



(12) 发明专利申请

(10) 申请公布号 CN 104904143 A

(43) 申请公布日 2015. 09. 09

(21) 申请号 201380069142. 7

(51) Int. Cl.

(22) 申请日 2013. 11. 26

H04B 10/80(2006. 01)

(30) 优先权数据

H04B 10/27(2006. 01)

61/729, 872 2012. 11. 26 US

H04Q 3/52(2006. 01)

(85) PCT国际申请进入国家阶段日

2015. 07. 07

(86) PCT国际申请的申请数据

PCT/CA2013/000976 2013. 11. 26

(87) PCT国际申请的公布数据

W02014/078940 EN 2014. 05. 30

(71) 申请人 维斯柯科技有限公司

地址 加拿大安大略省卡纳达

(72) 发明人 刘云曲 梁金伟 倪文达 黄昌成

(74) 专利代理机构 深圳市博锐专利事务所

44275

代理人 张明

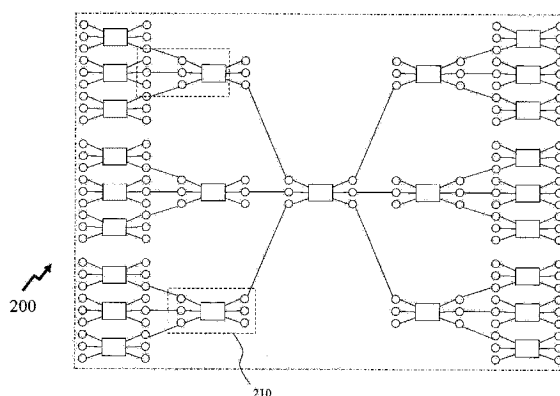
权利要求书3页 说明书19页 附图12页

(54) 发明名称

用于无源光交换的方法和系统

(57) 摘要

光网络变得无处不在,其可提供低成本、高速通信网络,为通过长城网络到海底的 FTTH 提供通信支持。提供给每根用户光纤的大量用户和高速意味着托管信息的数据中心内的信息检索和路由功能可成为速度和时延的瓶颈。根据本发明的实施例,发明人提出了基于全光无源光网络的架构,该架构支持用于减小时延的分配方式以及与架构的部署相关的协议。有利的是,这样的 POCXN 概念所采用的光学部件已经得到大批量制造技术以及用于增加吞吐量的 CWDM/DWDM 技术的支持。



1. 一种设备,包括:

多个分配级,每个分配级包括预定数量的具有输入端口和输出端口的无源光分路器,所述无源光分路器将所述输入端口中的每一个处的光信号同时耦合到所有所述输出端口;以及

多个互联级,每个互联级将所述多个互联级的第一预定级连接至所述多个互联级的第二预定级,以使得所述光信号的来自所述多个互联级的所述第一预定级的第一预定数量的无源光分路器的输出端的预定部分以预定的方式被耦合至所述多个互联级的所述第二预定级的第二预定数量的无源光分路器的输入端的预定部分。

2. 根据权利要求 1 所述的设备,其中,

所述设备包括至少一个具有三个输入端口和三个输出端口的无源光分路器。

3. 根据权利要求 1 所述的设备,其中,

所述设备包括预定数量的具有三个输入端口和三个输出端口的第一光分路器以及预定数量的具有两个输入端口和两个输出端口的第二光分路器。

4. 一种方法,包括:

提供一种无源光交叉连接网络 (POXN),该 POXN 包括第一组多个输入端口和第二组多个输出端口;

提供耦合至所述 POXN 的多个节点以使得所述多个节点中的一个节点发送的信号在不存在任何交换功能的情况下被自动分配到所述多个节点;

提供在每个节点上执行的协议引擎,所述协议引擎执行单个节点发现序列和至少一个数据转移序列的周期性重复处理;

其中,在所述至少一个数据转移序列期间每个节点所发送的数据依靠在所述节点发现序列期间请求发送数据的节点的数量以及每个请求发送的节点必须发送的数据量来进行确定。

5. 根据权利要求 4 所述的方法,其中,

提供所述 POXN,包括:

提供多个分配级,每个分配级包括预定数量的具有输入端口和输出端口的无源光分路器,所述无源光分路器将所述输入端口中的每一个处的光信号同时耦合到所有所述输出端口;以及

提供多个互联级,每个互联级将所述多个互联级的第一预定级连接至所述多个互联级的第二预定级,以使得所述光信号的来自所述多个互联级的所述第一预定级的第一预定数量的无源光分路器的输出端的预定部分以预定的方式被耦合至所述多个互联级的所述第二预定级的第二预定数量的无源光分路器的输入端的预定部分。

6. 根据权利要求 5 所述的方法,其中,

所述 POXN 包括至少一个具有三个输入端口和三个输出端口的无源光分路器。

7. 根据权利要求 5 所述的设备,其中,

所述 POXN 包括预定数量的具有三个输入端口和三个输出端口的第一光分路器以及预定数量的具有两个输入端口和两个输出端口的第二光分路器。

8. 根据权利要求 4 所述的方法,其中,

接收数据的时延在预定量内是恒定的,接收数据在所述 POXN 的输入端口处由所述多

个节点的第一节点发送至所述 POXN 的所有输出端口。

9. 一种方法,包括:

提供一种无源光交叉连接网络 (POXN), 该 POXN 包括第一组多个输入端口和第二组多个输出端口;

提供耦合至所述 POXN 的多个发射器节点以使得所述多个发射器节点的发射器节点发送的信号在不存在任何交换功能的情况下被自动分配到多个接收器节点, 每个接收器节点在预定波长上进行发射;

提供耦合至所述 POXN 的所述多个接收器节点以使得所述多个发射器节点发送的信号在不存在任何交换功能的情况下被自动分配到所述多个接收器节点, 每个接收器节点从所述多个发射器节点的所有正在发射的发射器节点接收所述信号;

提供多个电子处理元件, 所述多个电子处理元件从所述多个发射器节点接收多个接收信号, 每个电子处理元件从所述多个接收信号的每个接收信号中滤波掉不用于所述多个接收器节点中一个接收器节点的全部内容, 该接收器节点与该电子处理元件相关联; 以及

提供多个缓冲节点, 每个缓冲节点从所述多个发射器节点接收经滤波的接收信号并且将所述经滤波的接收信号排序用于传输到被连接至该缓冲节点的数据宿。

10. 根据权利要求 9 所述的方法, 还包括:

提供在所述多个发射器节点和所述多个接收器节点的每一个上执行的协议引擎, 每个协议引擎执行单个节点发现序列和至少一个数据转移序列的周期性重复处理;

其中, 在所述至少一个数据转移序列期间每个发射器节点发送的数据依靠在所述节点发现序列期间请求发送数据的节点的数量以及每个请求发送的节点必须发送的数据量来进行确定。

11. 一种方法, 包括:

提供第一无源光交叉连接网络 (POXN), 该第一 POXN 包括第一组多个第一输入端口和第二组多个第一输出端口;

提供第二无源光交叉连接网络 (POXN), 该第二 POXN 包括第一组多个第二输入端口和第二组多个第二输出端口;

提供多个数据源, 每个数据源被耦合至所述第一组多个第一输入端口的预定第一输入端、所述第一组多个第一输出端口的预定第一输出端、所述第二组多个第二输入端口的预定第二输入端、所述第二组多个第二输出端口的预定第二输出端。

12. 根据权利要求 11 所述的方法, 其中,

至少一个 POXN, 还包括:

提供耦合至所述 POXN 的所述多个接收器节点以使得所述多个发射器节点发送的信号在不存在任何交换功能的情况下被自动分配到所述多个接收器节点, 每个接收器节点从所述多个发射器节点的所有正在发射的发射器节点接收所述信号;

提供多个电子处理元件, 所述多个电子处理元件从所述多个发射器节点接收多个接收信号, 每个电子处理元件从所述多个接收信号的每个接收信号中滤波掉不用于所述多个接收器节点的一个接收器节点的全部内容, 该接收器节点与该电子处理元件相关联; 以及

提供多个缓冲节点, 每个缓冲节点从所述多个发射器节点接收经滤波的接收信号并且将所述经滤波的接收信号排序用于传输到被连接至该缓冲节点的数据宿。

13. 一种方法,包括:

提供多个无源光交叉连接网络 (POXN),每个 POXN 包括多个输入端口和多个输出端口;
以及

提供多个数据源,每个数据源被耦合至所述多个 POXN 的每个 POXN 的预定输入端以及被耦合至所述多个 POXN 的每个 POXN 的预定输出端。

14. 根据权利要求 13 所述的方法,其中,

至少一个 POXN,还包括:

提供耦合至所述 POXN 的所述多个接收器节点以使得所述多个发射器节点发送的信号在不存在任何交换功能的情况下被自动分配到所述多个接收器节点,每个接收器节点从所述多个发射器节点的所有正在发射的发射器节点接收所述信号;

提供多个电子处理元件,所述多个电子处理元件从所述多个发射器节点接收多个接收信号,每个电子处理元件从所述多个接收信号的每个接收信号中滤波掉不用于所述多个接收器节点的一个接收器节点的全部内容,该接收器节点与该电子处理元件相关联;以及

提供多个缓冲节点,每个缓冲节点从所述多个发射器节点接收经滤波的接收信号并且将所述经滤波的接收信号排序用于传输到被连接至该缓冲节点的数据宿。

用于无源光交换的方法和系统

相关申请的交叉引用

[0001] 本专利申请要求于 2012 年 11 月 26 日递交的题为“用于无源光交换的方法和系统”的美国临时专利申请 61/729,872 的权利,该专利申请的全部内容通过引用而包括在本申请中。

技术领域

[0002] 本发明涉及光网络,尤其涉及一种用于减小时延的无源光交叉连接网络。

背景技术

[0003] 近年来,我们见证了由不断增长的带宽需求所驱动的光接入和局域网的快速发展以及竞争技术的根本乏力,例如基于双绞铜线、同轴电缆或无线传输来进行传送的技术的乏力。因此,光纤传输已经成为用于通信的通用方式,这些通信涉及从到达城域的长程网络到宽带接入网络的各种通信,从而导致促使光因特网激增,以及不同媒体流(例如数据、语音和视频)汇集为光域中被递送到最终用户的因特网协议数据。光接入光纤提供了一种解决“最后一英里”瓶颈的经得住未来考验的解决方法,该解决方案不仅显著增加了网络容量还消除了从光域到电域的昂贵转变(反之亦然)。在局域网、城域网和广域网(LAN、MAN 和 WAN)中,光纤亦可提供可扩展的网络,所述可扩展的网络在 2.5Gbs 和 10Gb/s(OC-48/OC-192)时分复用的基础上采用与具有(例如)4、8、16、32、48 或 64 个波长的长程网络相同的波分复用(WDM)技术,尽管可通过离散波长进行扩展。

[0004] 这样的网络采用多种架构,包括但不限于点到点、线性(亦称作总线)、环形、星形、树形和网格以及单向流量(每光纤)(多根光线)和双向流量(每光纤)以及 TDM、粗波分复用(CWDM)和密波分复用(DWDM)技术。线性网络通常在长程网络中使用,网格网络以及相关的全连接网络也是如此。环形网通常出现在 LAN、MAN 和 WAN 中,而树形网用于光纤到户/路边/驻地(FTTH/FTTC/FTTP)(通常统称为 FTTx)。对网络拓扑进行的这些逻辑分类描述了数据在节点之间所行进的路径,该路径与节点之间的实际物理连接相对。逻辑拓扑通常与媒体接入控制(MAC)方法与协议紧密相关,在大多数情况下,MAC 方法和协议也可通过采用路由器和/或交换机来动态地重新配置。

[0005] 因此,这些光网络形成了一个网络结构,可向当今客户提供高速因特网接入、点播音视频内容分配、无限简单消息服务(SMS)等,从而使得用于通信、娱乐以及商业应用的无线和有线电子设备变得几乎无处不在。因此,用于存储内容的术语,比如“服务器集群”和“数据中心”,成为了表示成百上千的计算机服务器和存储驱动器(硬盘驱动器)被共置——以支持因特网上的数据存储和分配——的位置的代名词的一部分。内部采用光网络的“服务器集群”和“数据中心”是现在被称为企业光网络的一部分,除其他元件外,企业光网络还采用旨在满足应用程序和非通用计算应用需求的网络附加存储(NAS)和存储区域网络(SAN)。NAS 设备无非是专用于文件共享的服务器,并且在不关掉服务器以进行维护和升级的情况下使得更多的硬盘存储空间能够被增加到已经使用这些服务器的网络上。利用

NAS 设备,存储器不再是服务器的一个集成部分,而是由服务器对数据进行所有处理,这些数据由一个或多个 NAS 设备递送给服务器以及用户。NAS 设备不必放置在服务器内,而是可存在于网络(通常为 LAN)中的任何位置,并且 NAS 设备自身可由多个联网的 NAS 设备组成。SAN 是共享存储设备的高速子网络,其中每个存储设备仅包含用于存储数据的硬盘。理想情况下, SAN 的架构应使得所有存储设备对 SAN 所连接的网络上的所有服务器均可用。

[0006] 对于多于 1 亿的网站上估计超过 1000 亿的网页,数据中心包含海量数据。对于接入所有这些网站的大约 20 亿用户,连同不断增长的高带宽视频,容易认识到但难以理解的是,因特网每分钟的上传和下载数据量。目前,用户之间的全球 IP 业务量的年复合增长率(CAGR)介于 40%(基于 Cisco 的分析,参见 http://www.cisco.com/en/US/solutions/col_lateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html)与 50%(基于明尼苏达大学的明尼苏达因特网业务量研究(MINTS)分析)之间。到 2016 年,用户业务量预计将超过每月 100 艾字节,即超过 100,000,000 兆字节,或者超过每秒 42,000 千兆字节。然而,考虑到高峰期时超过 6 亿用户同时使用因特网高分辨率视频,峰值需求将会明显更高。

[0007] 所有这些数据将发生在经由数据中心流动的用户或者访问数据中心中的内容的用户之间,并因而发生在数据中心之间以及数据中心内,这些 IP 业务流必须被放大许多倍以建立真正的总 IP 业务流。数据中心布满被电缆架包围的高的电子设备机柜,数据通常被存储在大型且快速的硬盘驱动器上,计算机服务器接受请求并且使用当下的快速电子交换机来移动数据,从而访问正确的硬盘驱动器。在这些数据中心的外围,路由器将计算机服务器连接到因特网。同时,由于这些不断增长的需求,诸如云计算之类的演进应用也与日俱增,其中计算平台不再是独立的系统而是被托管在大规模数据中心中的同质互联计算基础设施,其被称为仓库级计算机(WSC),WSC 提供无处不在的互联平台作为共享资源用于具有某些要求的分布式服务,而这些要求难以通过数据中心的传统机架/服务器得到满足。这似乎不足以发展商业实践、商业模式、以及用户对于转移和/或存储的每比特成本的持续价格侵蚀(如果不是消除)的预期。

