CN 1349334 A,2002.05.15, 全文.
CN 1159723 A,1997.09.17, 说明书第3页第14-19行,第7页第6-10行,第8页第21-22行,第13页第26-30行,附图2.
EP 1001648 A2,2000.05.17, 全文.
CN 1326282 A,2001.12.12, 全文.
CN 1240564 A, 全文.
CN 1042286 A,1990.05.16, 全文.
US 5412646 A,1995.05.02, 全文.
CN 1056968 A,1991.12.11, 全文.

审查员 韩雪

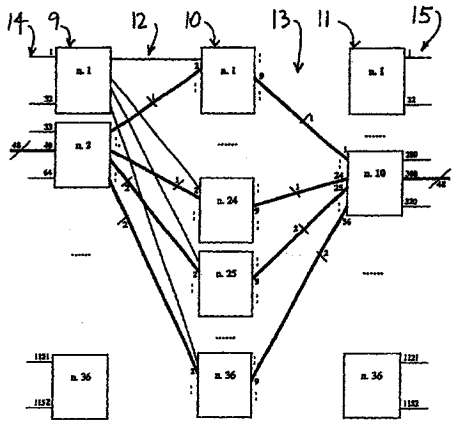
权利要求书 2 页 说明书 7 页 附图 10 页

(54)发明名称

交换网络

(57)摘要

一种构成一部分数字交叉连接的交换网络，用于在所请求的输入端和输出端之间连接数据帧，并且包括输入级9、中间级10和输出级11，每一个都由多个交换矩阵组成。每一个输入级交换矩阵具有一条到每一中间级交换矩阵的链路，通常由参考标记12指示，并且每一个中间级交换矩阵具有一条到每一输出级交换矩阵的链路，通常由参考标记13指示。根据本发明，通过交换矩阵独立地路由输入数据帧的各个时隙，从而减少了发生阻塞的可能性。



CN 1820537 B

1. 一种用于在定义的时隙内，在期望的输入端和期望的输出端之间交换交叉连接请求的数据帧的交换网络，包含由多个交换矩阵组成的输入级、由多个交换矩阵组成的中间级、和由多个交换矩阵组成的输出级，每一个输入级交换矩阵具有一条到每一个中间级交换矩阵的链路，并且每一个中间级交换矩阵具有一条到每一个输出级交换矩阵的链路，所述交换网络包括用于通过交换矩阵独立地路由每一帧的每一时隙的装置。

2. 一种如权利要求 1 所述的交换网络，其中路由装置被设置成通过中间级交换矩阵路由交叉连接请求帧的每一后续时隙，该中间级交换矩阵的链路具有最大瞬时数量的空闲时隙。

3. 如在权利要求 2 中所述的交换网络，其中所述链路在中间级交换矩阵与所请求的输出端的输出级交换矩阵之间。

4. 如在权利要求 3 中所述的交换矩阵，其中路由装置被设置成，如果在所述链路中多个中间级交换矩阵具有与输出级交换矩阵相等瞬时数量的空闲时隙，那么通过中间级交换矩阵路由每一后续时隙，该中间级交换矩阵具有起自接收所述请求的输入级交换矩阵的链路，并且该链路具有最大瞬时数量的空闲时隙。

5. 一种权利要求 1 至 4 任何一个中所述的交换矩阵，其中路由装置被设置成，当接收到交叉连接请求时，在中间级交换矩阵与该请求的输入以及输出级交换矩阵之间的链路中创建空闲时隙的列表。

6. 如在权利要求 1 至 4 任何一个中所述的交换矩阵，其中该链路是时分复用链路。

7. 如在权利要求 1 至 4 任何一个中所述的交换矩阵，其中中间级交换矩阵的数量小于每一输入级交换矩阵输入端数量的两倍。

8. 如在权利要求 1 至 4 任何一个中所述的交换矩阵，其中路由装置被设置成在输出级交换矩阵的输出端上将各自路由的时隙重组为帧。

9. 如在权利要求 1 至 4 任何一个中所述的交换矩阵，其中交换装置适合于接收标准数据业务，根据此标准数据业务创建了取代至少一些开销的内部生成的帧。

10. 如在权利要求 1 至 4 任何一个中所述的交换矩阵，其中所述链路的每一时隙对应的帧具有的数据速率至少为该链路数据速率的 1/16。

11. 如在权利要求 10 中所述的交换矩阵，其中该时隙的每一个都可以接纳 SDH 标准的 AU-3 帧。

12. 一种数字交叉连接，包括如在权利要求 1 至 11 任何一个中所述的交换矩阵。

13. 一种在定义的时隙内通过交换网络路由数据帧，以便完成在期望的输入端和期望的输出端之间的交叉连接请求的方法，其中该交换网络包含由多个交换矩阵组成的输入级，由多个交换矩阵组成的中间级、和由多个交换矩阵组成的输出级，每一个输入级交换矩阵具有一条到每一个中间级交换矩阵的链路，并且每一个中间级交换矩阵具有一条到每一个输出级交换矩阵的链路，本方法包括通过交换矩阵独立地路由每一帧的每一时隙的步骤。

14. 如在权利要求 13 中所述的方法，其中通过中间级交换矩阵路由交叉连接请求帧的每一后续时隙，该中间级交换矩阵的链路具有最大瞬时数量的空闲时隙。

15. 如在权利要求 14 中所述的方法，其中该链路在中间级交换矩阵与所请求的输出端的输出级交换矩阵之间。

16. 如在权利要求 15 中所述的方法，其中如果在所述链路中多个中间级交换矩阵具有与输出级交换矩阵相等瞬时数量的空闲时隙，那么通过中间级交换矩阵路由每一后续时隙，该中间级交换矩阵具有起自接收所述请求的输入级交换矩阵的链路，并且该链路具有最大瞬时数量的空闲时隙。

17. 如在权利要求 13 至 16 任何一个中所述的方法，其中当接收到交叉连接请求时，在中间级交换矩阵与该请求的输入以及输出级交换矩阵之间的链路中创建空闲时隙的列表。

交换网络

[0001] 本发明涉及一种用于在输入端和输出端之间交换信号的交换网络。

[0002] 本发明尤其涉及构成一部分数字交叉连接的交换网络。参考图1,这样的数字交叉连接通常包含一系列输入模块A、一系列输出模块B和链接它们的交换网络C。输入模块在各个输入端上接收以诸如同步数字层次结构(SDH)、同步光网络(SONET)、或者光传输网络(OTN)格式之类格式的数据流并处理那些数据流(例如,检测和删除维护和信令信息),以便创建传送到交换网络C的内部帧。交换网络C随后将内部帧路由到合适的输出端,并且输出模块B通过增加维护和信令信息来建立诸如SDH、SDNET或OTN之类的输出数据流。

[0003] 矩阵控制器单元D构成交换网络C的一部分并且负责控制交换行为和实现路由算法。

[0004] 用于将n个输入中任何一个交换到m个输出中任何一个的已知网络是多级网络,其使用至少一个中间级来在不增加阻塞的情况下减少交叉点的数量,并且在1953年由Charles Clos出版了技术论文之后被称为Clos网络。

[0005] 通常已知的Clos网络如图2所示。第一级1具有“a”个交换矩阵,其每一个具有“b”个输入端和“c”个输出端。中间级2具有“c”个交换矩阵,其每一个具有“a”个输入端和“a”个输出端,并且输出级3具有“a”个交换矩阵,其每一个具有“c”个输入端和“b”个输出端。

[0006] 每一个第一级1矩阵的输出端仅通过一条链路连接到所有中间级2矩阵的输入端,并且每一中间级2矩阵的输出端仅通过一条链路被连接到所有输出级3矩阵的输入端。

[0007] 分数c/b被称为扩展因子,并且与矩阵内部路径的数量有关并且因此具有发现从交换网络的输入端到输出端的路由的可能性。如果数“b“等于或超过数“a”减1的两倍,其近似为大于或等于2的扩展因子,那么根据Clos交换网络的理论,通过始终发现连接任何一个输入与任何一个输出的路径来完全填充网络的输出端是可能的。

[0008] 本发明涉及Clos交换网络,其中扩展因子小于2(它必须大于1,否者阻塞位置就可能存在于单个交换矩阵的内部),以便通过增加输入端和输出端的数量来提供增加的容量,而不过度的增加成本。然而,这增加了未能发现通过网络的路径的风险,并且必须使用复杂业务状态重排算法。

[0009] 参考图3说明了扩展比率小于2时阻塞如何发生的一个例子,其展示了每级具有9个矩阵的三级Clos网络,尽管在输入和输出级仅示出了一个矩阵。输入级4上的每一个矩阵具有8个输入端和9个输出端,中间级5上的每一个矩阵具有9个输入端和9个输出端,输出级6上的每一个矩阵具有9个输入端和8个输出端。扩展比率因此是9/8.

