

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04L 12/26 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200780005965.8

[43] 公开日 2009年3月11日

[11] 公开号 CN 101385276A

[22] 申请日 2007.3.14

[21] 申请号 200780005965.8

[30] 优先权

[32] 2006.3.29 [33] US [31] 11/277,868

[86] 国际申请 PCT/EP2007/052411 2007.3.14

[87] 国际公布 WO2007/110329 英 2007.10.4

[85] 进入国家阶段日期 2008.8.19

[71] 申请人 国际商业机器公司

地址 美国纽约

[72] 发明人 A·阿尔德里古亚 G·A·里克特
J·威廉姆斯

[74] 专利代理机构 中国国际贸易促进委员会专利
商标事务所
代理人 李向英

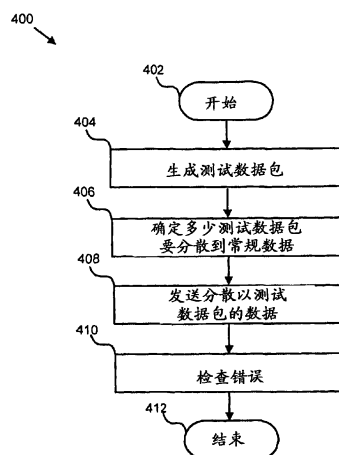
权利要求书2页 说明书15页 附图5页
按照条约第19条的修改1页

[54] 发明名称

用于对通信链路上的错误进行评估的设备、
系统和方法

[57] 摘要

说明了一种用于评估通信链路上的错误的设备、系统和方法。提供了用于评估错误的设备，该设备具有包含被配置为执行下列必要的步骤的多个模块的逻辑单元：创建一个或多个测试数据包，确定通过通信链路向接收节点发送的测试数据包的数量，通过通信链路向接收节点发送多个测试数据包，在多个数据包中分散测试数据包，以便接收节点能够连续地处理数据包，以及检查由接收节点接收到的测试数据包，以发现错误。



1. 一种用于评估通信链路上的错误的方法，所述方法包括：
创建一个或多个测试数据包；
确定通过通信链路向接收节点发送的测试数据包的量；
通过所述通信链路向接收节点发送多个数据包；
通过所述通信链路向所述接收节点发送一个或多个测试数据包；
在所述数据包中分散测试数据包，以便所述接收节点能够连续地处理所述数据包；以及
检查由所述接收节点接收到的数据包，以发现错误。
2. 根据权利要求 1 所述的方法，其中，确定测试数据包的量的过程包括，确定待分配的发送到所述接收节点的数据包的数量的百分比以发送测试数据包。
3. 根据权利要求 1 所述的方法，其中，发送测试数据包的过程包括有选择地向连续的测试数据包分配优先级。
4. 根据权利要求 1 所述的方法，其中，通过所述通信链路发送一个或多个测试数据包的过程包括，发送测试数据包的阈值级别，以模拟所述接收节点的预定错误率。
5. 根据权利要求 1 所述的方法，其中，在所述数据包中分散测试数据包的过程包括，在所述数据包中随机地分散测试数据包。
6. 根据权利要求 1 所述的方法，其中，检查由所述接收节点接收到的所述数据包以发现错误的过程包括，在错误寄存器中记录错误。
7. 根据权利要求 6 所述的方法，其中，检查由所述接收节点接收到的所述数据包以发现错误的过程包括，专门地检查测试数据包以发现错误。
8. 一种包括用于执行根据任何前面的权利要求之一所述的方法的所有步骤的装置的系统。
9. 一种计算机程序，包括用于当所述计算机程序在计算机系统

上执行时执行根据任何前面的权利要求之一所述的方法的所有步骤的指令。

用于对通信链路上的错误进行评估的设备、系统和方法

技术领域

本发明涉及评估通信链路上的错误，具体来说，涉及高速串行链路的实时可扩展性端口评估。

背景技术

在当今的工商界，产生了大量的信息。此信息常常对企业的成功是不可缺少的，可能是企业的最宝贵的商品。信息必须被处理、存储、更新，并由企业的运营单位共享。例如，制造商必须能够与其销售部、其支持人员、其供应商、其经销商，以及各种其他实体共享信息。结果，许多企业利用可能集中地放置的信息存储系统，与该企业有联系的所有人都可以访问。企业信息系统 (EIS) 就是一个这样的信息系统。企业可以具有访问一个或多个 EIS 的多个子实体或“客户端”。

企业可以利用一个处理器来管理其信息。然而，更加可能的是，企业具有链接在一起的多个处理器，以提高性能和计算能力。许多系统将多个处理器和数据存储单元组合为各种配置，包括共享存储器系统或分布式存储器系统。可以组织处理器的集合，以便每一个处理器都可以读取连接到不同的处理器的存储器。这种处理器的组织常常叫做“群集”。每一个处理器都可以直接连接到与这种多处理器组织中的处理器相连接的每一个存储器单元，并可以直接访问数据。常常，有四个或更多个处理器以及它们的存储器单元被以共享存储器配置组织在叫做“节点”的单一盒子中。然后，可以将节点连接在一起。

无论什么布局，处理器、群集或节点最有可能通过通信链路彼此连接在一起，而通信链路通常是插入到各种端口中的电缆。电缆常常是插入到串行端口中的高速互连电缆。随着处理器和存储器单元的布局变得更加复杂，在这些通信链路上产生数据传输错误的概率增大。

如此，常常对连接多个处理器或系统的各种端口或链路进行测试，以评估数据传输中的错误和性能。

通信链路的性能会受到许多不同因素的影响，包括电压、电路板阻抗、温度、电缆长度、连接器等等。如果端口或链路完全损坏，或电缆断开连接，则连接会停止工作。然而，也有电缆可能没有被正确地放置或部分地损坏，连接只是性能下降而没有中断的情况。

大多数系统是通过向接收节点或端口发送大量的数据来测试错误的。该节点或端口又发出包含大量的待通过计数器检查的数据的确认。然后，可以计算出性能错误。这种方法以及大多数其他方法所存在的问题是，测试垄断了该链路，禁止了跨该链路的所有其他通信流。此会妨碍节点或端口被正确地使用或测试。