[0008] 因此,有必要开发出在内部连接数据中心和 WSC 以及实现互联的经济有效且可扩展得方法,以使得数据中心和 WSC 以及电信网络能够应对带宽和速度的呈指数式增长的需求所带来的挑战,而又不会显著增长基础设施的成本和功率。同时,对访问内容时的低时延或无时延的期望带来了额外的压力。因此,数据中心互联(在本文中,为简单起见,使用数据中心这一术语来包括 WSC、NAS、SAN、服务器集群以及传统的数据中心)已经成为决定因特网和大量采用数据中心互联的商业、金融和个人应用的整体性能的关键因素。光纤技术已经在数据中心运行方面发挥了重要作用,并且将不断以此为目标,即以最小的时延、最低的成本、最低的功耗和最小的空间占用尽可能快地移动数据。因此,千兆比特以太网太慢,并且当部署诸如以太网、光纤信道以及 Intel 的光峰的 10Gb/s 解决方案时,会受到限制。光纤信道已经接近 16/20Gb/s,以太网正向 40Gb/s 和 100Gb/s 迈进。在 40Gb/s 和 100Gb/s 下,现有及正在形成的标准需要多个 10Gb/s 信道在平行多模光纤电缆上运行或者需要在单模光纤上进行波分复用(WDM)。对于 40Gb/s,多模光纤链路通常需要 8 根光纤,即每个方向具有 4x10Gb/s;对于 100Gb/s 则需要 20 根光纤,即每个方向具有 10x10Gb/s。按照标准,对于采用 OM3/OM4 多模光纤的 100GBASE-SR10 链路,这分别限定了 100m/150m 的距离。相

比之下,对于 100GBASE-LR4 单模光纤 (SMF) 链路以及诸如 10x10G MSA 的其他旧式电信解决方法,所限定的距离为 10km。10x10G MSA 也包括用于 40km 链路的提案标准。

[0009] 然而,当数据传输速度增加到给定的信息块在更短的时间段内被传输时,这些进化系统将无法解决时延,即提出数据请求与该数据被提供之间的时间,也无法解决会增加时延的网络故障。通常,这些光网络所采用的架构(例如)为线性、总线或环形网络,其中,网络上任意两个节点之间的时延取决于该两个节点与网络接入点之间的距离以及这些网络接入点在网络上的相对位置;或者为星形网络,其中,时延取决于其与核心的距离以及中央交换结构将数据从网络的一个“臂”路由到另一个臂的能力。当速度、成本和某些时延通过针对网络的“臂”采用光纤链路而得以解决时,中央交换结构仍采用较大的电子交叉连接以处理链路容量。

[0010] 光交换结构迄今主要聚焦于空间交换应用,例如保护、故障恢复,以及可重新配置的塞取多路复用器,交换速度大约为毫秒,而不是提供数据内的帧或其他块水平的交换所需的纳秒及皮秒。当用于在这些时间尺度上运行的光交换设计存在(例如)基于铌酸锂、砷化镓和磷化铟的光电路时,其成本、性能和可扩展性将这些光交换设计的部署限制在演示和实验测试床内。但是,明显的是,如果来自一个节点的数据被同时分配给所有节点而不是通过环形网络、线性网络、总线网络、网格网络或核心电子路由进行路由,则时延可被降低。

[0011] 根据本发明的实施例,发明人提出了基于全无源光网络的架构,并将全无源光网络称为无源光交叉连接网络 (POCXN),该架构支持用于减小时延的分配方式以及与架构的部署相关的协议。有利的是,这一 POCXN 概念采用已经得到大批量生产技术支持的光学部件。

[0012] 通过结合附图审阅下述对本发明的具体实施例的描述,本领域的普通技术人员将清楚本发明的其他方面和特征。

发明内容

[0013] 本发明的目的在于解决与光网络相关的现有技术中的问题,尤其涉及提供用于减小时延的无源光交叉连接网络。

[0014] 根据本发明的实施例,提供了一种设备,包括:

多个分配级,每个分配级包括预定数量的具有输入端口和输出端口的无源光分路器,所述无源光分路器将所述输入端口中的每一个处的光信号同时耦合到所有所述输出端口;以及

多个互联级,每个互联级将所述多个互联级的第一预定级连接至所述多个互联级的第二预定级,以使得所述光信号的来自所述多个互联级的所述第一预定级的第一预定数量的无源光分路器的输出端的预定部分以预定的方式被耦合至所述多个互联级的所述第二预定级的第二预定数量的无源光分路器的输入端的预定部分。

[0015] 根据本发明的实施例,提供了一种方法,包括:

提供一种无源光交叉连接网络 (POXN),该 POXN 包括第一组多个输入端口和第二组多个输出端口;

提供耦合至所述 POXN 的多个节点以使得所述多个节点中的一个节点发送的信号在不存在任何交换功能的情况下被自动分配到所述多个节点;

提供在每个节点上执行的协议引擎,所述协议引擎执行单个节点发现序列和至少一个数据转移序列的周期性重复处理;

其中,在所述至少一个数据转移序列期间每个节点所发送的数据依靠在所述节点发现序列期间请求发送数据的节点的数量以及每个请求发送的节点必须发送的数据量来进行确定。

[0016] 同意时,插入独立权利要求以进行完善。

[0017] 根据本发明的实施例,提供了一种方法,包括:

提供一种无源光交叉连接网络 (POXN),该 POXN 包括第一组多个输入端口和第二组多个输出端口;

提供耦合至所述 POXN 的多个发射器节点以使得所述多个发射器节点中的发射器节点发送的信号在不存在任何交换功能的情况下被自动分配到多个接收器节点,每个接收器节点在预定波长上进行发射;

提供耦合至所述 POXN 的所述多个接收器节点以使得所述多个发射器节点发送的信号在不存在任何交换功能的情况下被自动分配到所述多个接收器节点,每个接收器节点从所述多个发射器节点中的所有正在发射的发射器节点接收所述信号;

提供多个电子处理元件,所述多个电子处理元件从所述多个发射器节点接收多个接收信号,每个电子处理元件从所述多个接收信号的每个接收信号中滤波掉不用于所述多个接收器节点的一个接收器节点的全部内容,该接收器节点与该电子处理元件相关联;以及

提供多个缓冲节点,每个缓冲节点从所述多个发射器节点接收经滤波的接收信号并将所述经滤波的接收信号排序用于传输到被连接至该缓冲节点的数据宿。

[0018] 根据本发明的实施例,提供了一种方法,包括:

提供第一无源光交叉连接网络 (POXN),该第一 POXN 包括第一组多个第一输入端口和第二组多个第一输出端口;

提供第二无源光交叉连接网络 (POXN),该第二 POXN 包括第一组多个第二输入端口和第二组多个第二输出端口;

提供多个数据源,每个数据源被耦合至所述第一组多个第一输入端口的预定第一输入端、所述第一组多个第一输出端口的预定第一输出端、所述第二组多个第二输入端口的预定第二输入端、所述第二组多个第二输出端口的预定第二输出端。

[0019] 根据本发明的实施例,提供了一种方法,包括:

提供多个无源光交叉连接网络 (POXN),每个 POXN 包括多个输入端口和多个输出端口;以及

提供多个数据源,每个数据源被耦合至所述多个 POXN 的每个 POXN 的预定输入端以及被耦合至所述多个 POXN 的每个 POXN 的预定输出端。

[0020] 通过结合附图审阅下述对本发明的具体实施例的描述,本领域的普通技术人员将清楚本发明的其他方面和特征。

附图说明

[0021] 现在将仅以示例的方式参考附图对本发明的实施例进行描述,其中:

[0022] 图 1A 和 1B 描绘了根据现有技术所述的熔融光纤无源光功率分路器/合路器;

- [0023] 图 2 描绘了根据本发明实施例所述的 27x27 星形耦合器的示例性架构；
- [0024] 图 3 描绘了根据本发明实施例所述的 27x27 星形耦合器的示例性架构；
- [0025] 图 4 描绘了随根据现有技术和本发明实施例所述的星形耦合器的端口数变化的多余光损耗；
- [0026] 图 5 描绘了根据本发明实施例所述的结合星形耦合器的基于示例性 WDM 的链路拓扑；
- [0027] 图 6 描绘了根据本发明实施例所述的作为采用星形耦合器的网络协议一部分的发现序列；
- [0028] 图 7 描绘了根据本发明实施例所述的作为采用星形耦合器的网络协议一部分的数据转移序列；
- [0029] 图 8 描绘了根据本发明实施例所述的无源光交叉连接网络；
- [0030] 图 9 描绘了部署在二维超级 POXN 内根据本发明实施例所述的无源光交叉连接网络；
- [0031] 图 10 描绘了部署在二维超级 POXN 内根据本发明实施例所述的无源光交叉连接网络；以及
- [0032] 图 11 描绘了部署在三维超级 POXN 内根据本发明实施例所述的无源光交叉连接网络。

具体实施方式

[0033] 本发明涉及光网络，尤其涉及一种用于减小时延的无源光交叉连接网络。

[0034] 随后的描述仅提供了示例性实施例，并不用于限制本公开的范围、应用性或配置。更确切地说，随后对本发明示例性实施例的描述将向本领域技术人员提供能够实施示例性实施例的描述。应该理解，在不偏离所附权利要求中提出的精神和范围的情况下，可对一些元件的功能和布置做出各种改变。

[0035] 此部分以及整个公开中所使用的“便携式电子设备 (PED)”是指用于通信和其他应用的无线设备，该无线设备需要电池或者其他独立形式的能量用于供电。这包括但不限于移动电话、智能手机、个人数字助理 (PDA)、便携式计算机、寻呼机、便携式多媒体播放器、便携式游戏控制台、膝上型计算机、平板计算机以及电子阅读器。此部分以及整个公开中所使用的“固定电子设备 (FED)”是指用于通信和其他应用的无线和 / 或有线设备，该无线和 / 或有线设备需要连接到固定的接口以通电。这包括但不限于膝上型计算机、个人计算机、计算机服务器、信息亭、游戏控制台、数字机顶盒、模拟机顶盒、因特网使能设备、因特网使能电视以及多媒体播放器。

[0036] 无源光交叉连接：参考图 1A，其描绘了根据现有技术所述的熔融光纤无源光功率分路器 / 合路器的第一视图 100A 和第二视图 100B。参考第一视图 100A，其描绘了熔融 2x2 耦合器，其中，第一光纤 105A 和第二光纤 105B 分别在（例如）施加火焰的情况下在一区域中被拉伸和绞合到一起，以使得光纤软化并且可被拉制使得其纤芯和包层减小，从而增大一根光纤与另一光纤在长度为 d 的中央锥形区域 115 内的耦合。还描绘了第一锥形部 110A 和第二锥形部 110B，其中光纤朝向中央锥形区域 115 从其初始几何尺寸（例如 9 μm 纤芯和 125 μm 包层）逐渐变窄。因此，根据光纤的拉锥工艺和端面，一根光纤（例如第一光纤

105A) 内的光信号在两个输出端之间以功率 P_{OUT1} 和 P_{OUT2} 进行分配。因此 $0 \leq P_{OUT1} \leq P_{IN}$ 以及 $P_{OUT2} = P_{IN} - P_{OUT1}$, 以使得可实现多个分光比, 例如 10:90、25:75、50:50 等。通常在形成星形耦合器时, 使用 50:50 的耦合器。

[0037] 生成的熔融光纤耦合器易碎, 所以通常被组装在保护壳体中, 如第二视图 100B 所示。在这里, 熔融耦合器 135 使用环氧树脂 130 被安装在石英基板 145 上。光纤 (通常具有最初的 $250 \mu\text{m}$ 涂层) 处于管 120 内, 管 120 的每个端部同样被附接在石英基板 145 上。然后这一组件通常被插入金属管 125 中, 并且保持套管 150 被附接以密封金属管 125 的端部以及把持管 120。这样的封装的示例由第三视图 100C 中的第一组装分路器 155A 和第二组装分路器 155B 进行描绘, 155A 和 155B 分别使用 $900 \mu\text{m}$ 次级管和 $250 \mu\text{m}$ 初级涂层, 封装直径为 5.6mm 以及长度为 76mm ; 也通过第三视图 100C 中的第三组装分路器 160A 和第四组装分路器 160B 进行描述, 160A 和 160B 同样分别采用 $900 \mu\text{m}$ 次级管和 $250 \mu\text{m}$ 初级涂层, 封装直径为 4mm 。为了减小这样的耦合器的占用面积, 一些供应商, 例如 Viscore, 推出了直径为 3.0mm 并且长度为 54mm 或 60mm 的熔融光纤封装。在图 1 的第四视图 100D 中, 还描绘了第一至第三星形耦合器 165、170 和 175, 分别表示 4×4 模块、 16×16 模块以及 32×32 的安装在机架上的星形耦合器。

[0038] 参考图 1B, 其分别描绘了第一网络 180 和第二网络 190, 180 和 190 代表了用于实现具有 $M = 2^n$ 的星形耦合器所谓的“全混洗”架构以及 Banyan 架构, 其中对于根据现有技术所述的 4×4 、 8×8 、 16×16 等星形耦合器, $n = 2, 3, 4 \dots$ 等。实现“全混洗”的第一网络 100E 采用 $R = M/2$ 2×2 耦合器的 $n = 3$ 个臂, 其中, 在每对臂之间实现了“全混洗”。相比之下, 第二网络 190 同样采用 $R = M/2$ 2×2 耦合器的 3 个臂, 但是此时 2×2 耦合器的第一臂与第二臂之间的混洗和 2×2 耦合器的第二臂与第三臂之间的混洗是不同的。在 2×2 耦合器的第一臂和第二臂内, 混洗路由被划分以使得第一臂的上半部与所有的耦合器耦合并且第一臂的下半部同样与所有的耦合器耦合, 但在 2×2 耦合器的第二臂与第三臂之间, 混洗被分开以使得混洗的上半部仅与上半部耦合并且下半部同样仅与下半部耦合。根据第二网络 190 中所描述的 Banyan 架构, 在对星形耦合器设计进行扩展的情况下, $X = 2^{n-1}$ 耦合器作为 $R = M/2$ 2×2 耦合器的子集, 此时每个域区间为 $1 = 1, 2, \dots, n$; 对于这样的 32×32 的耦合器, 在每个域中, 两组 2×2 耦合器耦合域在 16, 8, 4 之间。

[0039] 因此, 明显的是, 这样的星形耦合器的复杂度具有严重的问题, 因为熔融光纤耦合器阵列必须按照每根光纤必须路由的位置的限定序列被连接在一起。这需要较大的器件, 例如在图 1A 中被表示为第三星形耦合器 175 的安装在机架上的 32×32 星形耦合器。进一步地, 随着 M 的增大, 插入损耗增大 3dB (无源分路损耗), 另外还有熔融光纤 2×2 耦合器的额外损耗 (对于高质量熔融耦合器通常为 0.4dB) 以及针对每个额外等级的 2×2 耦合器的互联损耗 (通常为 0.1dB)。