[0010] 考虑试图建立在输入级4的第一矩阵输入端1和输出级6的第一矩阵输出端1之间的交叉连接。假定后者的输出端2至8通过来自中间级5的矩阵3至9的连接而被使用和馈送。因此只经由中间级5的矩阵1和2来仅仅从第一级4的第一矩阵建立交叉连接。然而,如果由前进到除第一矩阵之外的任何一个输出级矩阵8的两个连接来使用输入级4第一矩阵的输出端1和2(其连接到中间级的前两个矩阵),则不能建立想要的交叉连接。

[0011] 本发明涉及在由各种数字标准定义的不同业务级上来交换数据帧,各种数字标准

诸如是 SDH、SONET 或者 OTN 标准。参考图 4，说明了基本 SDH 复用结构 (ITU 建议 -G.70X)。在 AU(管理单元)-3 和 AUG 数据帧之间的链路上的 x3 指示 AUG 数据帧包含三个 AU-3 数据帧。当 N ＝ 1 时，STM(同步传输模块)-N 帧可以包含一个 AUG 帧，也就是说，当 N ＝ 16 时包含 16 个 AUG 帧。通常以每秒 8000 帧来发送数据帧，因此较大帧的数据速率也相应的较大。STM-16 数据帧的数据速率近似等于 2.5Gbit/sec。

[0012] 在图 2 中所示的 Clos 网络的输入端、输出端和内部链路因此可以接纳预定数量的时隙，例如对应于 2.5Gbit/sec 的数据速率可以接纳 48 个时隙。每一时隙总是用填塞信息或有效数据来填充，在后一种情况中，数据是从特定输入端到交换网络的数据。时隙因此可以属于任何类型的业务。例如，它可以是 STM-1 的第三个，STM-4 的第 12 个等，或者可以不被使用，也就是说，用由交换网络自身产生的填塞信息来填充。

[0013] 对于这样的 Clos 网络来说已知的是，除了它的空间路由性能之外在它的内部链路中还允许数据的时分复用 (TDM)。参考图 5，设定在图 2 网络的任何一个特定交换矩阵的输入端“x”上的特定单元 7，交换矩阵 4 可以将它移动到输出端“y”(空间路由) 并且在“y”流的内部移动到一个新的时间位置 (时间复用)。

[0014] 申请人检查了在交换数字数据的 Clos 网络中的典型阻塞位置，并且注意到在装满内部链路的数据中的大量碎片。例如，进入交换网络的一些连接请求是低数据速率的，例如 STM-1，在该情况下，在上述例子中 48 个时隙中的三个将会被有效数据装满，并且 45 个将会被填塞信息填充，链路很空闲但或许对于 STM-16 来说却仍然不能处理另一请求。

[0015] 本发明提供了一种用于在定义的时隙中在期望的输入端和期望的输出端之间交换交叉连接请求的数据帧的交换网络，包含由多个交换矩阵组成的输入级、由多个交换矩阵组成的中间级、和由多个交换矩阵组成的输出级，每一个输入级交换矩阵具有一条到每一个中间级交换矩阵的链路，并且每一个中间级交换矩阵具有一条到每一个输出级交换矩阵的链路，所述交换网络包括用于通过交换矩阵独立地路由每一帧的每一时隙的装置。

[0016] 这允许更好地利用内部链路并相应地减少阻塞。

[0017] 有益地，扩展比率 (输入级交换矩阵的输出比输入) 小于 2，当先前已经倾向于 Clos 阻塞时优选地小于 1.25。

[0018] 有益地，交换矩阵具有时分复用链路。每一时隙可以接纳一个其额定数据速率至少是链路数据速率 1/16 的帧，优选地至少是链路数据速率的 1/48。

[0019] 现在将参考附图通过实例详细描述根据本发明构建的用于交换数据帧的交换矩阵，其中：

[0020] 图 1 是表示既是公知的又是根据本发明的包含交换网络的数字交叉连接的框图；

[0021] 图 2 是既是公知又是根据本发明的三级 Clos 交换网络的框图；

[0022] 图 3 是表示阻塞情况的部分已知 Clos 交换网络的特定实例的框图；

[0023] 图 4 是图示 STM SDH 数据帧结构的示意图；

[0024] 图 5 是描述图 2 的 Clos 交换网络的交换矩阵的交换操作的框图；

[0025] 图 6 是描述对根据本发明的三级 Clos 交换网络进行内部填充的一个实例的框图；

[0026] 图 7A 至 7I 是表示对到中间级交换矩阵的链路进行填充的示意图，和图 7J 是表示对到输入级交换矩阵的链路进行填充的示意图；

[0027] 图 8 是描述对图 6 的三级 Clos 网络进行内部填充方法的流程图；

[0028] 图 9 是对根据本发明的三级 Clos 网络进行内部填充的另一个实例的框图；

[0029] 图 10 是表示对进入并留在根据本发明的三级 Clos 交换网络的数据帧进行分割和重组的框图；

[0030] 图 11 是表示对根据本发明的三级 Clos 网络进行内部填充的另一个实例的框图；和

[0031] 图 12 是描述重排算法的流程图。

[0032] 参考图 1，本发明的交换网络构成数字交叉连接的一部分，并且与传统交叉连接一样，包括一系列输入模块 A 和输出模块 B，其分别删除和恢复维护信息和信令信息以便创建通过交换网络 C 而路由的内部帧。输入模块 A 通过用交换网络 C 所使用的信息或填塞信息替换输入 SDH、SONET 或 OTN 帧的一些字节来产生内部帧。删除的字节承载信令信息。输出模块 B 实现了反向的处理：它们用属于 SDH、SONET 或 OTN 标准的信息来替换那些内部使用的内部帧字节。

[0033] 本发明的交换网络还包括矩阵控制器单元 D，以便控制交换行为并实现根据发明的路由算法。

[0034] 参考图 6，将依照根据 SDH 标准接收输入业务的三级 Clos 网络来描述本发明的交换网络，其中在输入级 9、中间级 10 和输入级 11 的每一个中有 36 个交换矩阵。在输入级中的那些交换矩阵具有 32 个输入端和 36 个输出端，在中间级中的那些交换矩阵具有 36 个输入端和 36 个输出端，并且在输出级中的那些交换矩阵具有 36 个输入端和 32 个输出端。与传统的 Clos 网络一样，每一个输入级 9 的交换矩阵具有一个并且仅仅一个通常由参考标记 12 指示的到每一个中间级 10 交换矩阵的链路，并且每一个中间级 10 的交换矩阵具有一个并且仅仅一个通常由参考标记 13 指示的到每一输出级 11 交换矩阵的链路。到交换网络输入端和输出端的链路通常由参考标记 14、15 来指示。