从前面的讨论可以看出，显然需要一种改进的评估通信链路上的错误的设备、系统和方法。优选情况下，这样的设备、系统以及方法允许对一个正在被使用的端口进行端口错误评估而不必禁用该端口。这样的设备、系统，以及方法也将允许在进行端口错误评估的同时，在测试正在进行时维持正在运行的系统的特征，如高数据速率和随机数据发送功能。这里说明了这样的设备、系统，以及方法，并要求保护其权利。

发明内容

响应当前的技术状况，具体来说，响应当前技术中的还没有被当前可用的错误评估设备、系统以及方法解决的问题和需求，开发了本发明。相应地，本发明是为了提供克服了当前技术中的许多上文所讨论的缺点或全部缺点的用于评估通信链路上的错误的设备、系统、方法而开发的。

用于评估通信链路上的错误的设备具有被配置为在功能上执行下列必要的步骤的多个模块：创建一个或多个测试数据包，确定通过通信链路向接收节点发送的测试数据包的量，通过通信链路向接收节点发送多个测试数据包，在多个数据包中分散测试数据包，以便接收

节点能够连续地处理数据包，以及检查由接收节点接收到的测试数据包，以发现错误。在整个本说明书中，对通信链路的引用也可以包括接收节点，反之亦然。

所描述的实施例中的这些模块包括创建模块、确定模块、发送模块、分散模块，以及检查模块。在一个实施例中，该设备被配置为创建测试数据包，确定通过通信链路向接收节点发送的测试数据包的量，以便测试该通信链路和/或接收节点，以发现错误。该设备还可以被配置为以不禁用接收节点的速率发送与正常的数据包分散在一起的多个测试数据包。该设备检查接收到的测试数据包，以发现错误。

在一个实施例中，该设备进一步被配置为有选择地向测试数据包分配优先级。可以一直发送测试数据包，直到到达测试数据包的预定的阈值。在一个实施例中，此阈值是确定的，以便发送足够的测试数据包以模拟接收节点的预定的错误率。在一个实施例中，该设备将测试数据包错误记录到错误寄存器中。

本发明还提供了用于评估通信链路上的错误的系统。该系统可以作为节点的群集或以多个计算设备配置的方式来实现。在一个实施例中，该系统是作为一个或多个可扩展的企业服务器来实现的。

该系统可以包括至少一个中央处理单元 (CPU) 和处理器。该系统还可以包括高速缓存和高速缓存控制器。该系统包括用于连接到通信链路的至少一个端口。存储器可以包含用于在处理器上执行的模块。在一个实施例中，模块包括构成了上文所描述的的设备的那些模块。该系统可以包括用于创建一个或多个测试数据包的创建模块，用于通过通信链路向接收节点发送多个测试数据包的发送模块，以及分散模块，用于在多个数据包中分散测试数据包，以便接收节点能够连续地处理数据包。在一个实施例中，存储器包括检查模块，用于检查由接收节点接收到的测试数据包，以发现错误。总线可操作地连接了至少一个 CPU、处理器、高速缓存、高速缓存控制器、端口，以及存储器。

本发明还提供了用于评估通信链路上的错误的方法。所说明的实

施例中的方法基本上包括执行上文对于所描述的设备 and 系统的操作所提供的功能所需的步骤。在一个实施例中，该方法包括创建一个或多个测试数据包，以及确定通过通信链路向接收节点发送的测试数据包的量。该方法还可以包括通过通信链路向接收节点发送多个数据包，以及通过通信链路向接收节点发送一个或多个测试数据包。在一个实施例中，测试数据包被以不禁用接收节点的速率分散在数据包中。检查测试数据包，以发现错误。这里所说明的方法步骤还可以作为由计算机程序产品的计算机可读取的代码执行的操作来实现。

在整个说明书中对功能、优点的引用或类似的语言并不意味着，可以利用本发明实现的所有功能和优点应该是或位于本发明的任何单一实施例中。相反地，引用了功能和优点的语言被理解意味着，与实施例一起描述的特定功能、优点或特征包括在本发明的至少一个实施例中。如此，整个说明书中的对功能和优点的讨论，以及类似的语言可以，但不一定是指相同实施例。

此外，在一个或多个实施例中，本发明的所描述的功能、优点和特征可以以任何合适的方式来组合。那些本领域的普通技术人员将认识到，本发明可以在没有特定实施例的一个或多个特定功能或优点的情况下来实现。在其他情况下，在某些实施例中可以有其他功能和优点，不一定存在于本发明的所有实施例中。

通过下面的描述和所附的权利要求，本发明的这些特点和优点将变得显而易见，或者，如下面所描述的，也可以通过本发明的实施例的实践来了解。

附图说明

为了轻松地理解本发明的优点，将通过参考附图中显示的特定实施例，提供上文简要描述的本发明的比较具体的描述。可以理解这些附图只描述了本发明的典型的实施例，因此，不被视为对其范围的限制，将通过使用附图并利用额外的特征和细节描述和说明本发明，在附图中：

图 1 是显示了用于自动地检测电缆配置的系统的一个实施例的示意方框图;

图 2 包括了一对企业服务器的本发明的系统的一个实施例的后视图;

图 3 是显示了用于评估通信链路上的错误的设备的一个实施例的示意方框图;

图 4 是显示了用于评估通信链路上的错误的方法的一个实施例的示意图;

图 5 是显示了用于评估通信链路上的错误的方法的一个实施例的示意图。

具体实施方式

本说明书中描述的许多功能单元被标记为模块,以便特别强调它们的实现方式的独立性。例如,模块可以作为包括自定义 VLSI 电路或门阵列、诸如逻辑芯片之类的现成半导体、晶体管或其他分立元件的硬件电路来实现。模块还可以以诸如现场可编程门阵列、可编程阵列逻辑、可编程逻辑设备之类的可编程硬件设备来实现。

模块还可以以用于供各种处理器来执行的软件来实现。例如,可执行代码的标识的模块可以包括计算机指令的一个或多个物理或逻辑块,指令可以作为对象、过程、函数来组织。尽管如此,标识的模块的可执行代码不必在物理上位于一起,而是可以包括存储在不同位置的不同的指令,当这些指令在逻辑上联合在一起时,包括模块,并实现该模块的指定目的。