[0040] 因此, 考虑 27×27 星形耦合器的要求, 则所要求的等级数为 $n = 5$, 即 $M = 32$, 因为先前的等级, 即 $n = 4$, 支持高达 16 个端口。因此, $n = 5$ 的理想分路器的插入损耗为 $Loss = 3\text{dB} \times n = 3\text{dB} \times 5 = 15\text{dB}$, 以及对于典型的组装分路器, 插入损耗为 $Loss = 3.5\text{dB} \times 5 = 17.5\text{dB}$ 。因此, 有利于提供通过熔融光纤器件实现星形耦合器的途径, 可随着端口数的增加而减小器件的数量以及插入损耗。因此, 发明人对使用 $Z \times Z$ 耦合器的星形耦合器组件进行了分析, 其中 $Z = 2, 3, 4, \dots$ 。图 1B 的第一视图描绘了 1×3 耦合器的截面图, 某些耦合器

(比如 1x3 耦合器) 提供了高度对称的用于熔融的结构, 从而提供了较其他耦合器 (比如 1x4) 改进的均匀性。

[0041] 对本领域普通技术人员而言, 明显有利的是, 星形耦合器从一个输入端口到任一输出端口的时延实际上是恒定的, 这不同于环形、总线形和网格形架构中的节点到节点时延。考虑到光学结构中不包含有源器件的系统, 即光域中不存在用来提供信号放大或重新生成的光学放大, 则功率预算成为耦合器结构的端口密度的确定性因子。为了计算耦合器结构的功率预算, 首先需要估计来自系统的其余部分的功率损耗, 其主要包括复用器 / 解复用器插入损耗和光纤传输损耗。具体来说, 对于基于薄膜滤波器的 CWD, 复用器 / 解复用器造成的功率损耗大约为 2.5dB。假设损耗为 0.3dB/km 并且长度为 10km (通常为数据中心内的最大距离要求), 则复用器 / 解复用器与耦合器结构之间的每个光纤段的链路预算为 3dB。在大多数交叉连接应用中, 光纤长度会远小于 10km。在系统层面, 来自发射器到达接收器的光信号行进经过一个复用器、一个解复用器以及长度最大为 20km 的光纤, 从而产生 11dB 的功率损耗。考虑到 35dB 的系统功率预算 (这对于现有的光器件技术是可行的), 则针对耦合器结构的功率预算为 24dB。

[0042] 发明人也开发了用于具有较多的信道端口数量的星形耦合器的设计, 这些设计结合了熔融光纤耦合器设计 (比如 2x2 和 3x3 耦合器), 从而提供所需要的设计。参考图 2, 其描绘了根据本发明实施例所述的 27x27 星形耦合器的示例性架构 200, 其中采用了树型配置的 3x3 耦合器 210。每个 3x3 耦合器 210 具有 4.77dB 的本征损耗, 并且基于现有的熔融光纤技术批量制造的额外损耗为 0.5dB。如图 2 所示, 架构 200 最初将所有输入信号并入单个 3x3 耦合器 210 中, 然后在所有输出端之间再次使用 3x3 耦合器 210 相等地分路信号功率。对于 N 输入端以及 N 输出端, 这一设计需要 $2[\log_3 N]-1$ 级 3x3 耦合器 210。考虑到用于互联相邻级的接头损耗为 0.2dB 以及 0.5dB 的额外损耗, 则每个输出端处的功率损耗为 $5.47(2[\log_3 N]-1)-0.2$ dB。因此, 对于 24dB 功率预算, 发射器与接收器之间的端口数量 N 在超过该损耗之前只能达到 9。该设计的问题在于三分之二的功率在每个合路或分路操作期间被完全浪费掉。

[0043] 同样地, 参考图 3, 其描绘了根据本发明实施例所述的 27x27 星形耦合器的示例性架构 300, 其中采用了经修改的 Banyan 配置的 3x3 耦合器 330, 所述经修改的 Banyan 配置使用提供 9x9 星形耦合器的第一子块 310 以及使用 3 个 3x3 耦合器 330 的第二子块 320。可提供更高效设计的架构 300 使用 Banyan 或基线拓扑将 3x3 耦合器 330 互联以使得全部输入功率在相等功率分路的情况下被递送到输出端。与图 2 中的前一设计相比, 示例性架构 300 内的级数从 5 减至 3。更一般地说, 对于 N 输入端以及 N 输出端, 只需要 $[\log_3 N]$ 级的 3x3 耦合器 330, 这在任一输出端处对应于 $5.47[\log_3 N]-0.2$ dB 的功率损耗。对于 24dB 的功率预算, 耦合器结构的端口数量可以扩展到 $N=81$ 。这一情况下每个输出端处的功率损耗为 21.68dB。请注意, 如果具有相同额外损耗的 3.01dB 2x2 耦合器用于构建耦合器结构, 则该耦合器结构的端口数量可以高达 $N=64$, 输出端处的精确功率损耗为 22.06dB。

[0044] 输出端处的功率损耗包括: 功率分路造成的基本损耗, 以及由耦合器额外损耗和光纤接头损耗引入的附加损耗。使用 3x3 和 2x2 光耦合器对附加损耗的贡献分别为 $0.7[\log_3 N]-0.2$ dB 和 $0.7[\log_2 N]-0.2$ dB。图 4 对这些额外损耗进行了绘制, 其中, 从额外损耗和接头损耗的角度来看, 3x3 耦合器是明显优于 2x2 耦合器的构建块。因此, 基于发明

人所进行的分析表明,与基于 2x2 熔融光纤耦合器的现有技术的构建方法不同,星形耦合器在与熔融器件组装时,提供了更强的性能。相应地,考虑到 3x3 构建块,则随着 n 的增加,端口数量为 3、9、27、81 以及 243,而现有技术设计的端口数量为 2、4、8、16 以及 32。不过,发明人也分析了采用 3x3 耦合器和 2x2 耦合器的混合星形耦合器设计,使得中等数量的端口得以实现,其中支持的端口数量 N 可以是具有且仅具有质因数 3 和 / 或 2 的任意整数,例如 $N = 48$)。

[0045] 对本领域普通技术人员而言,明显有利的是,星形耦合器从一个输入端口到任一输出端口的时延(如图 3 和图 4 中的示例性架构 300 和 400 所示)实际上是恒定的,这不同于环形、总线形和网格形架构中的节点到节点时延。

[0046] 明显的是,可以使用各种其他架构,这些其他架构采用小型 ZxZ 耦合器(例如 2x2 和 3x3)的不同组合,从而提供所需的星形耦合器功能。同样明显的是,可使用其他小型 ZxZ 耦合器,只要这些 ZxZ 耦合器具有高性能,包括(例如)4x4 和 6x6 等。进一步地,某些架构可采用小型 MxN 耦合器,比如 3x1 和 4x2。当之前相对于星形耦合器和小型 ZxZ 耦合器进行表示和讨论的架构相对于采用熔融光纤器件技术进行讨论时,对于本领域的普通技术人员而言,明显的是,可交替采用自由空间或其他光波导技术以提供整个星形耦合器结构或对小型 ZxZ 耦合器构建块进行连接。自由空间技术连同微光 ZxZ 耦合器(同光纤一样)在不存在干扰(串扰)的情况下,允许不同路径之间存在低损耗跨接。进一步地,某些 POXN 架构可使用反射结构进行“折叠”以使得节点的发射器端口和接收器端口可被耦合至 POXN 的同一侧而不是节点的所有发射器端口被耦合至一侧且节点的所有接收器端口被耦合至其他侧。使用合适的附加光学器件时,在每个节点处,发射 / 接收信道在单根光纤上而不是一对光纤上进行双向发射。

[0047] 可选地,各种折叠的全连接网络(包括但不限于 Banyan、碟形以及全混洗网络)可用于部分或整体提供光耦合器结构。在本发明的其他实施例中,一个或多个绕射光栅可用作自由空间光学混合版本的一部分以提供分路功能。在本发明的其他实施例中,光栅可配置以使得可配置的结构能够得以实现。进一步地,在本发明的其他变型中,混合自由空间光耦合器可作为一个构建块,例如 4x4 和 6x6,以在根据本发明实施例所述的 POXN 中使用。在使用绕射光栅的某些折叠的 POXN 元件中,这些光栅中的一部分可被波长选择开关(WSS)所代替。

[0048] 上述的星形耦合器是使得能够在数据中心内构建光交叉连接网络的无源分配器件。因此,用于这样的交叉连接的主要光学器件为 N xN 非波长依赖型耦合器结构,其具有 N 个输入端和 N 个输出端。每个输入端的光功率在 N 个输出端之间进行等分以使得在光交叉连接网络中,在任一输入端与任一输出端之间建立回路以及在任一输入端与任一输出端之间建立时延时,不需要进行重新配置。这使得无源光交叉连接能够支持各种类型的业务。然而,最初也考虑到带宽效率会减小,因为在不同的输入 - 输出对之间不可能进行空间波长复用,在有源交换的交叉连接中会使用不同的输入 - 输出对。但是,为了完全理解光学技术的带宽效益,在耦合器结构的每个输入端处使用波分复用(WDM)。在该分析中,受到数据中心部署对成本的敏感性的推动,在现有设计中倡导 CWDM。凭借现有的 CWDM 技术,每个波长可在 10Gb/s 下运行,且每个端口可支持 18 个波长。因此,在这种情况下,光交叉连接网络的总带宽容量为 180Gb/s。此外,可在每个波长上使用时分复用(TDM)以提供子波长粒度下

的带宽利用率,这从以下分析中可明显看出。在本发明的其他实施例中,在根据链路长度、成本等产生的限制的情况下,当在 C 波段 (1528.77nm 至 1563.86nm) 内 100GHz 处使用 DWDM 的情况下波长的数量增至 44 个波长时或者使用交织器以及在 1550nm 区域中采用 S 波段和 L 波段扩展的情况下波长数量更多时,每个波长可在增大的数据速率下运行,例如 20Gb/s、40Gb/s 等。

[0049] 参考图 5,其描绘了传输系统 500,该传输系统 500 使用具有第一至第三节点 510、520 和 530 的交叉连接结构,第一至第三节点 510、520 和 530 分别表示耦合至 $2M \times 2M$ 耦合器结构 540 的 M 个节点。在本说明书的剩余部分,这样的交叉连接结构被称为无源光交叉连接 (POXC) 结构。在第一至第三节点 510、520 和 530 的每一个中,网络接口卡 (NIC) 550 具有发射端口和接收端口。每个发射端口 (作为复用器 580 的输出端) 配有位于发射器块 560 内的固定 (或可调) 发射器阵列,所有发射器均在不同的波长下工作,这些发射器由第一至第 N 发射器 565A 至 565N 表示。起初,这些不同的波长被不同的光纤所携带,其中,所有的波长信号随后通过复用器 550 并入一根光纤。接收端口是发射端口的反面,其能够接收所有 N 个波长。接收端口与发射端口具有相似的结构,其中,复用器 580 和发射器块 560 分别被解复用器 590 和接收器块 570 所代替。接收器块 560 包括第一至第 N 接收器 575A 至 575N。为使精确时钟同步和延迟测量能够实现,假设具有一个 NIC 550 的发射器和接收器与耦合器结构具有相同的光纤距离,以使得从信源 NIC 550 到信宿 NIC550 的传播延迟与从信宿返回信源的传播延迟相同。这一假设通常在实际系统中得以满足。

[0050] 无源光交叉连接协议:对于本领域普通技术人员而言,明显的是,由于其广播特性,POXC 结构意味着带宽资源在所有的输入端之间进行共享。因此,为了在输出端口避免来自不同输入端口的帧碰撞,重要的是开发一种在链路层协调端口传输的多接入协议。在前述描述中,18 波长 10G/s 的 CWDM 部署被认为提供了 180Gb/s 的总带宽容量。理想情况下,如果所有 N 个输入端都有数据要发送,则多接入协议应使得对于 N 个输入端,能够达到每端口 $180/N$ Gb/s 的平均传输速率;以及如果仅一个端口有数据要发送,则瞬时传输速率为 180Gb/s。但是,控制开销是无法避免的,从而导致任一情况下传输速率降低。

[0051] 发明人通过利用端口的对等特性以及介质的广播特性,提出了一种具有服务质量 (QoS) 支持的高度分布协议。被称为高效分布式接入协议 (HEDAP) 的协议,对于主-从基础设施可实现高于轮询方法的效率的同时保持完全分布式操作以避免单点故障。HEDAP 被分为两个阶段:发现阶段和数据传输阶段。发现阶段旨在实现可使网络的运行成本最小化的即插即用目标。在该阶段期间,POXN 中的端口将有机会发现该 POXN 中的其他端口、建立公共参考时钟、将时钟同步到所述参考时钟以及测量往返时间和回送时间。图 6 中的发现序列 610 已示出发现阶段,并且其后具有多个数据转移序列 650,每个数据转移序列在调度周期 640 内发生。下文给出了发现阶段的细节。

[0052] 图 7 所描绘的数据转移阶段在发现阶段之后,但是其完成的时间较发现阶段长,以使得由发现阶段引入的开销可实现最小化。在发现序列 610 之后,数据转移阶段被进一步分为多个调度周期 640,每个调度周期包括数据转移阶段 650。在发现阶段所发现的每个端口将有机会在每个调度周期内发送突发的帧。在突发的末尾,端口还会公布其在下一个周期期间需要发送的业务量。捎带方法使得所有其他端口能够通过基于广播的 POXN 获知请求的同时可用于维持——来自不同信源端口的业务突发之间的——端口间保护间隔的

带宽浪费最小化。与轮询方法（主节点轮询从节点，基于响应制作调度表，然后通知从节点其相应的调度表）不同，POXN 中的端口可在获知来自其他端口的所有公布之后，在本地层面单独制作自己的调度表。只要公共调度算法被所有端口使用，就可以避免调度冲突。这类类似于对等网络中所使用的方法，在对等网络的该方法中，公共哈希函数被所有的对等体所使用以建立分布式哈希表（DHT），而不需要中央服务器。利用发现阶段中所测得的回送时间，可对时钟漂移和传播延迟进行补偿，以使得来自不同端口的调度表可在周期内被及时地完全对准。因为不存在轮询方法中的轮询消息，周期可以一个紧接着一个，而之间的间隙很小。这进一步减小了与轮询方法相关的开销。

[0053] 考虑到端口的流失，发明人允许在数据转移阶段运行足够长时间以后重复发现阶段。发现阶段的重复使得：能够发现新的端口，时钟能够被重新参考与重新同步（当发生时钟参考端口故障时），以及往返时间和回送时间能够被再次测量（当发生时钟参考端口故障时）。发现阶段之间的时段应当足够长以容纳尽可能多的调度周期，同时应该足够短以使加入网络的新端口的延迟和对时钟漂移的影响最小化。协议可作为个别情况进行实施以将多个波长作为逻辑上的一个资源块进行管理，或者作为多个独立情况进行实施，每种情况针对一个波长，以实现更好的可管理性、灵活性和可靠性。两种实施模式具有相同的带宽效率。下面分别参考图 6 和图 7 对这些阶段进行描述。