[0035] 交换网络是如此以至于在 2.5Gbit/sec 的物理链路上业务从任何一个输入级交换矩阵传播到任何一个中间级交换矩阵，并从任何一个中间级交换矩阵传播到任何一个输出级交换矩阵。在链路 12、13、14、15 上的信号是由 48 个时隙组成的时分复用信号。每一个时隙具有 2.5Gbit/sec/48 的比特速率。这个速率是 AU-3 类型信号的比特速率。每一个时隙都被由交换网络产生的填塞信号或者有效信号填充，其可以是整个 AU-3 帧、三分之一的 STM-1 帧、十二分之一的 STM-4 帧、或者四十八分之一的 STM-16 帧。

[0036] 到交换网络的输入是以根据 SDH 标准在不同业务级上的数据帧的形式。每一个输入链路可以接纳在比特速率上等于或低于 STM-16 的所有数据帧，也就是说，VC3、VC4、STM-1、STM-4、STM-16。在多条输入链路上接纳较高比特速率的数据帧。例如，将 STM-64 安插在四条输入链路上。

[0037] 因为输入级交换矩阵 9 的扩展比率是 9/8，明显小于大约为 2 的 Clos 未阻塞扩展比率，因此如在图 1 和 2 中的公知交换网络一样，如果以整个帧来路由帧，那么交换网络对于这样的业务来说会很容易阻塞。

[0038] 根据本发明，矩阵控制器单元 D 包括用于通过交换矩阵独立地路由每一帧的每一时隙的装置。因此，参考图 8，当用户发出交叉连接请求时，该请求被分割为由各个时隙组成的 AU-3 较低数据流，每一个时隙包含一个 AU-3 帧。

[0039] 本发明的矩阵控制器单元 D 执行的下一步是创建中间级交换矩阵的列表，显示出链路的占用，所述中间级交换矩阵首先从发出请求的输入交换矩阵开始，然后到达包含所请求的输出的特定输出交换矩阵。首先，基于在中间级交换矩阵和所请求的特定输出级交换矩阵之间的链路 13 的占用情况来分类列表，与具有有效数据的最多空闲时隙的链路相对应的矩阵是第一个矩阵。如果存在多于一个具有同等数量空闲时隙的链路，那么就接着基于在到输入级交换矩阵的链路 12 中的空闲时隙数量来分类各个交换矩阵，其中在所述输入级交换矩阵上接收到了用户交叉连接请求。如果存在多于一个具有与在链路 12 中相同数量的空闲时隙和与在链路 13 中相同数量的空闲时隙的中间级交换矩阵，例如，当所有的时隙都没有有效数据时，那么基于在图 6 所示的数字次序或者按着一些其它预定次序来列出交换矩阵。

[0040] 在请求单向连接的情况下，对于交叉连接请求的第一 AU-3 时隙来说，在列表中的第一中间级交换矩阵的链路 12、13 中分配时隙。随后为交叉连接请求的下一个 AU-3 时隙分配该列表中下一个中间级交换矩阵的链路 12、13 中的时隙，以此类推。使用同一算法来分别处理每一时隙。

[0041] 现在参考图 6 来给出一个特定例子。假定不存在穿过交换网络的交叉连接，如此，所有链路 12、13 都具有 48 个被填塞信号填充的时隙。进一步假定需要用于 STM-16 数据帧的从第 40 个输入端到第 300 个输出端的单向连接。这意味着必需将在第 2 输入级交换矩阵上的第 8 个输入端与第 10 输出级交换矩阵上的第 12 个输出端互连。

[0042] 处理 STM-16 连接请求好像它是 48 个独立的 AU-3 单向连接请求一样。参考图 7A，36 个列与中间级交换矩阵相关。第一行 (Clos) 是中间级交换矩阵的数量，第二行是在特定中间级交换矩阵和所要求的输出级交换矩阵之间的链路的 AU-3 填充，并且第三行是在发起请求的输入级交换矩阵和特定中间级交换矩阵之间的链路的 AU-3 填充。AU-3 填充可以是在 0 到 48 之间的任意数。该算法为任何一个所要求的连接创建这样一个列表。

[0043] 图 7A 中示出了在请求之前的情况。在图 7B 中示出了在已经路由了第一个 AU-3 之后的情况。在图 7C 中示出了已经路由了第二个 AU-3 之后的情况。正如在图 7D 中所示，在已经路由了 36 个 AU-3 之后，所有的中间级交换矩阵 (Clos) 再次具有同样的优先权。在已经路由了最后的 AU-3 之后，顺序列表正如在图 7E 中所示。

[0044] 图 6 中示出了对中间级交换矩阵进行内部填充的一种表示。任何一个不涉及第 2 输入级交换矩阵和第 10 输出级交换矩阵的请求将以同样的方式被路由。

[0045] 现在考虑从第 700 个输入端的第 5 个 VC(虚拟容器)-4 帧到第 307 个输出端的第 2 个 VC-4 的单向连接，其中输出级交换矩阵与先前请求的交换矩阵相同。从交换网络的视点来看，这意味着第 22 个输入级交换矩阵的第 28 个输入端的 13、14 和 15 AU-3 时隙必须连接到第 10 个输出级交换矩阵的第 19 个输出端的 4、5 和 6 AU-3 时隙。该算法将处理这个类似于 3 个独立 AU-3 单向连接的 VC-4 连接。

[0046] 在图 7F 中示出了在请求之前的情况。这个列表考虑了 (参见行 2“输出“) 由第一连接引起的链路 13 的填充。在图 7G 中示出了在最后的 AU-3 之后的顺序列表。在图 9 中示出了这个新的内部填充。

[0047] 内部链路 12、13 是具有 48 个对数据可用的时隙的 TDM 链路。各自路由的 AU-3 将通常被馈送到相应链路中的第一空闲时隙，而与在 48 个时隙中的位置无关。因此，一个特

定的 AU-3 可以被容纳在一条链路的第一个 TDM 时隙中。下一个 AU-3 可以被容纳在所使用的下一条链路的最后一个 TDM 时隙中，并且第三个 AU-3 可以被容纳在 48 个时隙的行的末端中间的时隙中。重要的是在各个输出级交换矩阵的输出端上按顺序收集 AU-3。

[0048] 参考图 10，考虑 VC4 帧的路由（3 个时隙等同于三个 AU-3 字节“a，b，c”的重复序列）。矩阵控制器单元 D 以该图中所示的方式来配置输入级、中间级和输出级 Clos 块。与其它用户请求对应的繁忙时隙被表示为“x”。在输入级上分割输入字节“a，b，c”的序列，以便字节“a”进入中间级交换矩阵 i，字节“b”进入中间级交换矩阵 j，等等。使用每一个交换矩阵路将在链路的任意时隙上输入的字节路由到任意输出链路的任意时隙的能力。在链路 12 的时隙 2 内以及在链路 13 的时隙 47 中承载字节“a”，并且对于字节“b”和“c”也以此类推。各个输出级交换矩阵级 11 通过以“a，b，c”字节的正确顺序排列它们来再次映射正确的 VC4 结构。

[0049] 在上面描述的本发明交换网络的关键特征中的关键点是：时分复用（这使得通过交换矩阵硬件成为可能）；和将任何 SONET、SDH 或 OTN 业务分割为 AU-3 数据流级的可能性。

[0050] 可以把到交换矩阵的输入业务看作是一组 AU-3 数据帧，而不管到数字交叉连接的真实输入业务级（例如来自 SDH 设备的 STM-1、STM-4、STM-16 等和来自 SONET 设备的 STS-1、STS-3 等）。

[0051] 以平衡中间级交换矩阵填充的独立方式来路由所有的 AU3 时隙，首先朝向来自中间级的输出端，并且其次朝向到中间级的输入端。这减少或消除了由于在输入 / 输出端达到饱和之前链路 12、13 中没有一个被完全填充（即不是所有的时隙都被填充）所产生的片断。

