

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 15/173 (2006.01)

H04J 1/16 (2006.01)

H04L 1/00 (2006.01)

H02H 3/05 (2006.01)



# [12] 发明专利说明书

专利号 ZL 02802723. X

[45] 授权公告日 2007 年 4 月 18 日

[11] 授权公告号 CN 1311381C

[22] 申请日 2002.7.15 [21] 申请号 02802723. X

[30] 优先权

[32] 2001. 8. 22 [33] US [31] 09/934,884

[86] 国际申请 PCT/US2002/022220 2002. 7. 15

[87] 国际公布 WO2003/023637 英 2003. 3. 20

[85] 进入国家阶段日期 2003. 4. 22

[73] 专利权人 诺基亚股份有限公司

地址 美国德克萨斯州

[72] 发明人 文革·任

[56] 参考文献

US 6173324B1 2001. 1. 9

EP 1009133A2 2000. 6. 14

CN 1202772A 1998. 12. 23

EP 1001344A2 2000. 2. 17

WO 0108334A1 2001. 2. 1

审查员 丁文勃

[74] 专利代理机构 北京集佳知识产权代理有限公司

代理人 王学强

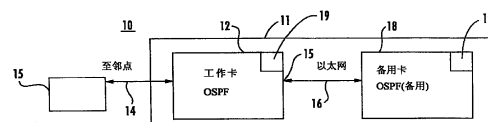
权利要求书 7 页 说明书 10 页 附图 4 页

[54] 发明名称

实现 OSPF 冗余的方法和系统

[57] 摘要

本发明关于在路由器(11)中实现链路层协议冗余的方法和系统,具体的,关于提供开放式最短路径优先(OSPF)路由协议的冗余。一个工作处理器(12)提供 OSPF 运算。在本发明中,一备用处理器(18)被连接至该工作处理器(12)。在启示同步阶段,所有来自工作处理器(12)的网络链路协议信息均被发送至备用处理器(18)。网络链路信息包括 OSPF 状态信息、OSPF 配置信息、OSPF 邻接信息、OSPF 接口信息和 OSPF 全局协议信息。然后,网络链路协议信息的任何更新都会立即被发送到备用处理器(18)。一旦工作处理器(12)发生故障,路由器(11)会被切换到备用处理器(18),且所有 OSPF 协议操作均在备用处理器(18)上进行。在本发明中,链路协议的所有状态立即开始运行,好象没有发生过故障一样。



1、一种链路协议冗余方法，包括下列步骤：

提供具有工作处理器的路由器；

将一备用处理器连接至所述工作处理器；

在将所述备用处理器连接至所述工作处理器后，将网络链路协议信息从所述工作处理器发送至所述备用处理器，以在所述备用处理器中进行与所述工作处理器的链路配置和链路协议状态的同步；

当检测到在所述工作处理器中有故障时，将所述路由器切换到所述备用处理器；

其中，所述链路协议的所有状态会立即起作用，好像没有故障发生一样。

2、如权利要求 1 所述的方法，其中所述链路协议是开放最短路径优先 OSPF 协议。

3、如权利要求 2 所述的方法，其中，所述链路协议信息是链路状态数据库信息、OSPF 配置信息、OSPF 邻接信息、OSPF 接口信息和 OSPF 全局协议信息。

4、如权利要求 2 所述的方法，进一步包含在所述工作处理器和所述备用处理器之间进行所述链路配置和链路协议状态的同步之后处理相同 OSPF 包的步骤。

5、如权利要求 3 所述的方法，其中，发送链路协议信息的步骤通过下列步骤实现：

为所述工作处理器的每个区域创建一个隐藏的 OSPF 接口；

为所述备用处理器的每个区域创建一个隐藏的 OSPF 接口；

从所述工作处理器的所述隐藏的 OSPF 接口发送所述链路状态数据库信息到所述备用处理器的所述隐藏的 OSPF 接口，直到所述备用处理器的所述链路状态数据库与所述工作处理器的所述链路状态数据

库同步。

6、如权利要求 5 所述的方法，进一步包括使用所述工作处理器的所述隐藏的 OSPF 接口和所述备用处理器的所述隐藏的 OSPF 接口发送所述 OSPF 配置信息、OSPF 邻接信息、OSPF 接口信息和 OSPF 全局协议信息的步骤。

7、如权利要求 5 所述的方法，其中所述链路协议信息采用进程间控制消息的形式。

8、如权利要求 7 所述的方法，其中所述 OSPF 配置信息根据存储在数据库中的命令行界面命令确定。

9、如权利要求 1 所述的方法，进一步包含下列步骤：

更新在所述工作处理器中的网络链路协议信息；以及

将所述更新的网络链路协议信息发送到所述备用处理器。

10、如权利要求 4 所述的方法，其中，所述发送步骤是基于 OSPF 协议的数据库交换处理的处理过程。

11、如权利要求 2 所述的方法，其中，

所述将网络链路协议信息从所述工作处理器发送至所述备用处理器，以在所述备用处理器中进行与所述工作处理器的链路配置和链路协议状态的同步的步骤包括：

在所述工作处理器装置上创建一个隐藏的 OSPF 接口，在所述备用处理器装置上创建一个隐藏的 OSPF 接口；

通过一通信链路连接所述工作处理器装置的所述隐藏的 OSPF 接口至所述备用处理器装置的所述隐藏的 OSPF 接口；

通过所述隐藏的 OSPF 接口利用 OSPF 协议同步 OSPF 路由数据库，使得当所述工作处理器装置的所述隐藏的 OSPF 接口和所述备用处理器装置的所述隐藏的 OSPF 接口在隐藏的 OSPF 接口上具有隐藏的邻接时同步所述 OSPF 路由数据库；