[0054] 发现阶段：系统启动时的发现阶段——当系统启动时，所有端口开始监听 POXN 一段时间。基于在该时段内不会检测到传输的事实，每个端口得知系统刚刚启动。然后，所有端口在各自的随机回退时段之后开始发送 DISCOVERY（发现）消息以使得自身能够被所有其他端口所识别的同时最小化消息碰撞。DISCOVERY 消息携带发送方端口的 MAC 地址、发送消息的时间戳、用于发现窗口的时段以及在第一调度周期中待发送的业务量。在进行发送时，端口也在其接收侧监听 DISCOVERY 消息。这些消息包括端口自身发送的消息。在没有传输协调的情况下，很有可能发生消息碰撞。碰撞的消息在端口的接收侧的 MAC 层进行检测。相似的技术已用于 EPON (IEEE 标准 802.3-2008) 中。通过在发送每个 DISCOVERY 消息之前强加随机回退来避免持续的碰撞。发现窗口始于所有端口对第一 DISCOVERY 消息的正确接收，并且其持续时间如该消息中指定的那样。

[0055] 由于每个端口可收听到来自信道的完全相同的消息流，所以端口可安全地推断出在该端口的本地接收器处所成功接收到的第一 DISCOVERY 消息也就是所有其他端口所成功接收到的第一 DISCOVERY 消息，以使得所述发现窗口是完全一致的。如果多个波长由单个协议实例进行管理，则可在同一时刻接收不同波长上的消息。在这种情况下，所有端口使用同一判据策略以选择相同的第一消息。所有 DISCOVERY 消息必须在发现窗口内被任意端口所接收。由于端口到端口的传播延迟，允许发送 DISCOVERY 消息的实际窗口（由 t_{TW} 表示）小于所通告的发现窗口 t_{TW} 。由于此时端口到端口的传播延迟未知，所以 20km 最大光纤距离上的最坏情况下的传播延迟 $T_{PP_MAX} = 100\ \mu.s$ 被假设为接近安全传输窗口。因此得到 $t_{LW} = t_{DW} - T_{PP_MAX}$ 。如果在 t_{TW} 内检测到任何消息碰撞（可以是或者不是其自身的 DISCOVERY 消息），则端口将在新的随机回退时段之后调度重新传输直到其自身的 DISCOVERY 消息在其本地接收器处被适当地接收（因而也被所有其他端口所接收）或者直到所述传输窗口 t_{TW} 结束。对 DISCOVERY 消息的成功接收表明发送方端口已为所有端口所知。这样的端口被称为成功端口。第一成功端

口决定发现窗口,并且其时钟还作为用于同步的参考时钟。在图6中,假设端口1为第一成功端口,并且端口2和端口3依次为第二成功端口和第三成功窗口。端口1则可决定其回送时间为 $T_{L1} = t_{12} - t_{11}$ 。同样地,端口2和端口3可决定其相应的回送时间分别为 T_{L2} 和 T_{L3} 。在第一成功端口所通告的时间窗口的时段到期之后,不允许其他端口进行发送。

[0056] 然后,第一端口将广播 CONFIRMATION(确认)消息以向每个已知端口总结信息,比如端口的 MAC 地址、来自端口的 DISCOVERY 消息的传输时间(即该 DISCOVERY 消息内的时间戳)、DISCOVERY 消息在第一端口处的接收时间、以及第一调度周期的业务请求。CONFIRMATION 消息还会通知后续数据传输阶段的开始时间 t_{C1} 、所包含的调度周期的总数量、以及用于下一发现阶段的发现窗口的大小。然后,每个端口可使用这些信息以决定到达第一端口的往返时间以及相对于第一端口的参考时钟的时钟偏移。以端口2为例。使 t_{12_rtt} 和 t_{offset} 分别表示其到达端口1的往返时间以及相对于端口1的时钟偏移。极易确定 $t_{13} = t_{21} t_{offset} + 1/2 t_{12_rtt}$, 以及 $t_{22} = t_{15} + t_{offset} + 1/2 t_{12_rtt}$ 。因此,得到 $t_{12_rtt} = (t_{22} t_{15}) + \left(t_{13} - t_{21} \right)$, 以及 $t_{offset} = 1/2 \left[\left(t_{22} t_{15} \right) - \left(t_{13} - t_{21} \right) \right]$ 。然后,端口2使用偏移值来校正其时钟。

[0057] 当 CONFIRMATION 消息被所有端口接收时,所有时钟都被同步到参考时钟。每个端口将按照各个端口成功发送其 DISCOVERY 消息的顺序以及各个端口在即将到来的第一调度周期中待发送的业务量,维护发现阶段所发现的所有端口的列表。利用该列表,每个端口可为所有端口制作调度表以决定其发送自身业务的时间和可发送的业务量。在假设所有端口使用相同的算法的情况下,极易看出每个端口在本地产生的调度结果完全相同以确保在第一调度周期内不存在与其他端口的突发碰撞。如果未接收到 CONFIRMATION 消息,则端口会再次重启。

[0058] 正常操作期间的发现阶段——一旦开始数据转移,每个数据转移阶段紧随发现阶段。在每个发现阶段期间,现有端口不发送 DISCOVERY 消息,因为相互认识。现有端口将监听和记录来自自由潜在新端口所产生的任何 DISCOVERY 消息的信息。当新端口试图加入 POXN 时,其需要等待 CONFIRMATION 消息以获知共享信道的所有现有端口,获取下一发现窗口的大小以及需要得知当前数据转移阶段中调度周期的数量以进一步确定下一发现阶段的开始时间。这意味着新端口在使自身被其他节点所认识之前,必须等待至少一个数据转移阶段。在等待时段内,新端口监视网络、记录来自其接收的所有消息的信息、并且追踪调度周期以计算下一发现窗口的精确开始时间。

[0059] 在下一发现阶段期间,只有新端口需要发送 DISCOVERY 消息。这将最小化碰撞。在发现窗口结束之后,在系统启动时,第一成功端口(或者更精确地,当前时钟参考端口)将发送 CONFIRMATION 消息,如果可能的话,会考虑端口故障。CONFIRMATION 消息包含:与现有和新加入端口有关的信息、后续数据转移阶段、以及下一发现阶段。所有端口将使用该信息以计划其将来的操作事件。如果未接收到 CONFIRMATION 消息,端口将会认为在发现阶段内时钟参考端口碰巧故障,然后重新启动。但是,考虑到下文所示的发现阶段的短持续时间,这样的故障事件应该很罕见。

[0060] 数据转移阶段——调度算法将决定端口可进行发送的时间以及可发送的量。不对哪些调度算法可被使用进行限制,只要所有端口遵循同一调度算法。例如,在图 6 中,假设所使用的调度算法决定在第一调度周期中发送数据业务的顺序为端口 2、端口 3、端口 1。然后,数据转移阶段开始于端口 2 在 $t_{23} = t_{C1} - TL_2/2$ 时刻发送其自身的数据突发以使数据突发如在 CONFIRMATION 消息中所定义的那样准确地在时刻 t_{C1} 到达耦合器结构。在完成其数据突发之后,端口 2 将公布其回送时间 t_{L2} 以及其对下一调度周期的请求(作为通过 REQUEST(请求)消息的捎带)。所有其他端口将在各自的列表中记录回送时间以及所请求的量用于下一周期计算。端口 3 在端口 2 之后进行发送。端口 3 将调度其业务突发以使得该业务突发正好在端口 2 所发送的捎带信息的最后一比特之后到达耦合器结构。端口 3 可以这样做,因为它知道到达耦合器结构的传播时间(即, $TL_3/2$)。如果即使在校正之后时钟也不完美,则需要一个保护时间以避免重叠。在完成其突发后,端口 3 会像端口 2 那样公布其回送时间以及对下一周期的请求。所有其他端口将进行记录。

[0061] 继续该处理直到最后一个端口完成其数据突发和捎带请求。当来自最后一个端口的 REQUEST 消息被所有端口接收时,每个端口获知所有端口的回送时间以及所有端口在第二调度周期中的请求。假设在第二周期中,调度算法所给出的传输顺序与第一周期中的传输顺序相同。然后,端口 2 马上开始第二周期。理想情况下,来自端口 2 的第一比特应当正好在上一周期的最后一比特之后到达耦合器结构。然而,问题在于端口 2 可能没有收到最后几个端口所发送的捎带请求,因为端口 2 需要较早发送其突发以减小由传播延迟导致的间隙。在未接收到最后几个端口的捎带请求的情况下,端口 2 仅具有部分请求信息用以计算针对下一周期的调度表。在这种情形下,端口 2 可能需要等待直到接收到来自所有端口的捎带请求,如图 6 所示。第二周期 t_{C3} 的开始时间由每个端口进行本地计算,即 $t_{C3} = t_{C2} + 1/2 \max\{TL_1, TL_2, TL_3\} + T_{proc} + TL_2/2$, 其中, t_{C2} 表示上一周期的最后一比特到达耦合器结构的时间, $1/2 \max\{TL_1, TL_2, TL_3\}$ 是该最后一比特从耦合器结构传播到最远的端口所用的时间, T_{proc} 是在端口处设置用于消息处理、算法运行等的恒定时间值,以及 $TL_2/2$ 是第一传输端口的第一比特在当前周期(即端口 2)到达耦合器结构的传播时间。请注意, t_{C2} 作为调度算法的输出在前一周期给出。

[0062] 同样,后续调度周期的开始时间相继得以确定。因此,通过从 CONFIRMATION 消息开始监听信道,信道接入事件的时间可贯穿整个数据转移阶段进行精确计算。这使得未知的新端口能够紧随数据转移阶段的最后一比特的接收,识别出发现窗口。周期之间的信道空闲间隙可通过开发更复杂的调度算法来消除,而今后可从此方面入手进行研究。

[0063] 在时刻 t_{15} ,当发送 CONFIRMATION 信息时,端口 1 仅知道其自身的回送时间。为使数据转移阶段能够在任一端口处开始,端口 1 假设最远端口与耦合器结构相距 10km。使 $TCP_{MAX} = 50 \mu s$ 表示对应的单程传播时间。因此,时间 t_{C1} 计算为 $t_{C1} = t_{15} + t_{CFMtrans} + t_{L1}/2 + 2TCP_{MAX} + T_{proc}$, 其中, $t_{CFMtrans}$ 表示消息传输时间。此外,由于每个端口在第一调度周期结束时得知所有端口的回送时间,所以此后端口之间的传输顺序可以随周期而改变。很容易看出,上述协议可支持 QoS,因为只要所有端口使用相同的确定性调度算法,就不会对调度算法增加任何特殊要求,这使得本发明的算法简单且对于不同应用都很具吸引力。

[0064] 当端口停止运行时,其他端口将检测到这一情况,因为停止运行的端口不再发送

对下一周期的请求。所有其他端口（除了在调度以后的周期时假设停止运行的端口不会发送任何数据之外）不需要执行任何操作。如果停止运行的端口是时钟参考端口，则所有其他端口仍以相同的方式运行直到下一发现阶段为止。在下一发现阶段期间，所有其他端口将从各自的列表中移出失效的端口。列表上的第二端口则变成时钟参考端口。

[0065] 功率效率——当端口不发送数据时，该端口关闭其发射器以避免干扰其他端口的传输。这也有助于节能。此外，当端口在非高峰期具有较少的数据要发送时，该端口可以简单地完全关闭自身的某些发射器使得看起来该端口在相应的波长上“停止运行”。这一特性进一步降低了能量，并且当协议基于每波长实施时，能够获得很好的支持。当业务需求增加时，这些发射器可通过利用周期性发现阶段这一机会再次开启。

[0066] 带宽效率——在下文中，通过数值示例来示出所提出的协议的带宽效率。如下面的表 1 所示，所使用的参数基于现有的 EPON 技术。假设耦合器结构已互联 64 个端口。首先，考虑由整个发现阶段产生的带宽开销，假设发现阶段每分钟被触发一次。发现窗口被认为是 $500\ \mu\text{s}$ ，其中， $100\ \mu\text{s}$ 用于消息传播，以及 $400\ \mu\text{s}$ 用于随机回退。应当注意，消息仅在耦合器结构内可能发生碰撞。因为耦合器结构内的消息传输时间和传播延迟两者都很小，所以两个或多个帧碰撞的几率实际上非常小。发现阶段以 CONFIRMATION 消息的传输而结束，CONFIRMATION 消息到达最远端口所花的时间为 $100\ \mu\text{s}$ 。因此，发现阶段的开销估计为 $[(500+100)\times 10^{-6}]/60\times 100\% = 0.001\%$ ，这对带宽效率的影响可忽略，因而不将其考虑在内。也就是说，带宽效率完全决定于数据转移阶段。假设一个简单的调度算法，该调度算法不会消除调度周期之间的信道空闲间隙，如图 6 和图 7 所示。信道空闲间隙包括：耦合器结构至最远端口的传播延迟、每个端口内恒定的处理时间 T_{proc} 、以及下一周期的第一传输端口到耦合器结构的传播延迟。在最坏的情况下，每个传播延迟为 $T_{\text{MAX}}^{\text{CP}} = 50\ \mu\text{s}$ 。因此，所提出的协议的带宽效率大约为 $[1-(10\times 10^{-9}+50\times 10^{-6}\times 2+2000\times 10^{-9}\times 63)/10^{-3}]\times 100\% = 77.4\%$ 。因此，每个端口具有 2.72×10^5 字节 / 调度周期的平均突发大小，或者等效地，具有 2.18Gb/s 的平均传输速率。如果一个端口有数据要发送，则该端口可用的最大突发大小为 1.74×10^7 字节，对应于 139.3Gb/s 的瞬时传输速率。明显的是，可通过使用高级调度算法进一步提高带宽效率，该高级算法可使调度周期之间的信道空闲间隙变小。

[0067] 协议——协议的一个示例如下所示。

定义 L 为排序列表

定义 R 为当前时间待发送的业务量

定义 Mac 为端口的 Mac 地址

定义 T_{Wait} 为初始等待时间

定义 $T_{\text{Discovery}}$ 为发现窗口大小

定义 $N_{\text{Scheduling}}$ 为两个发现阶段之间的调度周期的数量

定义 T_{First} 为 CONFIRMATION 封包的发送时间与数据转移封包的第一比特到达耦合器结构的时间之间的时间间隙

函数 getclock 返回本地时钟时间

函数 receive 返回帧或者所接收帧的突发

函数 send 发送帧或者帧的突发

函数 create 创建帧

```
函数 setTimeout 设置一件超时事件
初始化状态 = New
初始化 discovery_count = 1
1:while getclock<T_Wait do
2:packet = receive()
3:If packet = Confirmation_Packet then goto 40
4:end while
5:discovery_count = 0
6:if status = New&&discovery_count = 0then
7:while channel-busy do listening
8:end while
9:discovery_packet = create(Mac, Timestamp = getclock, T_Discovery, R)
10:set a random delay
11:send(discovery_packet)
12:end if
13:packet = receive()
14:if packet is corrupted goto 6
15:end if
16:if packet is redundant then
17:drop packet
18:goto 13
19:end if
20:if packet = discovery_packet then
21:loopback_time = clock-packet.timestamp
22:status = Old
23:end if
24:if packet is the first received then
25:settimeout(packet.T_Discovery)and goto 29 when timeout
26:first_port = packet.Mac
27:end if
28:insert packet info into L
29:goto 13
30:if Mac = first_port then
31:Confirmation_Packet = create(L, T_first, N_Schedule)
32:send(Confirmation_Packet)
33:end if
34:else
35:packet = receive()
36:If packet = Confirmation_Packet
```

```

37:clock_error = ((getclock-packet.timestamp)-(packet.Mac.receive_
time-packet.Mac.timestamp))/2
38:
    roundtrip_time = ((getclock-packet.timestamp)+(packet.Mac.
receive_time-packet.Mac.timestamp))/2
39:end else
40:count = N_Schedule
41:While count != 0do
42:count--
43:settimeout(schedule(L,T_first))goto 47when timeout
44:packet = receive()
45:insert packet info into L
46:goto 44
47:if discovery_count = 0then send(burst,R)
48:end while
49:discovery_count = 0
50:goto 6