实际上,可执行代码的模块可以是单个指令、或许多指令,甚至可以在多个不同代码段、在不同的程序中,以及跨多个存储设备地分布。同样,这里可以在模块内标识和显示操作数据,并可以以任何合适的形式来实现并在任何合适的数据结构类型内组织。操作数据可以作为单个数据集来收集,也可以在不同的位置上分布,包括在不同的存储设备上分布,可以至少部分地,只作为系统或网络上的电子信号

存在。

在本说明书中对“一个实施例”或“实施例”的引用，或类似的语言，意味着，涉及实施例所描述的特定功能、结构或特征包括在本发明的至少一个实施例中。如此，在本说明书中出现的短语“在一个实施例中”或“在实施例中”以及类似的语言可以，但不一定都是指同一个实施例。

对信号承载介质的引用可以呈现能够生成信号、使信号生成，或使数字处理设备上的机器可读取的指令的程序执行的任何形式。信号承载介质可以通过传输线路、光盘、数字视盘、磁带、伯努利驱动器、磁盘、穿孔卡片、快闪存储器、集成电路或其他数字处理设备存储设备来实现。

此外，在一个或多个实施例中，所描述的本发明的功能、结构或特征可以以任何合适的方式来组合。在下面的描述中，提供了许多具体细节，如编程、软件模块、用户选择、网络事务、数据库查询、数据库结构、硬件模块、硬件电路、硬件芯片等等的示例，以提供对本发明的实施例的全面的了解。然而，那些本领域技术人员将认识到，本发明可以在没有在一个或多个具体细节的情况下来实现，或利用其他方法、组件、材料等等来实现。在其他情况下，没有显示或详细描述已知的结构、材料、或操作，以避免模糊本发明的某些方面。

图 1 显示了根据本发明的用于评估通信链路上的错误的系统 100 的一个实施例的示意方框图。在一个实施例中，系统 100 是可扩展的企业服务器。系统 100 包括至少一个中央处理单元 102。在一个实施例中，系统 100 包括紧密封装在一起以形成一个节点的四个 CPU 102。系统 100 包括处理器 104。在一个实施例中，处理器是 2.8 GHz 处理器。在其他实施例中，处理器是 3.0 GHz 处理器。系统 100 也包括高速缓存 106 和高速缓存控制器 108。高速缓存可以是 2 MB L3 高速缓存。在一个实施例中，高速缓存是 XceL4 Server Accelerator Cache。在一个实施例中，控制器 108 包括或被配置为 SMP 扩展控制器 109。利用 SMP 扩展控制器 109，可以在多个系统之间进行高速通信。本领域技术人员认识到，可以使用各种

处理器和硬件配置来实施本发明的原理。在一个实施例中，该系统是可升级的，以包括四路、八路、十六路和三十二路处理。

系统 100 包括存储器 110 和存储器控制器 112。在一个实施例中，存储器是具有存储器镜像、热交换，以及热添加功能的主动式存储器。在一个实施例中，存储器 110 包括 2 GB 的 DDR 存储器，每八路系统，都可扩展到 64 GB。本领域技术人员认识到，可以扩展存储器，以符合系统的大小。

在一个实施例中，存储器包括错误评估器 114。如下面比较详细地讨论的，存储器控制器包括构成错误评估器的用于在处理器上执行的模块。在一个实施例中，模块或错误评估器构成了 SMP 扩展控制器 109 的一部分。在一个实施例中，存储器控制器包括用于创建一个或多个测试数据包的创建模块。它还可以包括用于确定通过通信链路向接收节点发送的测试数据包的量的确定模块。存储器控制器 112 还可以用于通过通信链路向接收节点发送多个测试数据包的发送模块。分散模块可以将测试数据包分散在多个数据包中，以便接收节点能够连续地处理数据包。检查模块可以检查由接收节点接收到的测试数据包，以发现错误。本领域技术人员应认识到，错误评估器可以驻留在存储器 110 中。

系统 100 包括用于连接到通信链路的至少一个端口 116。在一个实施例中，系统 100 包括多个扩展端口 118 和外部端口 120。在一个实施例中，端口 116 包括通过 PCI 桥 124 进行连接的外围组件互连 (PCI) 插槽 122。端口 116 还可以是高速串行端口。如这里所使用的，术语“端口”也可以是节点或接收节点。

系统 100 可以包括通过 PCI 桥 124 与处理器进行通信的小型计算机系统接口 (SCSI) 126。在一个实施例中，SCSI 126 可以是具有集成的独立磁盘冗余阵列 (RAID) 并被其支持的双通道 SCSI。系统 100 还可以包括以太网 128 连接。在一个实施例中，以太网 128 连接是集成的双吉比特以太网 128。

系统 100 还可以具有视频 130、通用串行总线 (USB) 132、键盘和鼠标 134，以及远程监控程序适配器 (RSA) 136 的连接。系统

100 包括用于可操作地连接至少一个 CPU 102、处理器 104、高速缓存 106、高速缓存控制器 108、端口 116、存储器 110、存储器控制器 112 的总线 138。在一个实施例中，总线 138 可操作地连接 SCSI 126、以太网 128、视频 130、USB 132、键盘/鼠标 134，以及 RSA 136。

本领域技术人员认识到，所显示的系统 100 可以包括 SMP 扩展模块或中心电子综合体 (CPC)。在一个实施例中，如下面比较详细地讨论的，两个 SMP 扩展模块可以使用两个扩展端口连接在一起，以形成单一节点八路配置。然而，PCI 桥 124 也具有双向远程扩展 I/O 端口 (RXE 端口)，用于连接到远程配置管理器 (参见图 2)。

现在请参看图 2，系统 200 的实施例包括多个可扩展的企业服务器。本领域技术人员应认识到，可扩展性可以保证系统升级和扩展，以便提高容量、功能和性能。在此示例中，系统 200 包括第一服务器 202、第二服务器 204，以及远程配置管理器 206。第一服务器 202 和第二服务器 204 使用高速率数据电缆 210 连接在一起。另外，第一服务器 202 和第二服务器 204 使用诸如以太网电缆 208 和高速率数据电缆 210 之类的离线式通信电缆连接到远程配置管理器 206。配置管理器 206 可以生成第一服务器 202 和第二服务器 204 之间的连接的连接拓扑图。另外，配置管理器 206 还可以管理第一服务器 202 和第二服务器 204 之间的通信仲裁。本领域技术人员应认识到，错误评估器 116 可以位于一个或多个服务器 202、204 中，或位于远程配置管理器 206 中。