[0052] 设定为满足用户需求而要到达的特定输出级交换矩阵，该算法通过平衡在中间级交换矩阵和特定输出级交换矩阵之间的链路的填充（用时隙号 0 到 48 来测量）来选择所使用的中间级交换矩阵。返回到参考图 9 给出的上述实例，在已经安插了第二请求之后，在 VC4 到 STM2 56 的任何一个业务级上的一个后续的请求 --VC4 需要到达 10 号输出级交换矩阵，因此不使用 1、2、3、25、26、27、28、29，...，36 号中间级交换矩阵，因为它们到 10 号输出级交换矩阵的链路的填充等于 2，并且由于它们到 10 号输出级交换矩阵的链路的填充等于 1，因此将首先考虑用 4、5、6，...，24 号中间级交换矩阵。

[0053] 反过来参考图 8，用不同于单向连接（也就是说，从一个输入端到一个输出端的连接）请求的方式来处理广播连接（也就是说，从一个输入端到多于一个输出端的连接）的请求。在后一情况下，如果在适当位置已经有穿过交换网络的连接，这些连接与使用同一输出级交换矩阵的广播请求相关，那么仅在已经承载广播请求的输出级交换矩阵中路由连接。在这种输出级交换矩阵上的输入端被连接到两个或更多个输出端而不是仅仅一个。

[0054] 如果在适当位置不存在使用同一输出级交换矩阵的连接，该算法寻找已用于路由同一广播的先前支路的中间级交换矩阵。如果存在这样的中间级交换矩阵，该算法基于上面提到的优先权列表再次通过将该输入端连接到多个输出端的那些矩阵来在它们之间进行选择。如果没有这种正在使用的中间级交换矩阵，使用满足下述请求的优先权列表的第一路径，即是：链路 12（输入端到中间级）和 13（中间级到输出端）是空闲的。

[0055] 举个例子，假定在图 9 中所示的 VC-4（第 700 个输入端的第 5 个 VC-4）也需要连接到第 1137 个输出端的第 14 个 VC-4，将单向连接改变为广播连接。这在图 11 中被图示。

这个新的支路不是在同一个输出级交换矩阵上（它是在第 36 级而不是第 10 级上），因此不能使用同一输出交换矩阵。跟前面一样，这次第 22 个输入级交换矩阵的第 28 个输入端的 13、14 和 15 AU-3 不得不被连接到第 36 个输出级交换矩阵的第 27 个输出端的 40、41 和 42AU-3。照例该算法将处理类似于 3 个独立 AU-3 连接的这个 VC-4。

[0056] 在图 7H 中示出了在该请求要求另外连接之前的情况。应该注意的是，对每一个 AU-3 来说，由于来自第 700 个输入端的第 5 个 VC-4（中间级交换矩阵 1-3）的先前支路而已经存在从输入端到中间级的路径，使得仅仅通过规划来自那些交换矩阵的其它链路 13 来开通在中间级上的第二支路。当路由最后的 AU-3 时，顺序列表在图 7I 中示出。注意到，如果另外请求的输出流是在第 10 个输出级交换矩阵上（在 289-320 范围内的输出端），该算法将会在第 10 个输出级交换矩阵上创建广播连接。

[0057] 上述路由算法的计算机仿真已经显示没有出现阻塞的情况，因此不需要重排。相信这就是将数据业务分割为 AU-3 结构的原因。

[0058] 然而，设想在硬件出错的情况下将会发生阻塞，例如中间级交换矩阵出错。该算法因此包括正好考虑了这种情况的重排策略。

[0059] 重排是一种在保持输入 / 输出状态的同时改变内部矩阵占有情况的方法。目的之一是产生比旧内部状态更适合容纳新路径的新内部状态；另一个目的就是重新分配避免中间级交换矩阵次序混乱的内部路径。因此，重排是一种根据某些策略提供请求的智能序列。注意到，通常由用户发起的提供应当是提供请求的随机序列。

[0060] 使用之前所述的路由方法来模拟所有的许多不同的重排策略，并且所有的都能起作用。为了区别它们，考虑输入级交换矩阵的输入端 32 到输入端 36。虽然重排策略失败，但是所选择的一个策略在解决阻塞状况数量方面显示出了较好的性能。

[0061] 参考图 12，重排算法工作如下。

[0062] 首先每一个 n- 支路广播连接（仅仅对于在输入端或中间级上开通的广播连接）被认为是一系列的 n- 单向连接。随后考虑每一个交叉连接的每一个 AU-3 单元。

[0063] 现在确定在哪个输入级交换矩阵开始重排是必需的。为每一个输入级交换矩阵计算正在使用的输出 AU-3 的总数。以降序来计算根据该总数所排列的这种矩阵的列表。如果存在多于一个具有同一值的输入级交换矩阵，通过考虑正在使用的输入 AU-3 单元的总数以降序来排列它们。

[0064] 提取该列表的第一单元，从最低的输入端开始重新重新（使用正常的路由算法）所有单向连接（这个输入级交换矩阵的所有单向连接）。从最低的输入端开始重新路由（使用修改的路由算法）所有广播连接（这个输入级交换矩阵的所有广播连接）。这个修改的路由算法首先选择可以接受最高数量广播支路的中间级交换矩阵。如果存在许多这样的交换矩阵，通过考虑正在使用最低数量的 AU-3 的一个来区别它们。重复这个操作直到所有的广播支路已经被安插了。

[0065] 现在给出一个特定的重排实例。假定输入 / 输出端状态如下：

[0066] 从第 40 个输入端到第 300 个输出端的 STM-16 单向连接；从第 700 个输入端的第 5 个 VC-4 到第 307 个输出端的第 2 个 VC-4 以及到第 1137 个输出端的第 14 个 VC-4 的 VC-4 广播连接（正如在图 11 中所示）；另外，从第 390 个输入端的第 9 个 VC-4 到第 50 个输出端的第 2 个 VC-4 的 VC-4 单向连接。

[0067] 如图 7J 所示，显示了输入级交换矩阵输出和输入链路 AU-3 填充的列表。首先考虑 2 号交换矩阵，并且将它的单向连接重新路由（没有广播连接）。接下来要考虑的输入级交换矩阵是 22 号。没有单向连接，因此重新路由广播连接的两个支路。最后一个输入级交换矩阵是路由单向连接的 13 号。