```

操作时段	时间
发现窗口	500 μ s
调度周期	1ms
端口间保护间隔	2000ns
处理时间 T_{PROC}	10ns
T_{MAX}^{CP} 最坏情况下端口到耦合器结构的传播延迟, 以及耦合器结构到端口的传播延迟	50 μ s

表 1 :HEDAP 协议总结表

[0068] 上述示例性协议是可为连接至 POXN 的节点提供所要求的新颖网络功能的多个协议之一。正如上文所讨论的, 每个节点会发送和接收数据, 执行协议的节点的协议引擎会处理该数据, 从而形成由使用一个或多个微处理器的节点所完成的总体处理的一部分。

[0069] 基于无源光交叉连接的分布式光交换机: 在前述部分, 关于将无源光交叉连接用作连接具有低时延的 N 个节点到具有 $2N \times 2N$ 个端口的 POXN 的光网络的基础, 给出了一个协议, 其中, POXN 的 N 个端口从 N 个节点接收 CWDM (或 DWDM) 信号, 并且 N 个端口将来自 POXN 的分配的 CWDM (或 DWDM) 信号耦合至所述 N 端口。在本说明书的该部分, 发明人给出了基于 POXN 的分布式光交换机。我们将其看作用于光交叉连接 (OXC) 发射器 / 接收器 (TxRx)

的经调度的 1TnR 设计,并且在下文中描述了所形成的 POXN 分布式光交换机 (POXDOS) 如何等同于无阻塞交换机。

[0070] 在 POXDOS800 的 1TnR 实施中,参考图 8,所有发射器 (Tx) 810 可同时工作,发送来自各自连接的数据源 805 的数据,因为各个 Tx 810 在不同的波长 (WL) 上运行。每个接收器 850 在光层接收所有 WL,但是仅能够对一个全速 WL (来自仅一个 Tx) 进行处理。这将峰值接收比特率限制为 1WL 比特率。这样一来,这仅需要 1WL 比特率用于接收数据,发明人认为这等效于交换机,因为所有端口可以同时工作。相应地,每个 Tx 810 被耦合到 POXN 815 的一个端口,以及每个接收器 850 被耦合到 POXN 815 的另一端口,其中,每个接收器 850 的输出端耦合至数据宿 840。

[0071] 在每个接收器 850 内,WDM 820 对来自 NTx 810 的 N 波长进行解复用,将经解复用的 WL 耦合至光接收器 825 的阵列。光接收器 825 的输出端则被耦合到第一电子处理单元 830,其中,对于每个信道,第一电子处理单元 830 仅允许用于数据宿 840 的数据包前进到第二电子处理单元 835,第二电子处理单元 835 将这些允许的数据包按顺序排列到数据宿 840,从而消除了来自在不同 WL 上同时发送的数据包的冲突。相应地,每个接收器 850 可同时接收来自多个发射器的数据。

[0072] 对于本领域的普通技术人员,明显的是,信宿地址冲突可被实施使得在任一时间,对于一个特定的 Rx 端口,仅一个 Tx 被允许发送全速业务包到该 Rx 端口。此外,解决信宿地址冲突的协议也可作为光接收器阵列选择/锁定电子接收器 850。相应地,交换智能和控制可在本发明的实施例中进行分配。

[0073] 在本发明的某些实施例中,所有端口可以诸如 1TnR 的方式结合诸如如上所述的调度协议进行工作以交换业务控制信息。相应地,与数据业务相比,低速业务允许所有接收器在空闲时监听其他端口的状态消息,进而充分利用 POXN 的广播特性。相应地,接收器锁定在协议的光/数字域中,并且 POXDOS 在高速/全速下工作。在输送序列完成后,接收端口切换回空闲模式并且广播其空闲状态。当一个端口与另一端口进行交换时,其他端口可建立其他链路,但不是达到占用的 Rx 端口的链路直到该占用的 Rx 端口被释放。

[0074] 相应地,具有相关联的 Tx/Rx 的一个端口可同时从一个端口接收而向另一端口发送,因而使用调用方式协议来提供扩展的业务控制。根据本发明的另一实施例,所有 N 个接收器可同时运行以递送 $N \times WL_{\text{BITRATE}}$ 的接收速度,从而充分利用了全 POXN 能力。然而,这还要求每个接收器支持相当于 $N \times WL_{\text{BITRATE}}$ 的数字处理能力,造价会很昂贵,因而可选择性地设置在某些 POXDOS 实施中。

[0075] 图 9 描绘了部署在二维超级 POXN (2D Hyper POXN) 900 内根据本发明实施例所述的无源光交叉连接网络 (POXN)。相应地,如所描绘的那样,所述 2D Hyper POXN 900 包括功能单元 920,所述功能单元 920 经由 POXN 910 被耦合至垂直路由 930、水平路由 940 以及下一垂直行的功能单元 920 内的功能单元 920。经由垂直路由 930,功能单元 920 还被连接到同一水平行中的相邻功能单元 920。经由水平路由 940,数据中心还被连接到下一垂直行的功能单元 920 内的下个功能单元 920。相应地,从功能单元 920 提取出的数据可路由到其他功能单元 920 或者分别路由到水平/垂直路由 940 和 930,并且在此被分配到整个 2D Hyper POXN 900。2D Hyper POXN 900 内的边缘元件 (为清楚起见,未示出) 经由外部以及内部连接到 2D Hyper POXN 900 内的元件上。

[0076] 图 10 描绘了部署在二维超级 POXN(2D Hyper POXN)1000 内根据本发明实施例所述的无源光交叉连接网络 (POXN)。对本领域普通技术人员而言,明显的是,如上所述的 POXN 可视为线性总线(等效于同时向该线性总线上的所有节点提供信号)或者可视为具有“一到多”和“一到一”切换功能的光交叉连接。相应地,在图 10 和 2D Hyper POXN 1000 中,POXN 可被描述为线性总线元件,即 POXN A 1010 和 POXN B 1020,其中,从功能单元 1030 到 POXN 的连接为从该功能单元 1030 到该 POXN 的一个端口的连接。相应地,如图 10 和 2D Hyper POXN 1000 所示,每个功能单元 1030 连接至两个 POXN,即分别为 POXN A 1010 和 POXN B 1020。相应地,对本领域普通技术人员而言,明显的是,一个或两个 POXN 可被无源光交叉连接 (POXC) 结构(如上文关于图 5 所描述的结构)所代替。

[0077] 参考图 11,其描绘了部署在三维超级 POXN(3D Hyper POXN)1100 内根据本发明实施例所述的无源光交叉连接网络 (POXN)。正如图 10 所示,对本领域普通技术人员而言,明显的是,POXN 可视为线性总线(同时向该线性总线上的所有节点提供信号)。相应地,在图 11 和 3D Hyper POXN 1100,POXN 可被描述为线性总线元件,即 POXN A 1110、POXN B 1120 和 POXN C 1130。从功能单元(例如,处理器 A 1140,存储器 1150 或者处理器 B 1160)到 POXN 的连接在物理层面为从功能单元到 POXN 的被描述为 POXN 的线性总线表示的端口的连接。相应地,如图 11 和 3D Hyper POXN 1100 所示,3D Hyper POXN 1100 内的每个功能单元被描述为连接到两个 POXN,即分别为 POXN A 1110 和 POXN B 1120,同时 3D Hyper POXN 1100 内处于左上行中的这些功能单元也连接至 POXN C 1130,以使得其被连接到 3 个 POXN,即分别为 POXN A 1110、POXN B 1120 和 POXN C 1130。这样一来,POXN C 1130 起作用以在不同层的功能单元之间进行耦合,不同层的功能单元经由 2D Hyper POXN 进行连接,如图 10 中的 2D Hyper POXN 1000 所描述的那样。这些层在 3D Hyper POXN 1100 内表示为层 1 至层 8。边界 1170 内的那些功能元件被描绘为仅被连接到两个 POXN。可选地,这些功能单元也可连接至第三 POXN C 1170 或者可连接至其他 POXN。

[0078] 相应地,对本领域普通技术人员而言,明显的是,一个、两个或全部三个 POXN 可被无源光交叉连接 (POXC) 结构(如上文关于图 5 所描述的结构)所代替。对本领域的普通技术人员而言,明显的是,这些概念可扩展至一般的 N 维网格 (NDMESH),其中,每个功能单元连接至不同组合中的 POXN 和 / 或 POXC。2D Hyper POXN 1000 和 3D Hyper POXN 1100 形成了被发明人命名为“超级光学云 (HOC)”的离散实施例。

[0079] 上文描述中给出了具体的细节以帮助完全理解实施例。但是,应该理解,本发明的实施例可以不通过这些具体细节而实施。例如,可框图形式给出电路,以使不必要的细节不会模糊这些实施例。在其他示例中,为了避免模糊实施例,没有显示一些公知的电路、过程、算法、结构和技术的非必要细节。

[0080] 可以用各种方式来实施上面描述的技术、框图、步骤和手段。例如,可以用硬件、软件或两者的结合来实现这些技术、框图、步骤和手段。对于硬件实施,可以在一个或多个特定用途集成电路 (ASIC)、数字信号处理器 (DSP)、数字信号处理器件 (DSPD)、可编程逻辑器件 (PLD)、现场可编程门阵列 (FPGA)、处理器、控制器、微控制器、微处理器、设计用于执行上述功能的其他电子单元和 / 或它们的组合中实施处理单元。

[0081] 还应注意,可将这些实施例描述为以流程表、流程图、数据流程图、结构图或框图的形式描绘的过程。尽管流程图可以描述作为连续处理的操作,然而,所述操作中的许多操

作可以并行或同时执行。此外,操作的顺序可以重新设置。当过程的操作完成时,该过程即被终止,但是该过程可能包括未包含在图中的附加步骤。过程可对应于方法、函数、步骤、子例程、子程序等。当过程对应于函数时,过程的终止对应于该函数对调用函数或主函数的返回。

[0082] 此外,实施例可通过硬件、软件、脚本语言、固件、中间件、微码、硬件描述语言和/或以上各项的任意组合来实现。当以软件、固件、中间件、脚本语言和/或微码的形式实施时,用以完成必要任务的程序代码或代码段可存储在机器可读介质中,例如存储介质。代码段或机器的可执行指令可表示步骤、函数、子程序、程序、例程、子例程、模块、软件包、脚本、等级或指令、数据结构和/或程序语句的任意组合。代码段可通过传递和/或接收信息、数据、变元、参数和/或存储器内容而被耦合至另一代码段或硬件电路。信息、变元、参数、数据等可通过任何合适的手段进行传递、转发或发送,该手段包括内存共享、消息传递、令牌传递、网络传输等。

[0083] 对于固件和/或软件实施,可通过模块(例如,步骤、函数等)来实施方法,该模块执行本申请中所述的功能。任何使指令具体化的机器可读介质可用于实施本申请中所述的方法。例如,软件代码可存储在存储器中。存储器可以在处理器内实施或者在处理器外部实施,并且根据实施方式的不同而改变,其中,当存储器被用来执行软件代码时,存储器被用来存储供后续执行的软件代码。正如本文所使用的,术语“存储器”是指任何类型的长期、短期、易失性、非易失性、或其他存储介质,并且不限于存储器的任何特定类型、或存储器的数目或存储存储器的介质类型。

[0084] 此外,本文中的术语“存储介质”可以表示一个或多个用于存储数据的装置,包括只读存储器(ROM)、随机存取存储器(RAM)、磁性RAM、核心存储器、磁盘存储介质、光存储介质、闪存装置和/或其他用于存储信息的计算机可读介质。术语“机器可读介质”包括但不限于便携式或固定存储设备、光学存储设备、无线信道、和/或各种能够存储、包含或承载指令和/或数据的其他介质。

[0085] 在一个或多个实施例中,本申请中所描述的方法可通过机器来执行,该机器包括一个或多个接受含有指令的代码段的处理器。对于本申请中所描述的任一方法,当机器执行指令时,该机器即执行所述方法。任何能够执行一组指令(顺序或其他)的机器都被包括在内,该组指令指定了该机器所要执行的动作。因此,典型的机器示例可为包括一个或多个处理器的典型处理系统。每个处理器可包括一个或多个CPU、图像处理单元和可编程DSP单元。处理系统还可包括存储器子系统,其包括主RAM和/或静态RAM、和/或ROM。可包括总线子系统,用于部件之间的通信。如果处理系统需要显示器,则可包括这样的显示器,例如液晶显示器(LCD)。如果需要手动数据输入,则处理系统还包括输入装置,比如一个或多个字母输入单元(例如键盘)、指向控制装置(例如鼠标)等。

[0086] 存储器包括含有指令的机器可读代码段(例如,软件或软件代码),当被处理系统执行时,所述指令用于实现本申请中所描述的一种或多种方法。软件可整体位于存储器中,或者在由计算机系统执行时完全或至少部分位于RAM内和/或处理器内。因此,存储器和处理器也可组成包括机器可读代码的系统。

[0087] 在替代性实施例中,机器作为单独的装置运行或者可以联网部署的方式连接至(联网)其他机器,机器可在服务器-客户网络环境中以服务器或客户机的身份运行,或者

在对等或分布式网络环境中,以对等机的身份运行。所述机器可以(例如)是计算机、服务器、服务器簇、计算机簇、网页应用、分布式计算环境、云计算环境或者任何能够执行指定该机器所要采取的动作的一组指令(顺序或其他)的机器。术语“机器”也可包括一组机器,这些机器可单独或结合执行一组(或多组)指令以完成本文中所述的一种或多种方法。

[0088] 本发明的示例性实施例的前述公开已经表明了其示意和说明的目的。并不意在穷尽本发明或限制本发明现在所公开的精确形式。根据上述公开,对本领域的普通技术人员而言,本文中所描述的实施例的各种变型和修改将是明显的。本发明的范围仅由所附权利要求及其对等物进行限定。

[0089] 进一步地,在描述本发明的示例性实施例的过程中,说明书可能将本发明的方法和/或过程表示为特定序列的步骤。然而,所述方法或过程不依赖于在此所述的特定步骤顺序,从这个意义上说,所述方法或过程不应限制于所述特定的步骤顺序。正如本领域普通技术人员所理解的那样,其他步骤顺序也是有可能的。因此,说明书中所给出的特定顺序不应被理解为对权利要求进行限制。另外,涉及本发明的方法和/或过程的权利要求不应当局限于按照所写顺序执行其步骤,并且本领域的普通技术人员能够容易理解顺序可能发生变化,并且仍未超出本发明的精神和范围。