相应地，图 1 显示了基本系统 100 的实施例。通过使用各种链路 210，可以将图 1 中的系统连接到另一个系统，这给系统提供了可扩展性。在图 2 中所显示的实施例中，图 1 中的系统通过高速链路，与另一个四 CPU 存储器和 PCI 插槽连接在一起。本领域技术人员应认识到，系统 200 可以包括连接在一起的任意数量的系统。例如，两个系统可以连接在一起，看起来像“8 路”系统。在一个实施例中，四个系统可以连接在一起，看起来像“16 路”系统。在另一个实施例中，系统和/或节点可以被配置为，看起来像“32 路”系统等等。

这些“多节点”系统可以通过高速串行链路连接在一起。如图 2 所示, 底盘 202 和 204 可以通过 SMP 扩展连接而连接在一起。远程配置管理器 206 或“RXE 扩展连接”常常用于提供更多输入/输出功能。在一个实施例中, 底盘 202 和 204 是多节点系统的示例。在图 2 中所显示的实施例中, 每一个底盘 202 和 204 都在单一底盘中包括最多八个处理器。

现在请参看图 3, 该图显示了用于评估通信链路上的错误的设备 300。该设备包括用于创建一个或多个测试数据包的创建模块 302。在一个实施例中, 每一个测试数据包都容纳两千字节的数据。本领域技术人员应认识到, 创建模块 302 可以创建各种大小和长度的测试数据包。测试数据包被配置为通过通信链路发送到接收节点。测试数据包可以包括一个或多个字节的标头信息, 并可以包括标头或页脚, 以有助于在通信链路的另一侧检查测试数据包中的任何错误。通信链路可以包括, 但不限于, 串行端口、连接点等等。测试数据包可以包括预定的数据, 可以保证当在通信链路的另一端接收到数据包时进行更好的比较。

设备 300 也包括确定模块 304, 用于确定通过通信链路向接收节点发送的测试数据包的量, 以便评估通信链路上的错误。在一个实施例中, 测试数据包的量是发送到接收节点的全部数据包的百分比。例如, 设备 300 可以通过通信链路发送数据包的集合, 包括大致 1% 的测试数据包和 99% 的常规数据包。在另一个实施例中, 测试数据包可以包括最多大约发送的数据包的 90%。

确定模块 304 可以基于预定时间阈值确定要发送的测试数据包的量。例如, 设备 300 可以在一个或多个预定的时间量内发送测试数据包。在另一个实施例中, 发送的测试数据包的数量可以包括预定数量。在另一个实施例中, 发送的测试数据包的量近似于接收节点的预定的错误率。如果接收节点具有工业标准的错误率, 则确定模块 304 可以确定发送足够的测试数据包以近似于已知的错误率。例如, 如果特定接收节点的错误率是每 10^9 字节的数据 1 个错误, 则确定模块可以发送 10^9 字节的测试数据包, 以便对接收节点中的错误情

况获得较佳的了解。本领域技术人员应认识到，为近似于接收节点的低错误率，必须通过通信链路向接收节点发送足够的数据。

在一个实施例中，确定模块 304 监视通信链路上的通信量，以确定要通过通信链路发送多少测试数据包。例如，确定模块 304 可以分析许多作业位于通过通信链路发送到接收节点的队列中。确定模块 304 还可以监视接收节点的性能，并增大发送的数据包的数量，直到接收节点的性能减到某一水平。如此，确定模块 304 可以确定发送足够的测试数据包，以便充分地评估在通信链路上发生的错误，而同时不禁用通信链路。

设备 300 包括用于通过通信链路向接收节点发送多个测试数据包的发送模块 306。在一个实施例中，发送模块 306 被配置为通过通信链路向接收节点发送一个或多个数据包。发送模块 306 可以包括适当的标头及发送测试数据包数据和/或常规数据所需的其他信息。在一个实施例中，通过编程逻辑执行发送模块的功能。在另一个实施例中，发送模块的功能是通过系统硬件来执行的。

发送模块 306 向每一个测试数据包分配优先级。在一个实施例中，向不同的测试数据包分配不同的优先级。可以相对于发送测试数据包的顺序，给测试数据包分配各种优先级。测试数据包被标记为测试数据包，并发送到通信链路（在一个实施例中，是可扩展性/RIO 链路）的接收节点。在一个实施例中，测试数据包是高优先级数据包。发送模块 306 可以向连续的测试数据包分配相同或不同的优先级。然后，可以利用优先级仲裁发送测试数据包。例如，可以利用具有公平仲裁的优先级通过通信链路发送测试数据包。本领域技术人员应认识到，可以利用许多优先级和仲裁方案来发送其中分散了测试数据包的数据包。

可以连续地发送测试数据包，以保证在通信链路上数据包之间的交互。在一个实施例中，发送模块 306 发送测试数据包的阈值级别，以模拟接收节点的预定的错误率。如上文所讨论的，阈值可以基于特定通信链路或接收节点的已知平均错误率。发送模块 306 可以持续发送测试数据包，直到到达预定的阈值。发送模块 306 还可以调节

发送测试数据包的速率，以便防止发送太多的测试数据包，而太多的测试数据包可能会禁用通信链路。发送模块 306 能够发送测试数据包和常规数据包两种。

设备 300 包括分散模块 308，用于将测试数据包分散在多个数据包中，以便接收节点能够连续地处理数据包。当通信链路没有被禁用时，接收节点能够连续地对数据包进行处理。本领域技术人员应认识到，一次发送太多的测试数据包可能会导致通信链路和/或接收节点对常规数据流量变得禁用。在这些情况下，接收节点不能连续地处理常规数据包。分散模块 308 以允许接收节点和/或通信链路持续处理常规数据包的速率将测试数据包分散在常规数据包数据流中。

可以通过在常规数据包流的开始、结束或中间发送测试数据包，来分散测试数据包。可以通过在常规数据包流的开始、结束或中间发送一个或多个测试数据包的组，来分散测试数据包。在一个实施例中，分散模块 308 随机地在数据包或数据包流中分散测试数据包。在一个实施例中，分散过程是通过软件来完成的。在另一个实施例中，利用硬件的随机数发生器来将测试数据包随机地分散到数据包流中。