[0068] 在不脱离本发明范围的情况下可以做出各种变形。因此，代替在输入和输出级之间只有单个中间级，而可多于一个，例如在输入级和输出级之间可以包括三个级。

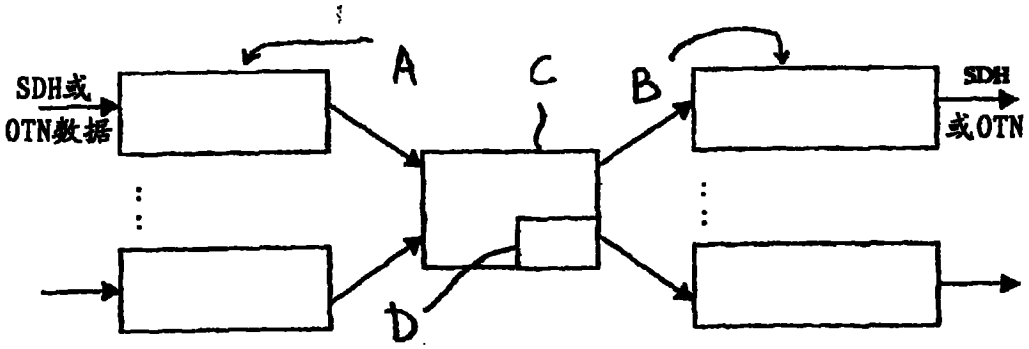


图 1

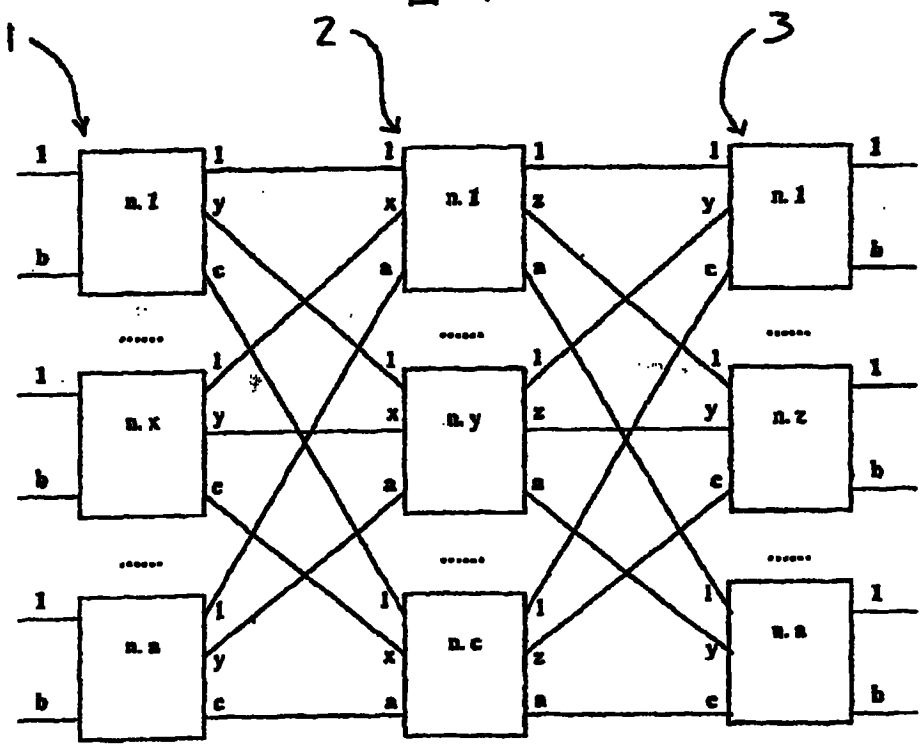


图 2

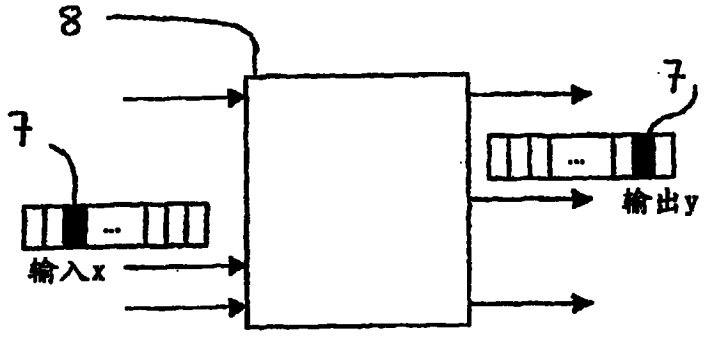


图 5

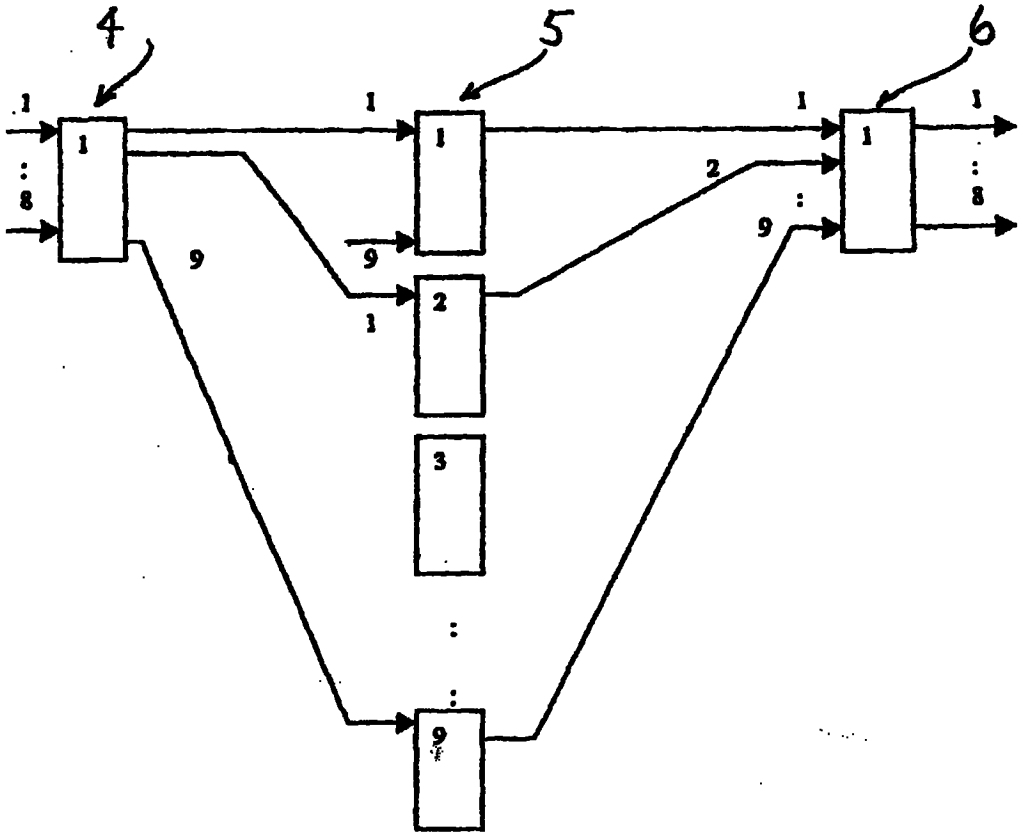


图 3

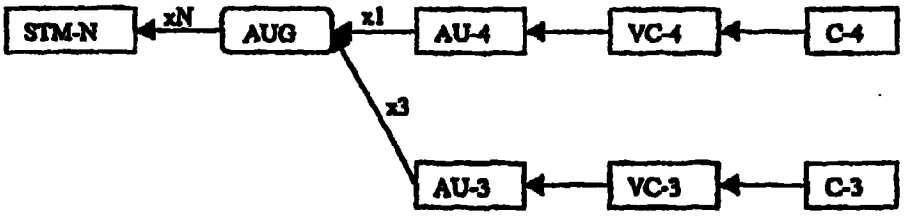


图 4

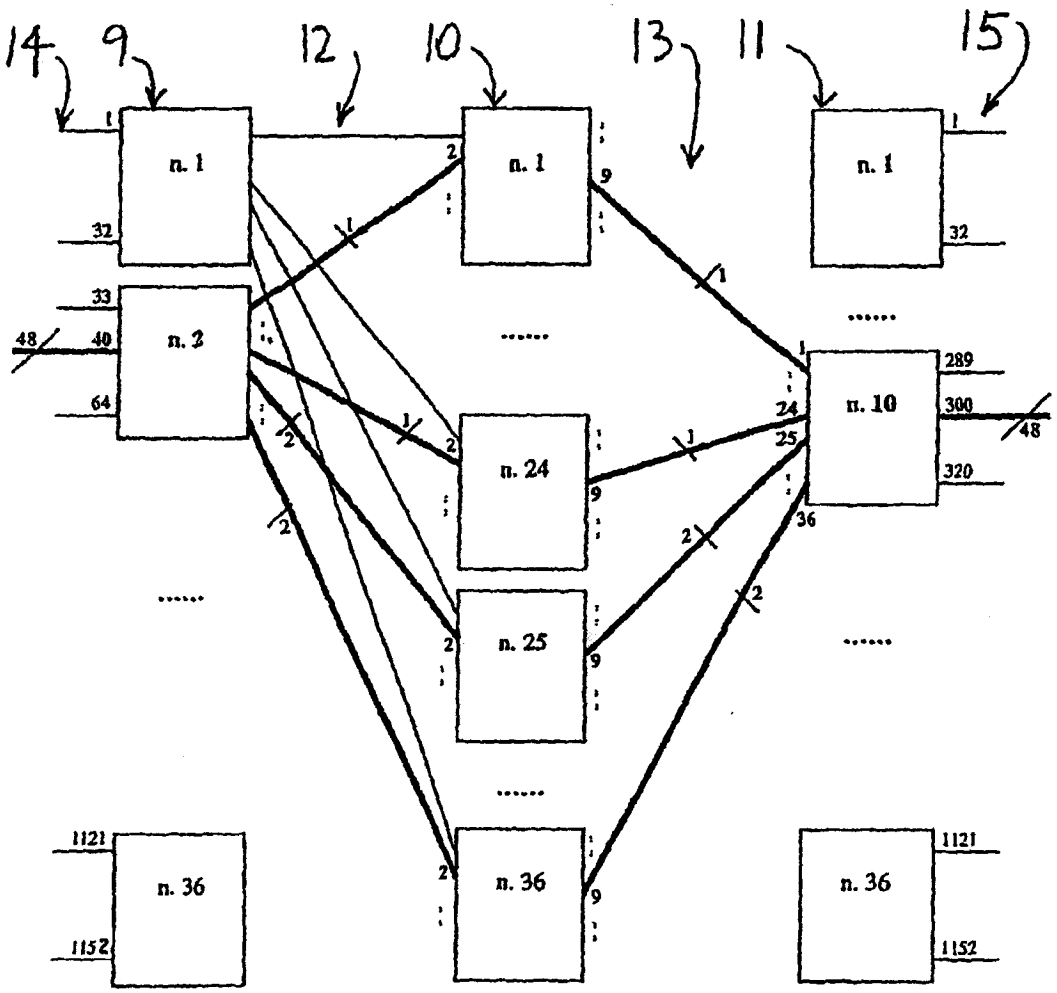

14
9
12
10
13
11
15
n. 1
n. 1
n. 1
n. 2
n. 24
n. 25
n. 36
n. 36
n. 10
n. 36
1
32
33
40
48
64
2
9
24
25
36
289
300
320
1121
1152

图 6

Clos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
输出	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
输入	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

图 7A

Clos	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	1
输出	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
输入	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

图 7B

Clos	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	2	1
输出	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
输入	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

图 7C

Clos	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
输出	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
输入	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

图 7D

Clos	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	25	26	27	28	29	30	31	32	33	34	35	36
输出	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2
输入	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2