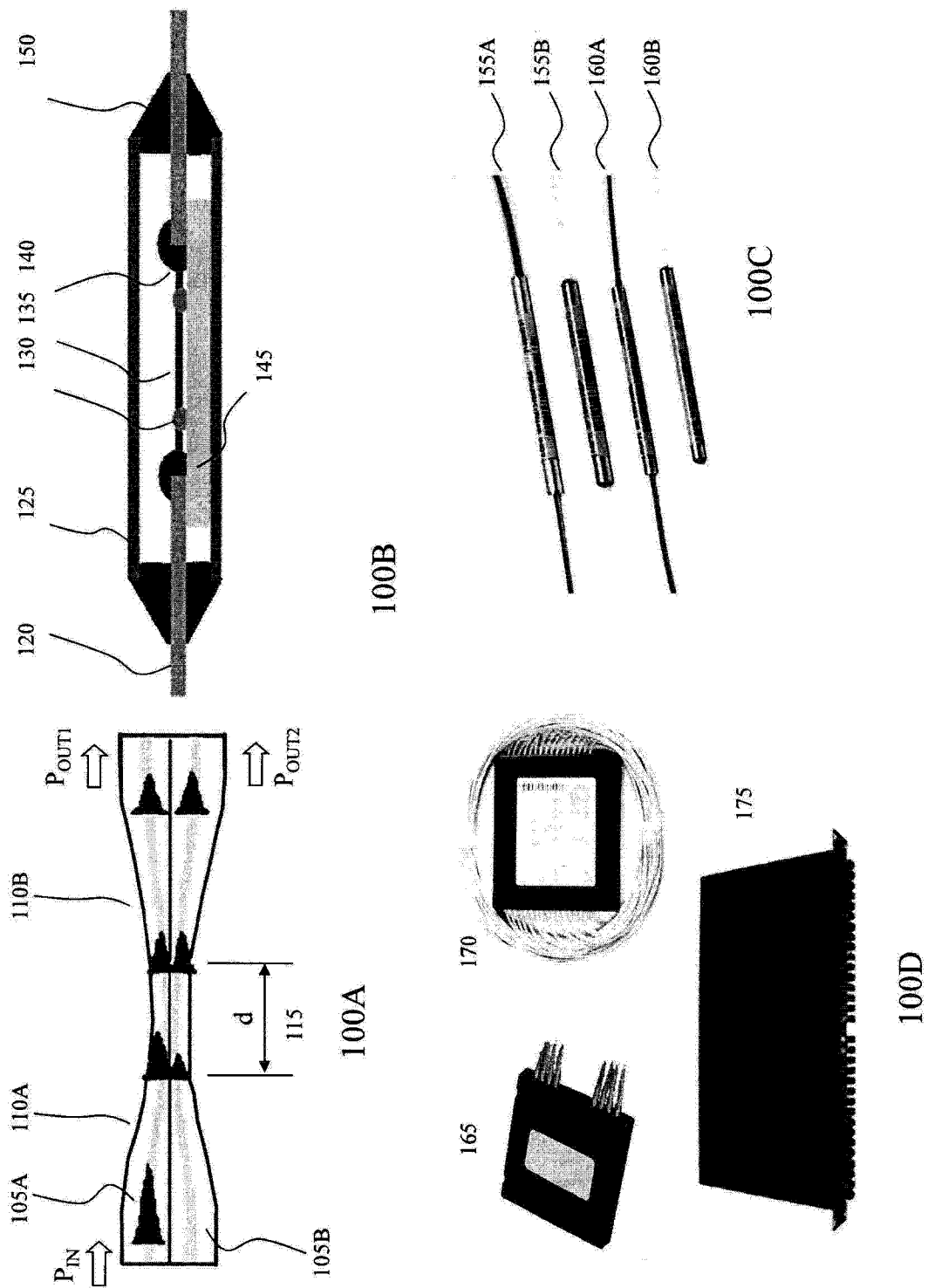


图 1A

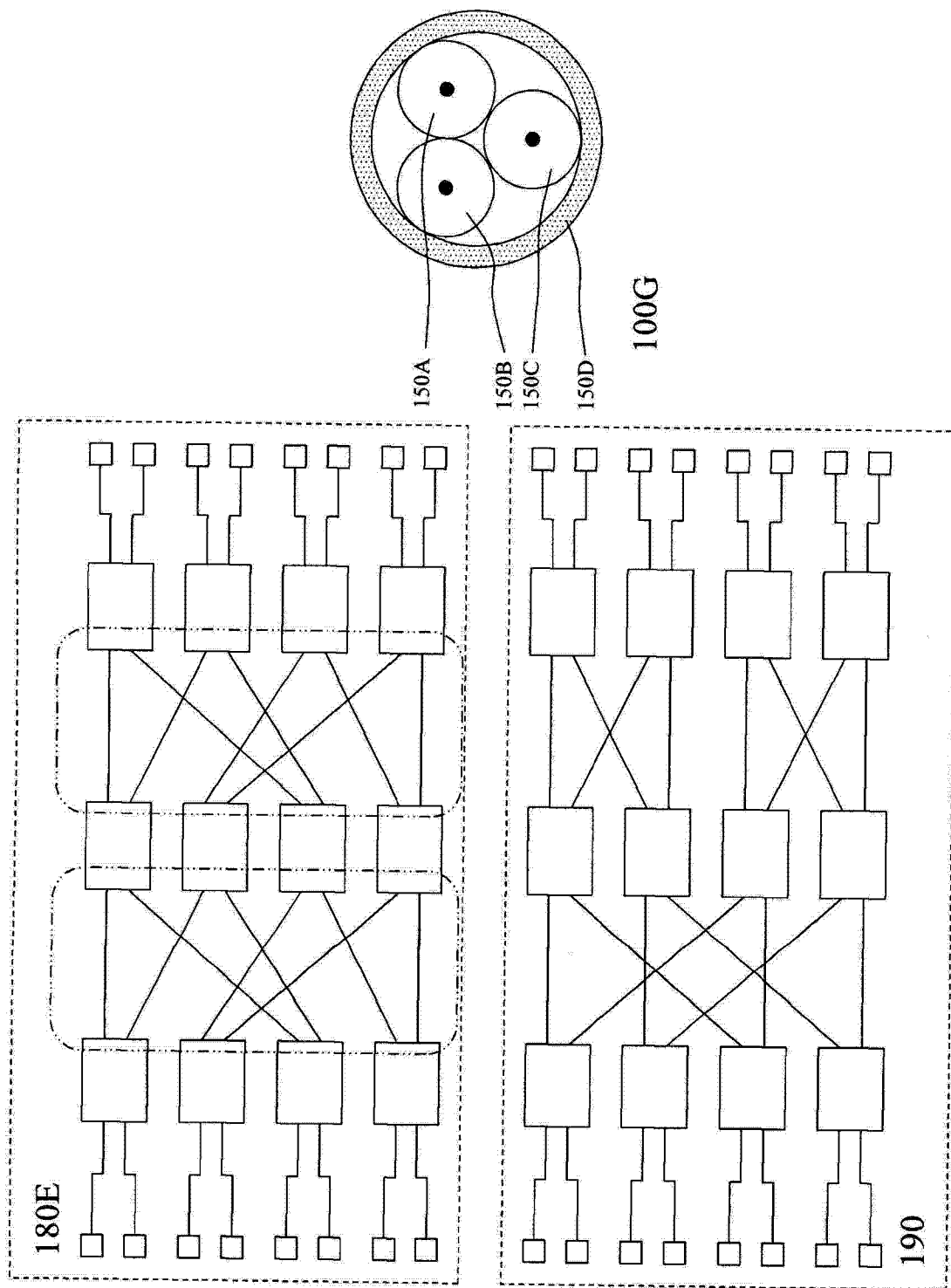


图 1B

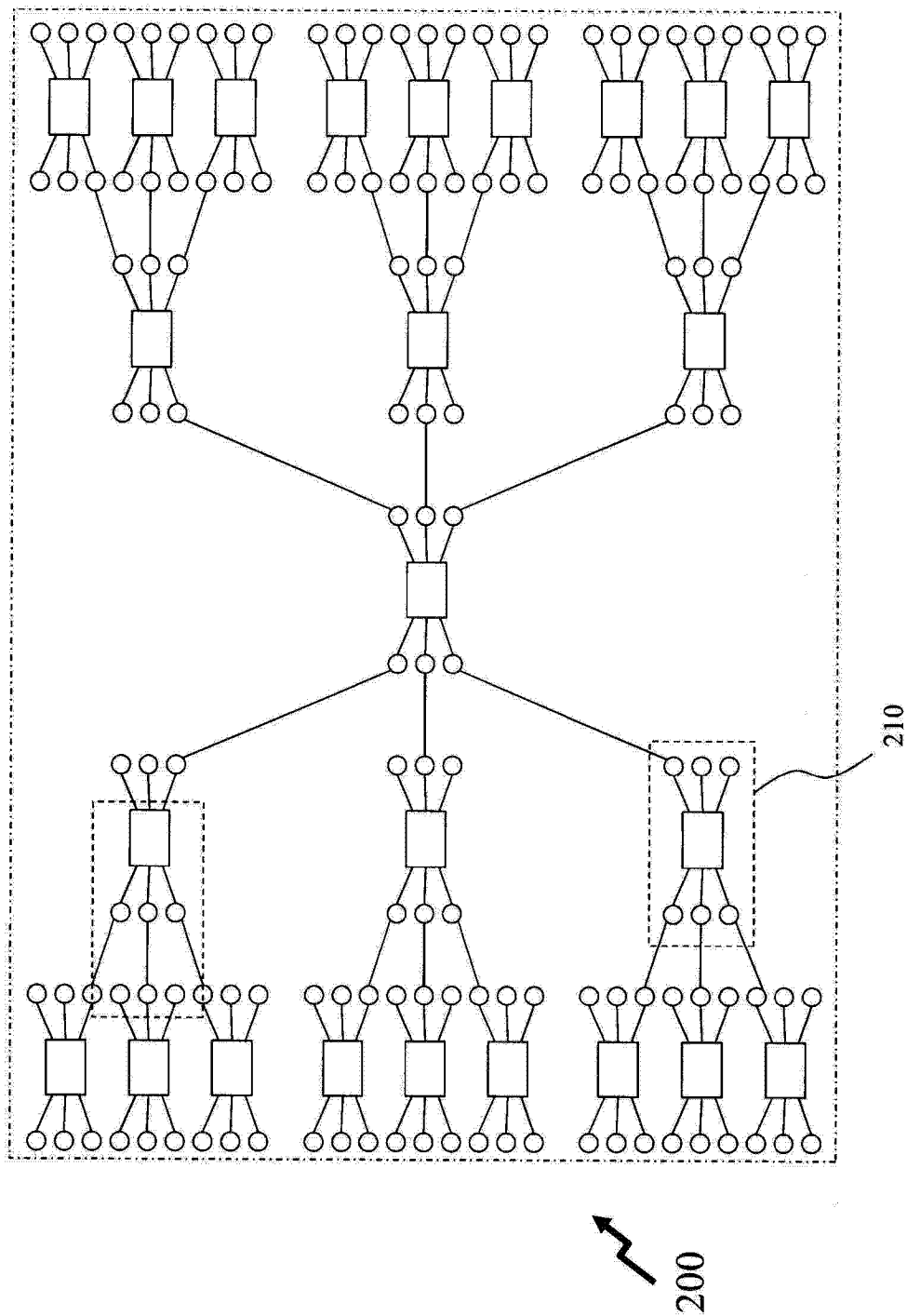


图 2

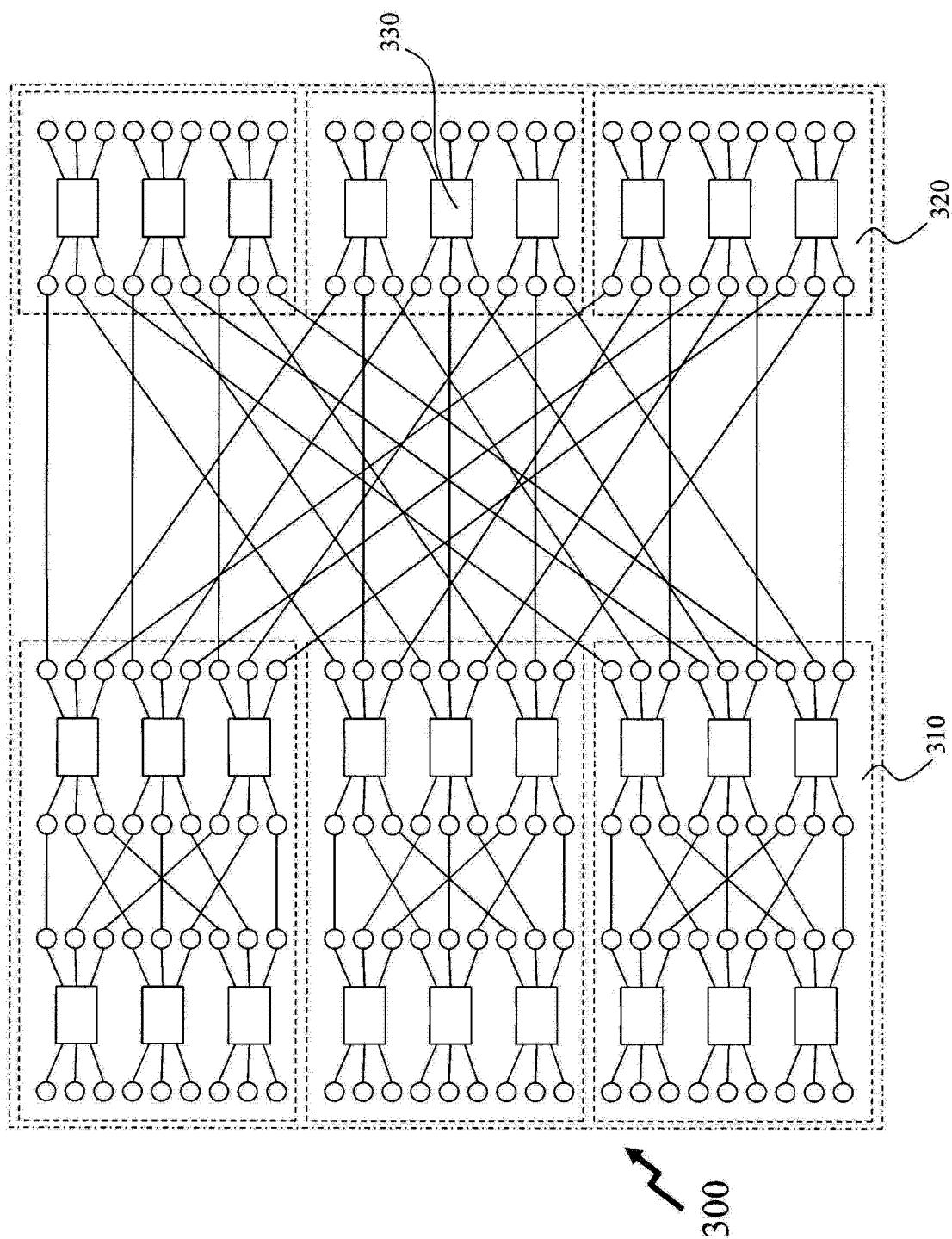


图 3