设备 300 还包括检查模块 310，用于监视或检查由接收节点接收到的测试数据包，以发现错误。在一个实施例中，检查模块 310 只检查测试数据包以发现错误。在一个实施例中，检查模块 310 使用接收节点中的错误计数器。错误计数器被设置为只存储在测试过程中来自测试数据包的信息，如此，收集有关接收到的数据包的数量信息，以及带有错误的数据包的数量。本领域技术人员应认识到，这可使系统对错误率进行计算，而同时系统还可进行正常的通信。在一个实施例中，检查模块 310 基于已知的测试数据包大小来检查错误。

在一个实施例中，检查模块 310 使用循环冗余码 (CRC) 检查，来检查通信链路上的错误。检查模块 310 可以对接收节点之间的通信链路上的所有数据传输使用多字节 CRC 错误检查协议。发送系统可以计算 CRC，并将它附加到发送的数据中。接收系统将基于全部的消息——包括附加的 CRC 字节，来计算新的 CRC。

在另一个实施例中，检查模块 310 通过使用总线奇偶校验来检

查通信链路上的错误，验证在数据传输操作过程中数据没有被改变。检查模块 310 还可以使用信道检查报告和错误日志记录，来检测硬件错误并记录错误。本领域技术人员应认识到，检查模块 310 可以使用其他方法来检查特定数据流中的错误。

设备 300 还可以包括用于选择一个或多个测试数据包的优先级的寄存器。寄存器可以在设备或系统芯片集内定义，以允许寄存器确定测试数据包利用率。在一个实施例中，寄存器是可编程的。寄存器可以对发送测试数据包的速率进行编程。例如，可以利用 0 和 10 之间的值对寄存器进行编程，以便分别提供 0 和 100% 的测试利用率。例如，寄存器可以规定，每十个数据包中的“n”个可以是测试数据包。剩余的 10-n 个数据包是常规的或非测试数据流量。

寄存器可以是错误寄存器，检查模块 310 可以在错误寄存器中记录测试数据包错误。在一个实施例中，设备包括多个寄存器。寄存器可以被编程为只记录测试数据包，并指出哪些测试数据包有错误。在一个实施例中，在错误寄存器中没有收集有关在任何测试阶段的常规数据流量的信息。

如此，系统允许错误评估器设备 300 测定通过特定链路传输的数据量。系统可以允许在短时间量内发送大量的数据，或在长时间内发送少量数据，而不会堵塞链路。这可以保证系统或用户在系统仍在运行时进行诊断。相应地，系统允许在不关闭系统的情况下测试链路。它可在系统仍在运行时对多节点链路进行诊断。

本领域技术人员应认识到，各种模块的某些功能可以重叠或由其他模块执行。例如，可以由创建模块 302 执行确定要发送的数量的过程。此外，确定模块 304 的某些或全部功能可以由发送模块 306 来执行，发送模块 306 的某些或全部功能也可以由确定模块 304 来执行。类似地，分散模块 308 的某些或全部功能可以由发送模块 306 来执行，反之亦然。可以进一步理解，设备 300 的功能可以由在处理器上执行的软件来执行，或由在设备的或系统的硬件或固件中定义的寄存器来执行。

随后的简要流程图一般是作为逻辑流程图阐述的。如此，所描述

的顺序和标出的步骤表示所呈现的方法的一个实施例。可以有在功能、逻辑或效果上等效于所说明的方法的一个或多个步骤或其一部分的其他步骤和方法。另外，所使用的格式和符号只是说明该方法的逻辑步骤，不对该方法的范围作出限制。虽然可以在流程图中使用各种箭头和线条类型，但是，它们不对对应的方法的范围作出限制。实际上，可以使用某些箭头或其他连接符来只表示方法的逻辑流程。例如，箭头可以表示所描述的方法的所列举的步骤之间的未指定的持续时间的等待或监视时间段。另外，执行特定方法的顺序可以，也可以不严格地遵循所显示的对应的步骤的顺序。

现在请参看图 4，该图显示了用于评估通信链路上的错误的方法 400。在一个实施例中，方法步骤包括执行上文所概述的系统和设备的功能所需的那些步骤。方法 400 开始 402，创建 404 用于通过通信链路发送到接收节点的一个或多个测试数据包。然后，方法 400 确定 406 通过通信链路发送到接收节点的测试数据包的量。确定测试数据包的量的过程可以包括监视通过通信链路发送的数据包的量。如果要发送的测试数据包的量足以阻塞通信链路发送常规数据流，则发送较少的测试数据包。在一个实施例中，确定是否要发送更多测试数据包的过程是动态地进行的。如上文所讨论的，确定步骤 406 可以包括确定待分配的发送到接收节点的数据包的数量的百分比以发送测试数据包。它还可以包括确定要发送的测试数据包的具体数量或发送测试数据包所采用的速率。确定步骤 406 还可以确定发送测试数据包的时间量。

多个测试数据包与多个数据包一起通过通信链路发送到接收节点。在一个实施例中，数据中被分散以测试数据包发送 406。测试数据包被分散到数据包中，以便接收节点能够连续地处理数据包。如此，测试数据包是以可变速率分散的，以测试通信链路，而不会禁用通信链路发送和接收其他数据的能力。

数据可以以“脉冲数据传输”的方式发送，其中，以多个字节发送数据，无需微处理器的干涉。在一个实施例中，在跨通信链路传输之前，数据被从硬盘移到缓冲器中。发送数据 406 的过程可以包括向

测试数据包分配优先级。在一个实施例中，在通信链路上发送 406 足够的测试数据包，以模拟通信链路和/或接收节点的预定错误率。

测试数据包可以以人工方式每隔一定的时间间隔分散到数据包中。在一个实施例中，可以基于任意数量的性能因素，自动地将测试数据包分散到数据包流中。例如，如果系统的资源由于某种原因被完全占用，则可以在系统有资源可以被释放时发送测试数据包。

在一个实施例中，将测试数据包随机地分散到数据包中。本领域技术人员应认识到，随机数据通常比以特定模式或模式集发送数据产生更多的错误，这可以提供对通信链路上的任何错误的更清楚的评估。