图 7E

Clos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
输出	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2
输入	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

图 7F

Clos	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	3	2	1
输出	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
输入	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

图 7G

Clos	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	3	2	1
输出	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
输入	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

图 7H

Clos	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	3	2	1
输出	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
输入	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

图 7I

Clos	2	22	13	1	3	4	5	6	7	8	9	10	11	12	14	15	16	17	18	19	20	21	23	24	25	26	27	28	29	30	31	32	33	34	35	36
输出	48	6	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
输入	48	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

图 7J

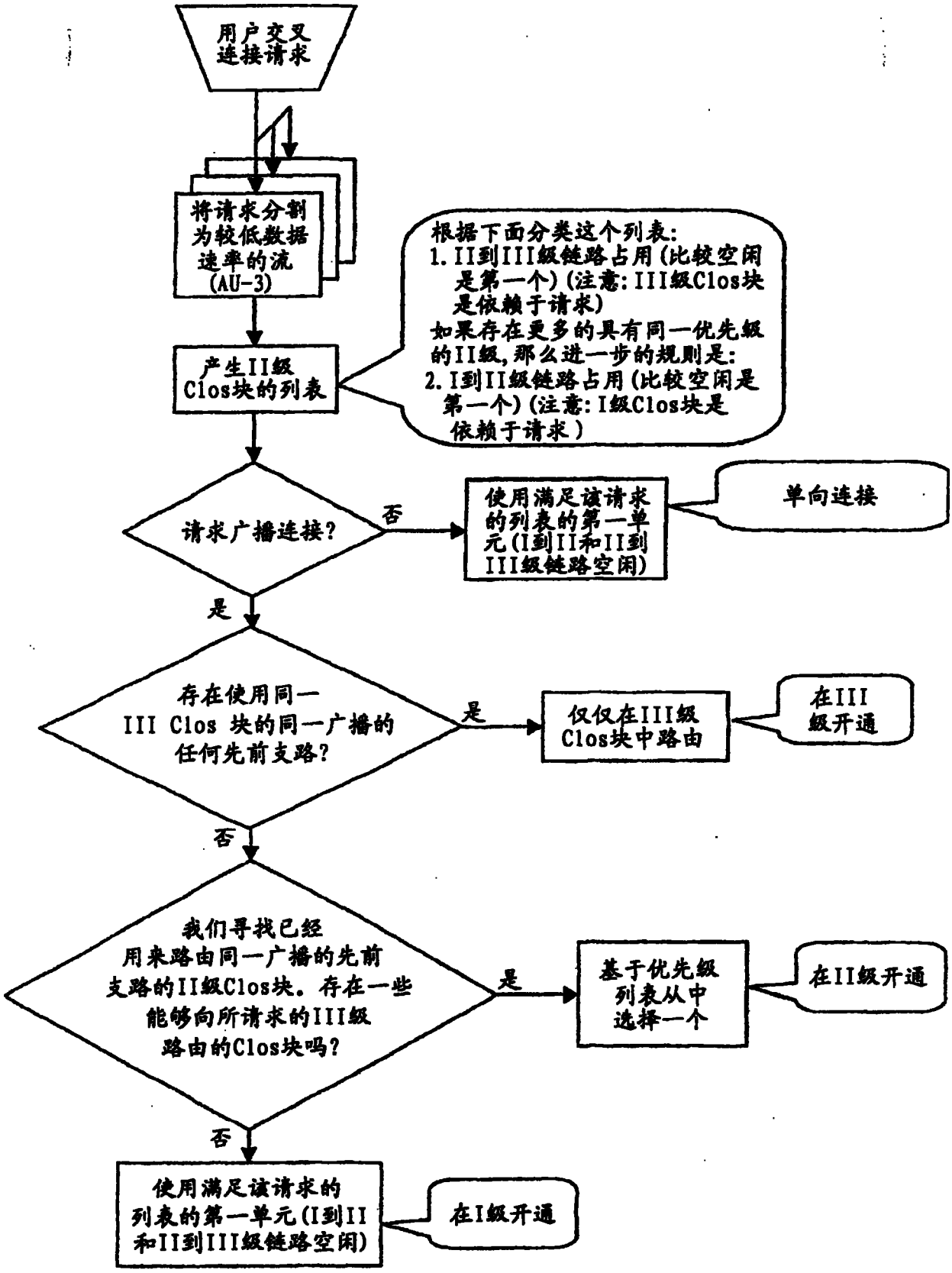

用户交叉连接请求
将请求分割为较低数据速率的流(AU-3)
产生II级Clos块的列表
根据下面分类这个列表:
1. II到III级链路占用(比较空闲是第一个)(注意:III级Clos块是依赖于请求)
如果存在更多的具有同一优先级的II级,那么进一步的规则是:
2. I到II级链路占用(比较空闲是第一个)(注意:I级Clos块是依赖于请求)
请求广播连接?
否
使用满足该请求的列表的第一单元(I到II和II到III级链路空闲)
单向连接
是
存在使用同一III Clos 块的同一广播的任何先前支路?
是
仅仅在III级Clos块中路由
在III级开通
否
我们寻找已经用来路由同一广播的先前支路的II级Clos块。存在一些能够向所请求的III级路由的Clos块吗?
是
基于优先级列表从中选择一个
在II级开通
否
使用满足该请求的列表的第一单元(I到II和II到III级链路空闲)
在I级开通