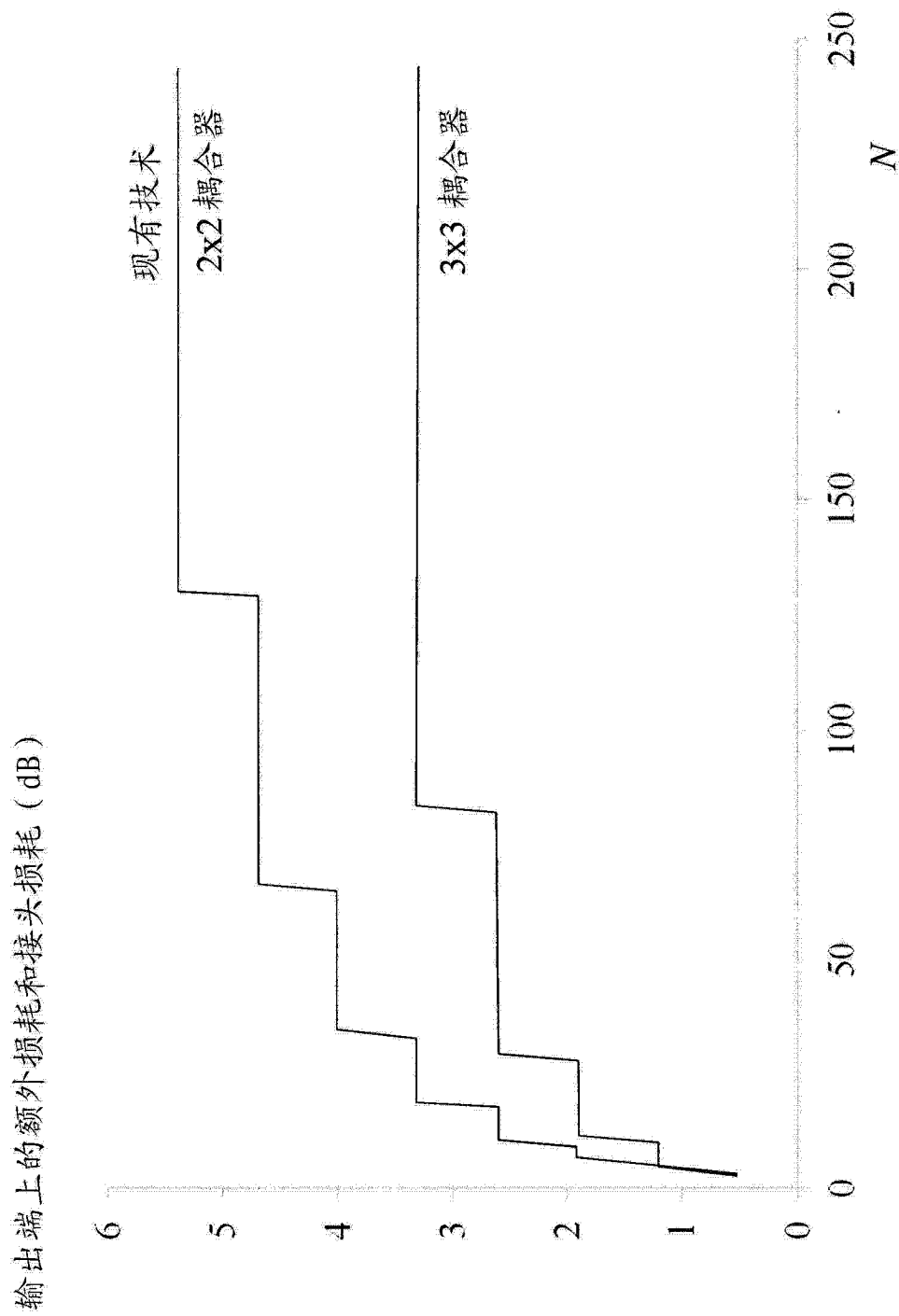


图 4

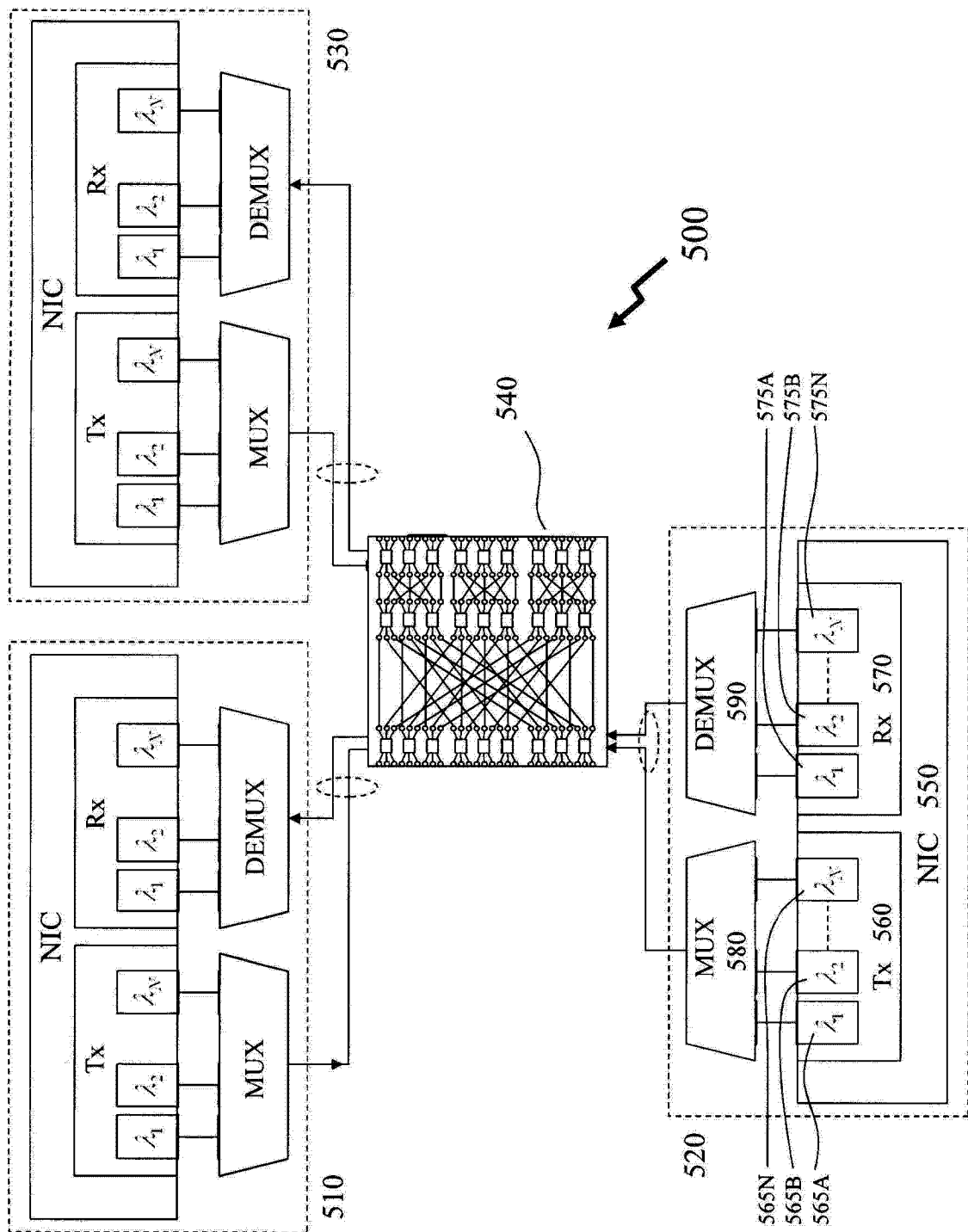


图 5

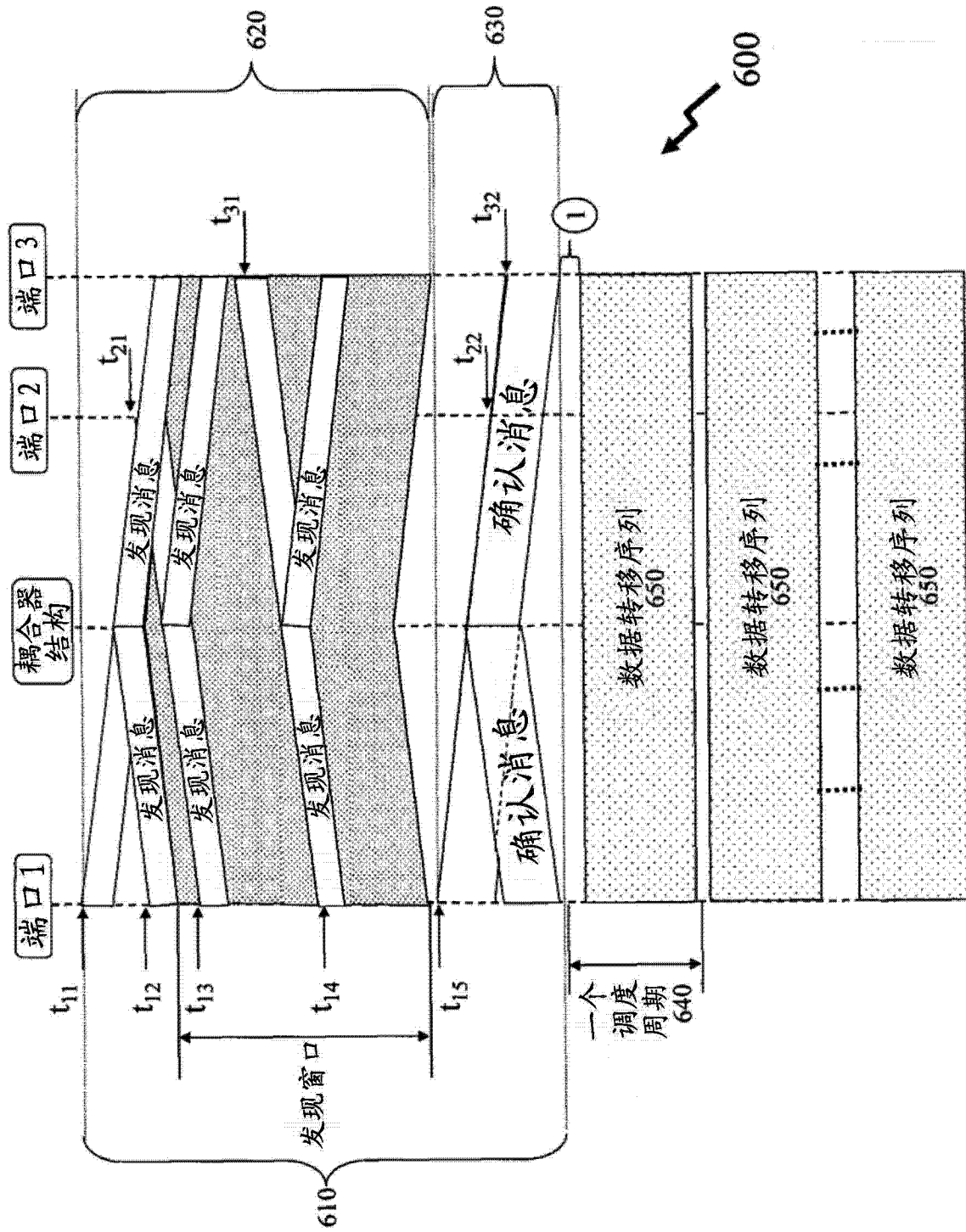


图 6

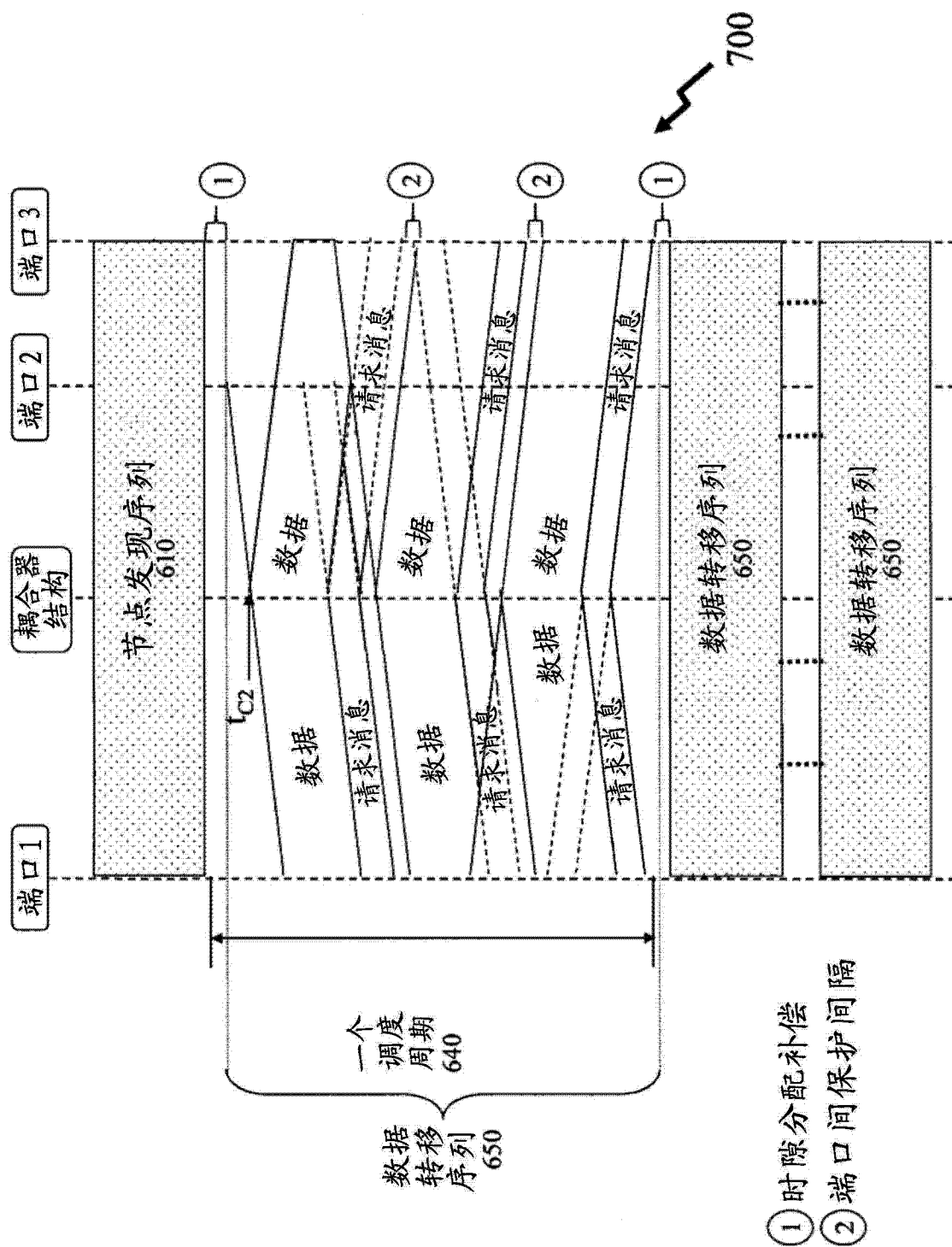


图 7