检查 410 数据包，以查找错误，方法 400 结束 412。在通信链路的另一端由接收节点接收到数据包之后，检查 410 数据包。检查数据包的过程包括在错误寄存器中记录错误。寄存器可以是硬件芯片集的一部分，并可以被编程，以确定测试数据包优先级和测试数据包利用率。在一个实施例中，只检查测试数据包的错误。测试数据包可以包括随机数据，该随机数据提供错误概况，在一个实施例中消除了检查对常规数据的 CRC 测试的必要性。

方法还可以包括重复步骤 402 到 410 的处理的步骤，直到满足预定的错误阈值。

现在请参看图 5，该图显示了用于评估通信链路上的错误的另一种方法 500。方法开始 502，生成 504 一个或多个测试数据包。可以监视通信链路上的数据流 506。确定 508 将测试数据包分散到数据中的速率。然后，将测试数据包通过通信链路分散 510 到待测试的数据包。所属领域的技术人员应了解，通信链路可以一起地或单独地包括，电缆或线路、发送点以及结束点。在一个实施例中，结束点是接收节点。

检查 512 通信链路的性能，以便确定发送的测试数据包的数量是否将通信链路的性能抑制在无法接受的水平。例如，如果正在发送太多的测试数据包，则可能会将常规数据的流动妨碍到无法接受的水平。此水平可以是预定的，也可以基于各种因素和用户的需求，进行

调节。

然后，确定预定的错误阈值 514。例如，发送最小数量的测试数据包以近似于特定通信链路的已知错误阈值可能是有益的。例如，如果对于特定端口或其他接收节点的标准错误率是每百万字节的数据一个错误，方法 500 可以发送一百万字节的数据，以便获得在通信链路上发生的错误的清晰的概况。如果没有到达阈值，则重新检查通信链路的性能 516。如果性能低于所需的水平，则重新评估 508 将测试数据包分散到数据中的速率。如果性能正常，则分散 510 测试数据包的过程继续。当满足 514 测试数据包的所需的阈值时，方法结束 518。

在一个实施例中，上文所描述的方法是计算机程序产品或有形地实现了可由数字处理设备执行的机器可读的指令的程序的信号承载介质的操作的形式实现的。在一个实施例中，计算机程序产品包括或有形地实现了具有计算机可使用的用于有助于访问系统信息的程序代码的计算机可读取的介质。计算机程序产品包括实现上文所描述的方法的计算机可使用的程序代码。该方法和对应的程序代码还可以包括执行上文所描述的模块的功能所需的步骤。

在不偏离本发明的精神和基本特征的情况下，本发明可以以其他特定形式来实现。所描述的实施例在各个方面都只作为说明性的，而不是限制性的。因此，本发明的范围由所附权利要求而不是由前面的描述指出。在权利要求的等效内的含义和范围内的所有更改都将包括在它们的范围内。