图 8

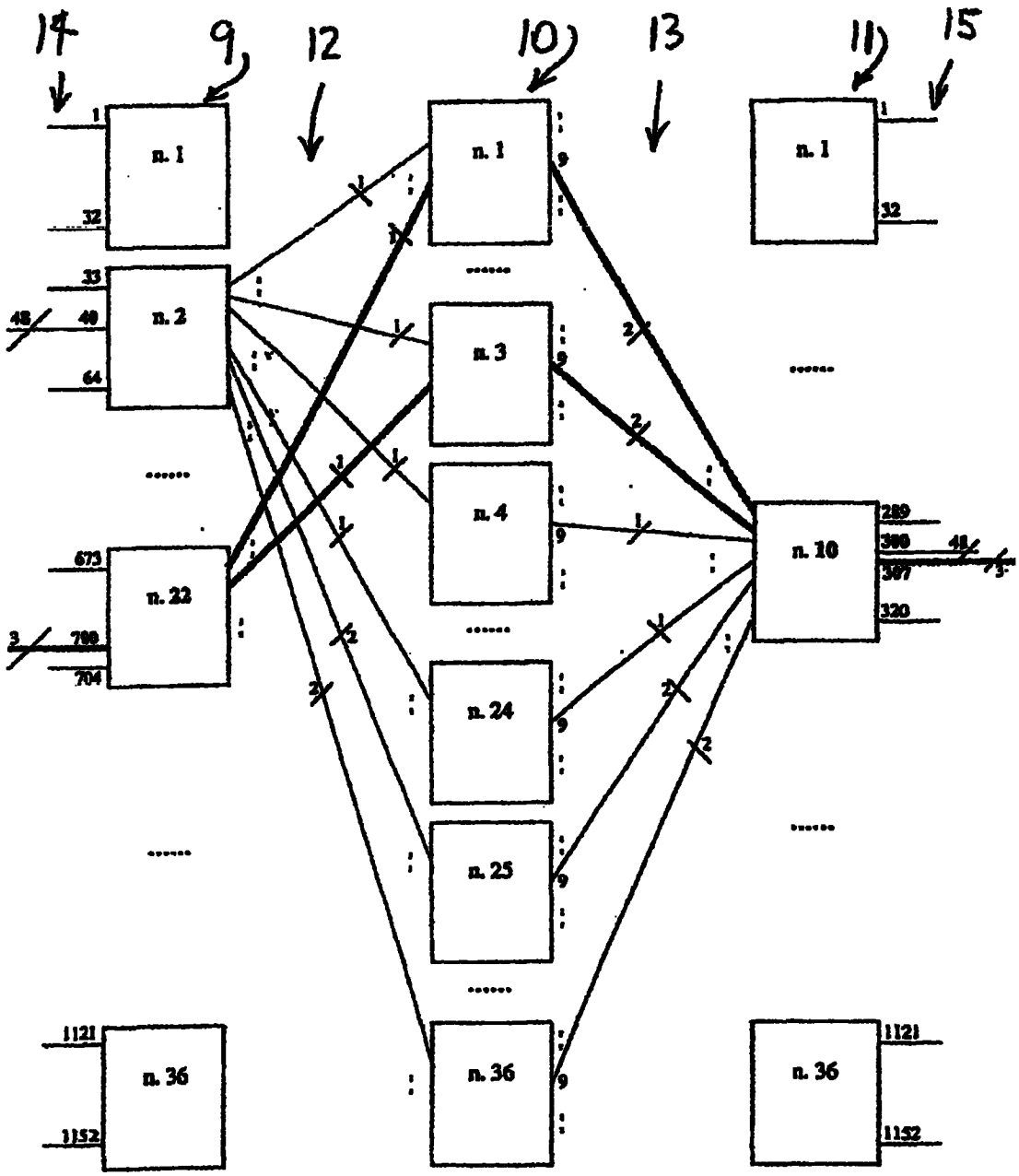

14
9
12
10
13
11
15
n. 1
n. 2
n. 22
n. 1
n. 3
n. 4
n. 24
n. 25
n. 36
n. 1
n. 10
n. 36
1
32
33
48
49
64
673
3
708
704
1121
1152
289
288
48
3
287
320