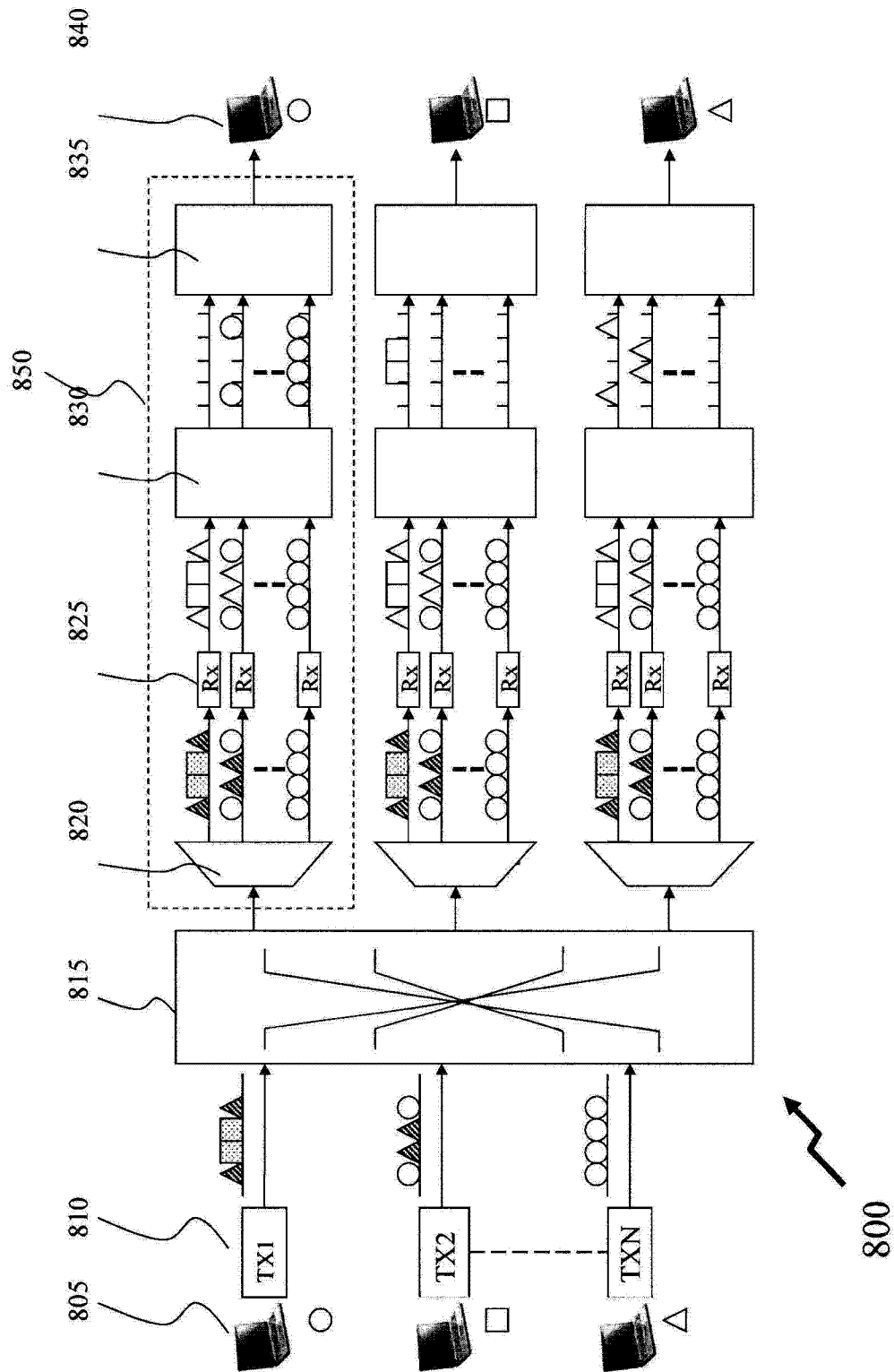


图 8

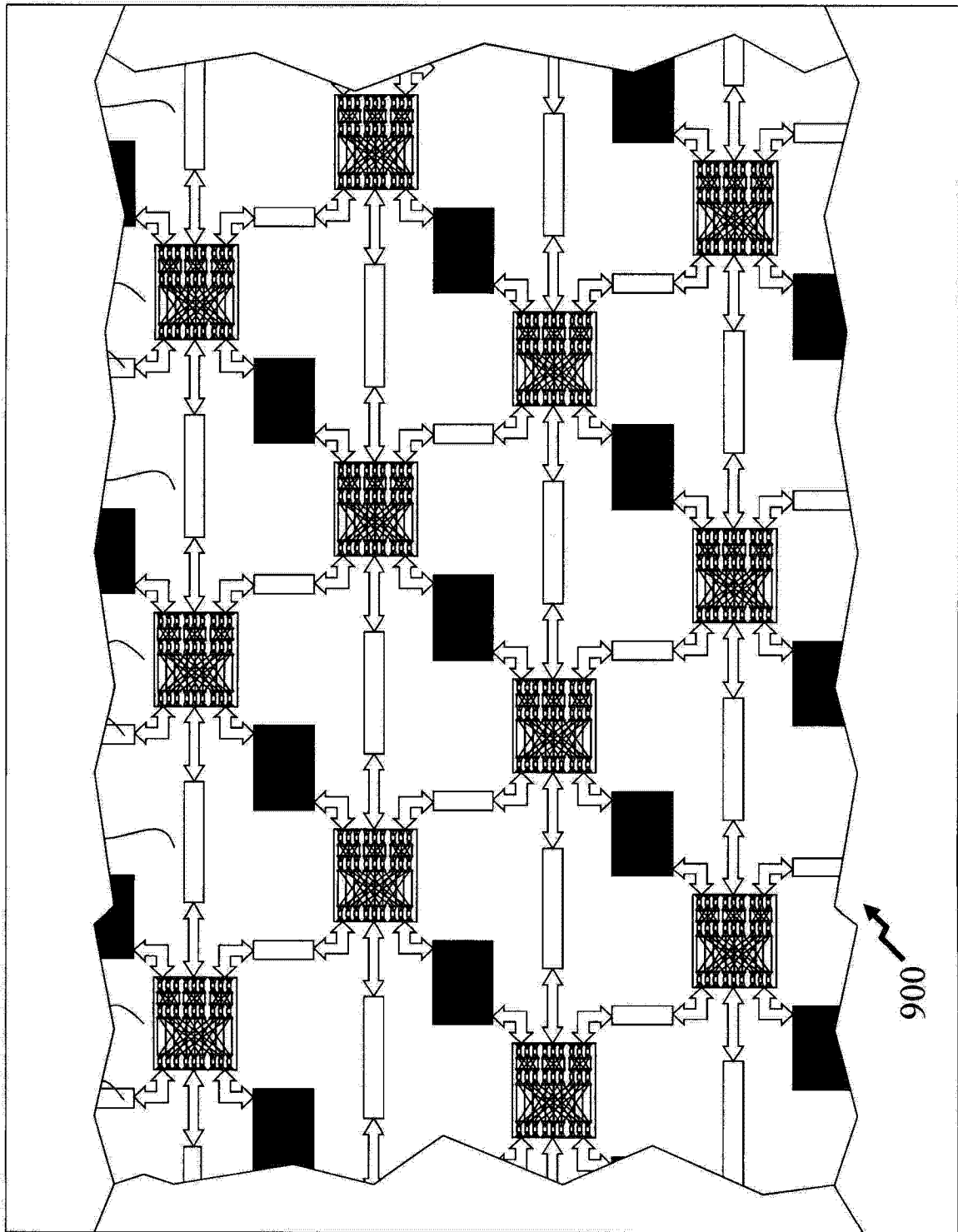


图 9

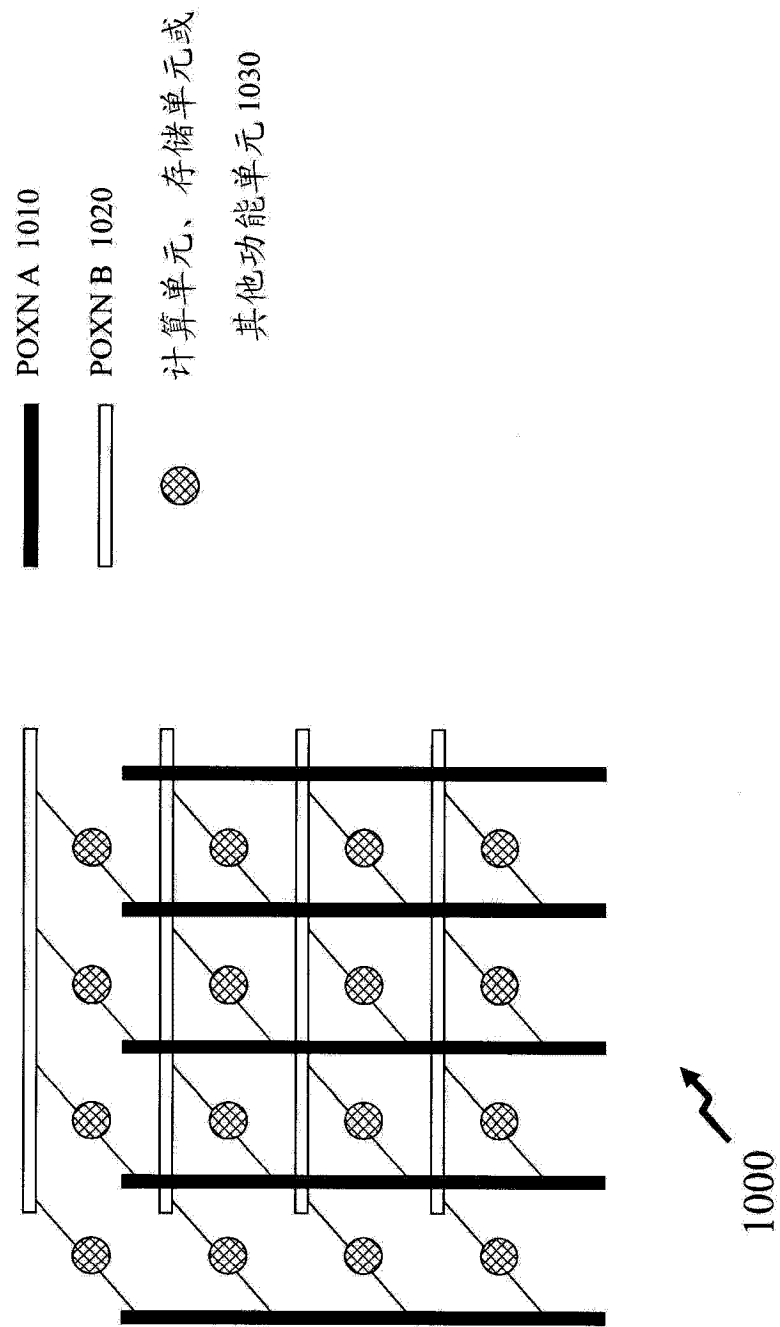


图 10

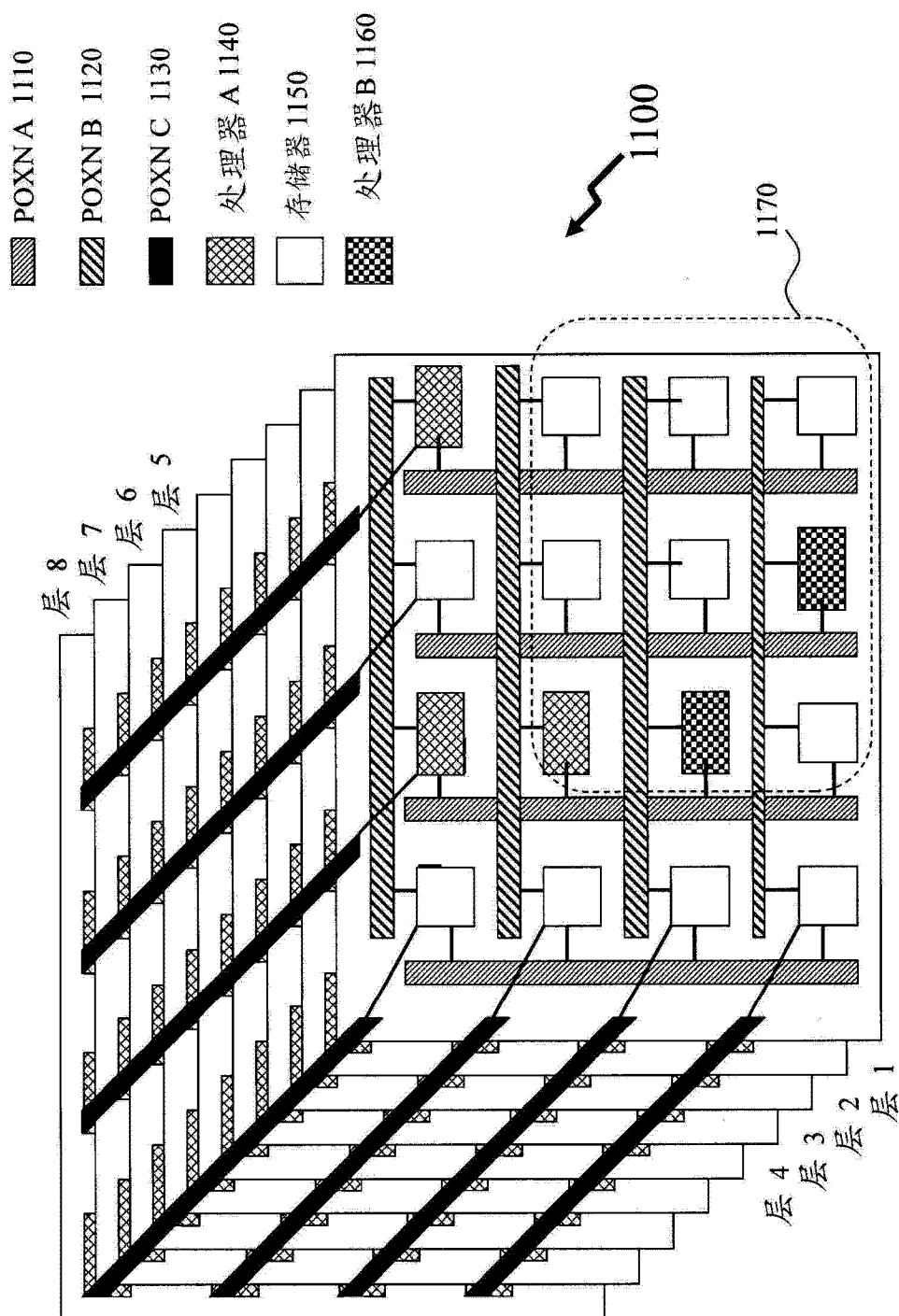


图 11