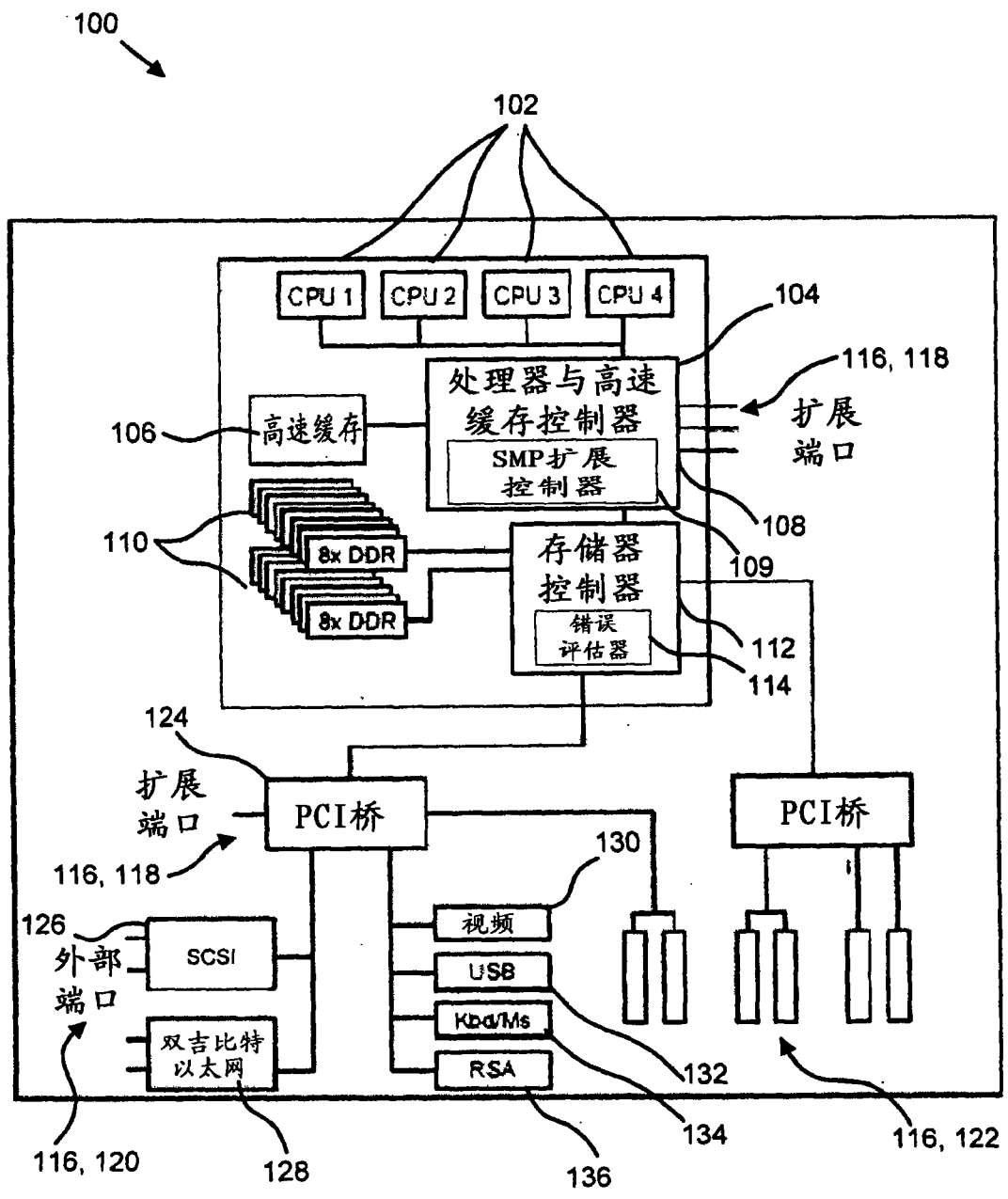


图 1

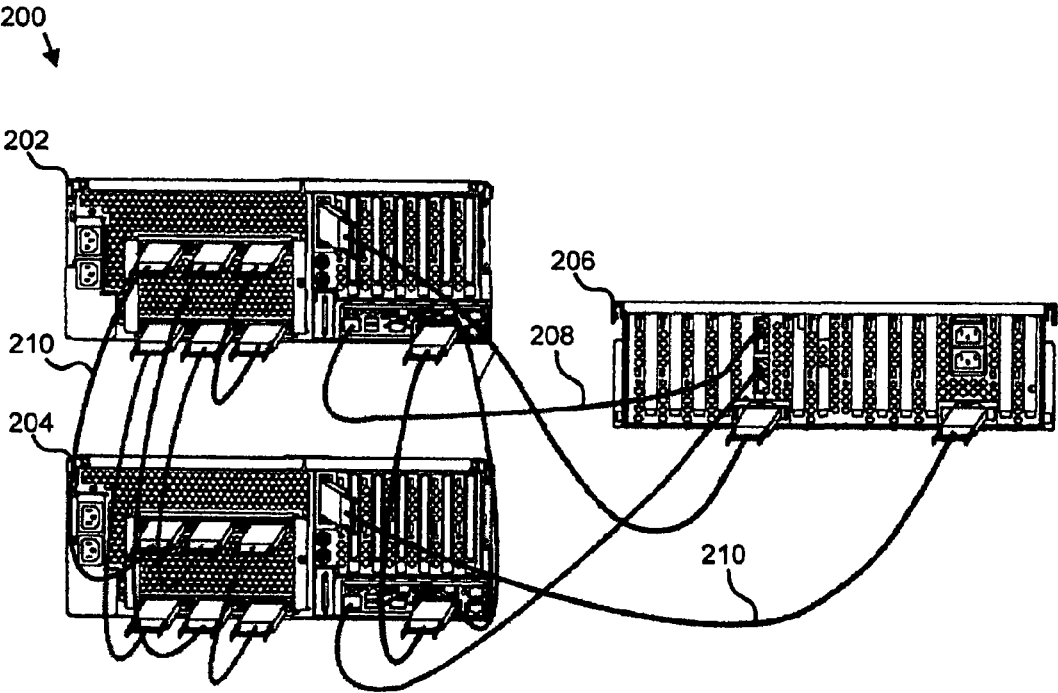


图 2

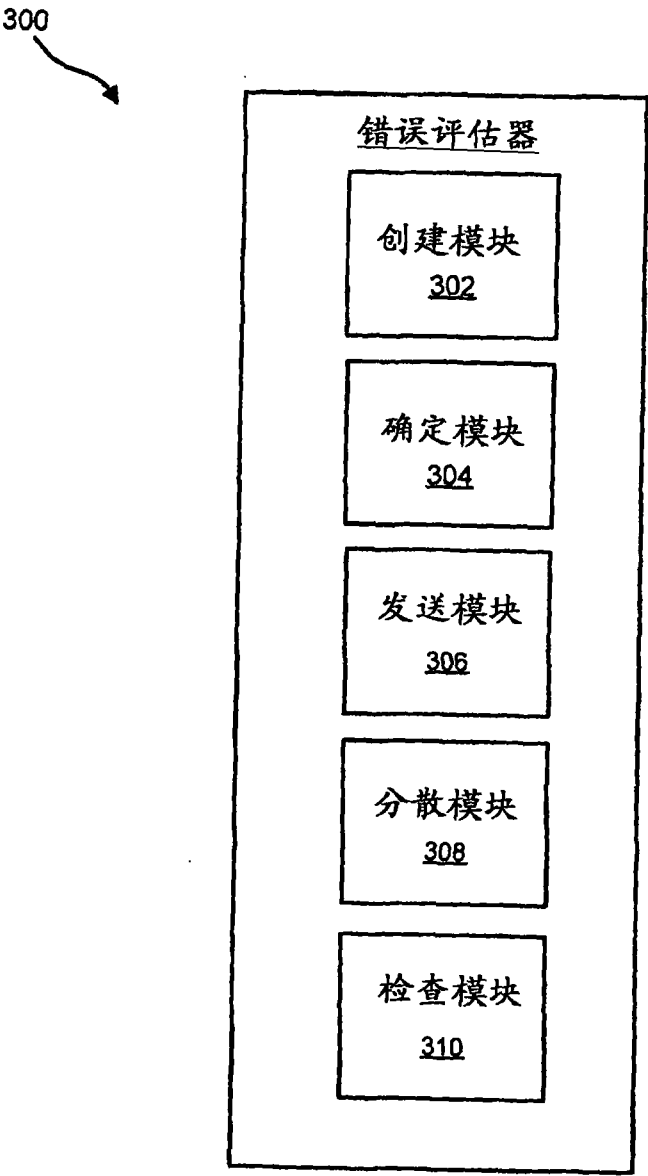


图 3

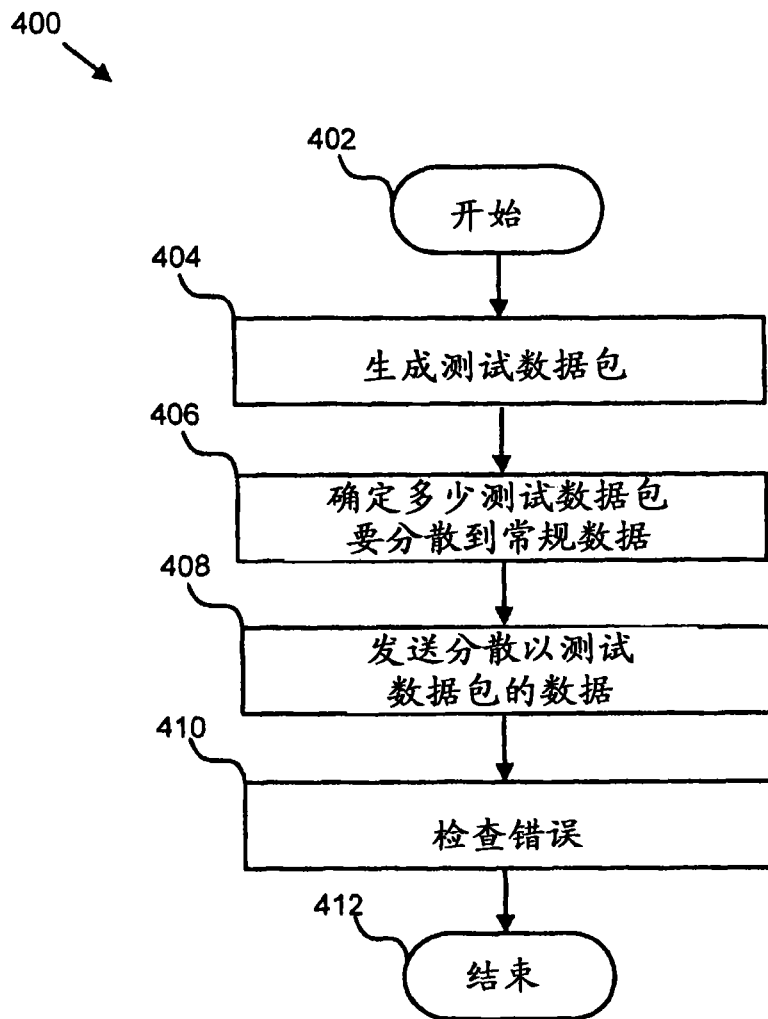


图 4

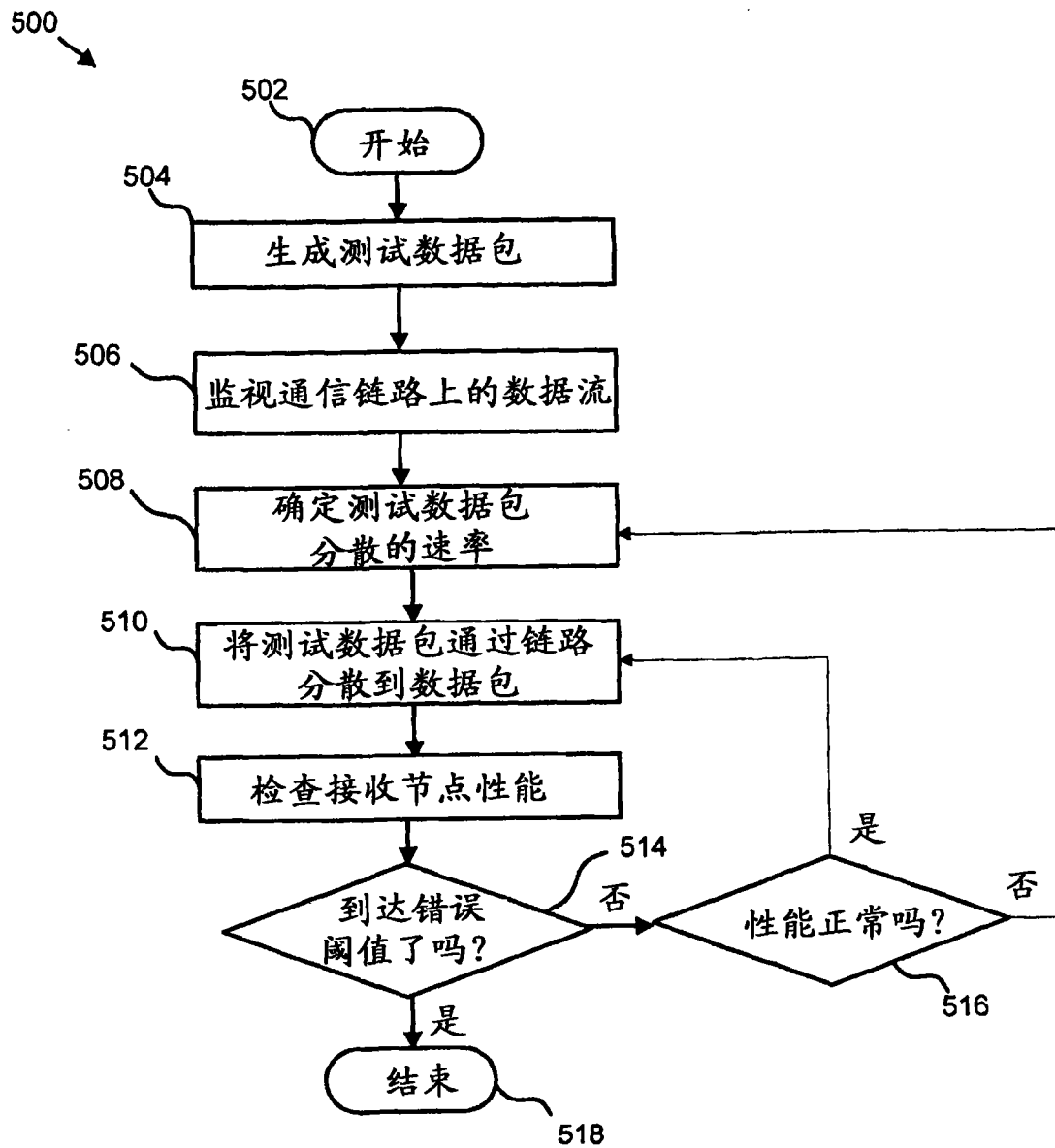


图 5

1. 一种用于评估通信链路上的错误的方法, 所述方法包括:
创建一个或多个测试数据包;
确定通过通信链路向接收节点发送的测试数据包的量;
通过所述通信链路向接收节点发送多个数据包;
通过所述通信链路向所述接收节点发送一个或多个测试数据包;
在所述数据包中分散测试数据包, 以便所述接收节点能够连续地处理所述数据包; 以及
检查由所述接收节点接收到的数据包, 以发现错误, 其特征在于:
确定测试数据包的量的过程包括, 确定待分配的发送到所述接收节点的数据包的数量的百分比以发送测试数据包。
2. 根据权利要求 1 所述的方法, 其中, 发送测试数据包的过程包括, 有选择地向连续的测试数据包分配优先级。
3. 一种包括用于执行根据任何前面的权利要求之一所述的方法的所有步骤的装置的系统。
4. 一种计算机程序, 包括用于当所述计算机程序在计算机系统上执行时执行根据任何前面的权利要求之一所述的方法的所有步骤的指令。