图 9

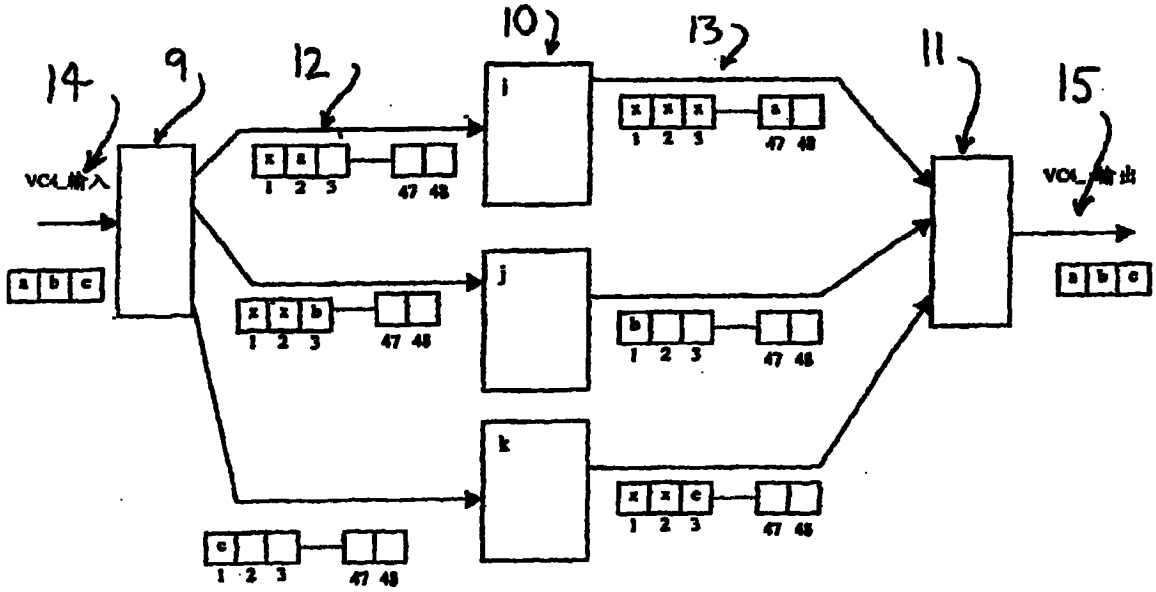


图 10

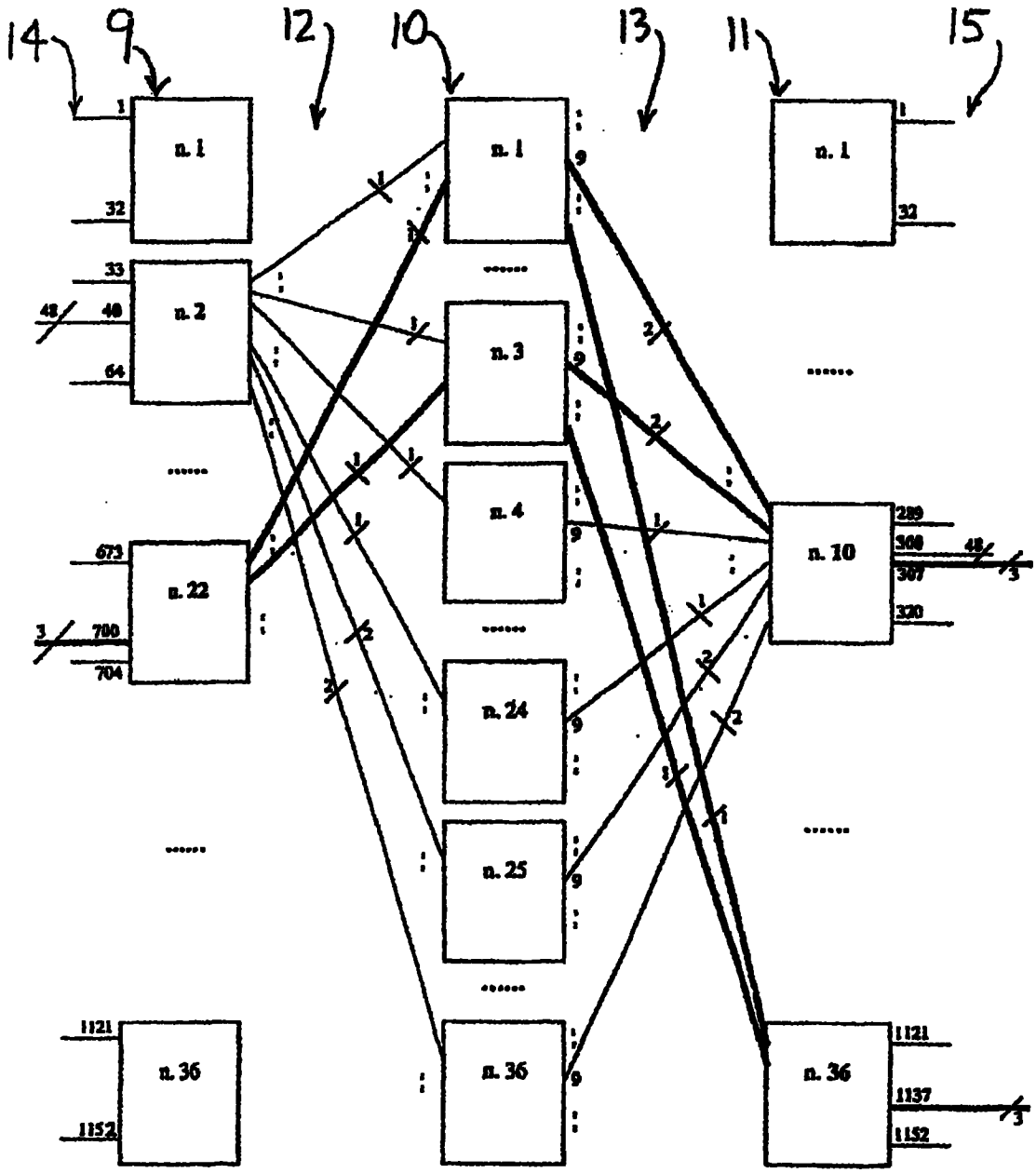


图 11

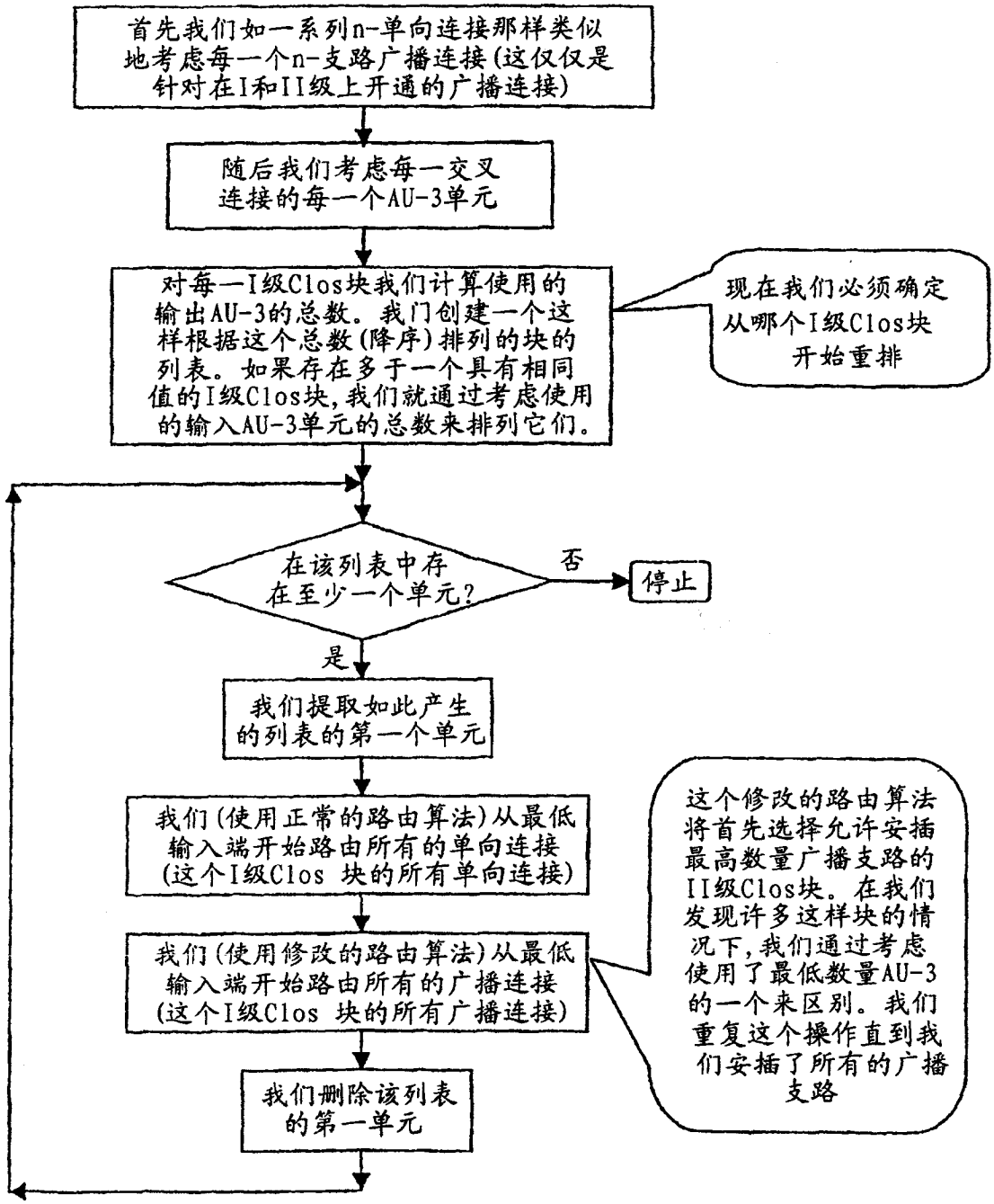

首先我们如一系列n-单向连接那样类似地考虑每一个n-支路广播连接(这仅仅是针对在I和II级上开通的广播连接)
随后我们考虑每一交叉连接的每一个AU-3单元
对每一I级Clos块我们计算使用的输出AU-3的总数。我门创建一个这样根据这个总数(降序)排列的块的列表。如果存在多于一个具有相同值的I级Clos块,我们就通过考虑使用的输入AU-3单元的总数来排列它们。
现在我们必须确定从哪个I级Clos块开始重排
在该列表中存在至少一个单元?
否
停止
是
我们提取如此产生的列表的第一个单元
我们(使用正常的路由算法)从最低输入端开始路由所有的单向连接(这个I级Clos 块的所有单向连接)
我们(使用修改的路由算法)从最低输入端开始路由所有的广播连接(这个I级Clos 块的所有广播连接)
这个修改的路由算法将首先选择允许安插最高数量广播支路的II级Clos块。在我们发现许多这样块的情况下,我们通过考虑使用了最低数量AU-3的一个来区别。我们重复这个操作直到我们安插了所有的广播支路
我们删除该列表的第一单元

图 12