通过通信链路从所述工作处理器装置的所述隐藏的 OSPF 接口发送

OSPF 协议信息到所述备用处理器装置的所述隐藏的 OSPF 接口，以便镜像所述工作处理器装置和所述备用处理器装置的状态；

除去所述工作处理器装置的所述隐藏接口和所述备用处理器装置的所述隐藏接口。

12、如权利要求 11 所述的方法，其中所述 OSPF 协议信息是 OSPF 配置信息、OSPF 邻接信息、OSPF 接口信息和 OSPF 全局协议信息。

13、如权利要求 12 所述的方法，其中所述 OSPF 配置信息根据存储在数据库中的命令行界面命令确定。

14、如权利要求 11 所述的方法，进一步包含下列步骤：

更新在所述工作处理器装置中的网络链路协议信息；且

发送所述更新的网络链路协议信息到所述备用处理器装置。

15、如权利要求 11 所述的方法，其中，所述同步 OSPF 路由数据库的步骤是基于 OSPF 协议的数据库交换过程的处理过程。

16、如权利要求 11 所述的方法，进一步包含在同步所述工作处理器和所述备用处理器之间的所述链路配置和链路协议状态后处理相同 OSPF 包的步骤。

17、在一个路由器中提供链路协议冗余的系统，包括：

工作处理器；

备用处理器；

把网络链路协议信息从所述工作处理器发送到所述备用处理器的装置，用于在备用处理器中实现与所述工作处理器的链路配置和链路协议状态的同步；

当检测到在工作处理器中有故障时把所述路由器切换到所述备用处理器的装置；

其中，所述链路协议的所有状态立即起作用，好像没有故障发生一样。

18、如权利要求 17 所述的系统，其中所述链路协议是开放最短路

径优先协议。

19、如权利要求 17 所述的系统,其中所述链路协议信息是链路状态数据库信息、OSPF 配置信息、OSPF 邻接信息、OSPF 接口信息和 OSPF 全局协议信息。

20、如权利要求 19 所述的系统,其中所述把网络链路协议信息从所述工作处理器发送到所述备用处理器的装置包括:

为所述工作处理器装置的每个区域创建一个隐藏的 OSPF 接口的装置;

为所述备用处理器装置的每个区域上创建一个隐藏的 OSPF 接口的装置; 和

用于从所述工作处理器的所述隐藏的 OSPF 接口发送所述链路状态数据库信息到所述备用处理器的所述隐藏 OSPF 接口,直到所述备用处理器的所述链路状态数据库与所述工作处理器的所述链路状态数据库同步的装置。

21、如权利要求 20 所述的系统,其中,所述把网络链路协议信息从所述工作处理器发送到所述备用处理器的装置包括利用所述工作处理器的所述隐藏的 OSPF 接口和所述备用处理器的所述隐藏的 OSPF 接口发送所述 OSPF 配置信息、所述 OSPF 邻接信息、所述 OSPF 接口信息和所述 OSPF 全局协议信息。

22、如权利要求 21 所述的系统,其中所述 OSPF 配置信息根据存储在数据库中的命令行界面命令确定。

23、如权利要求 17 所述的系统,进一步包含:

在所述工作处理器处更新网络链路协议信息的装置;

发送所述更新的网络链路协议信息到所述备用处理器的装置。

24、如权利要求 17 所述的系统,其中所述把网络链路协议信息从所述工作处理器发送到所述备用处理器的装置包括:

冗余卡管理器,用于维护用于所述协议的任务的所述链路协议状

态的同步状态机，所述网络链路协议信息通过所述冗余卡管理器发送。

25、如权利要求 24 所述的系统，进一步包括用于确定所述任务的所述链路协议状态并向所述冗余卡管理器发送所述链路协议状态的任务管理器。

26、如权利要求 24 所述的系统，其中所述用于切换所述路由器到所述备用处理器的装置包含软件冗余管理器，其与所述冗余卡管理器交互以显示从所述工作处理器到所述备用处理器的切换。

27、如权利要求 24 所述的系统，其中在将备用处理器连接到所述工作处理器之前，所述任务的所述状态进入为初始状态的 OSPF\_FAULT\_INIT 状态。

28、如权利要求 24 所述的系统，其中，所述任务的所述状态进入 OSPF\_FAULT\_VERIFY 状态，其为在所述工作处理器和所述备用处理器的所述链路配置同步期间进入的状态。

29、如权利要求 24 所述的系统，其中在将所述链路协议信息从所述工作处理器发送到所述备用处理器期间，所述任务的所述状态进入 OSPF\_FAULT\_SYNC 状态，所述链路协议信息包括链路状态数据库信息、OSPF 配置信息、OSPF 邻接信息、OSPF 接口信息和 OSPF 全局协议信息。

30、如权利要求 24 所述的系统，其中在所述发送网络链路协议信息之后，所述任务的所述状态进入 OSPF\_FAULT\_FULL 状态，所述 OSPF\_FAULT\_FULL 状态为热备用状态，其中所述备用状态可立即接管所述备用处理器的所有操作。

31、如权利要求 17 所述的系统，其中所述工作处理器是一现用 OSPF 控制卡。

32、如权利要求 17 所述的系统，其中所述备用处理器是一个备用 OSPF 控制卡。

33、如权利要求 17 所述的系统，其中所述把网络链路协议信息从所述工作处理器发送到所述备用处理器的装置基于 OSPF 协议的数据库交换过程进行处理。

34、如权利要求 17 所述的系统，进一步包含：

在所述工作处理器和所述备用处理器之间同步所述链路配置和链路协议状态后处理相同的 OSPF 包的装置。

35、如权利要求 18 所述的系统，其中，所述把网络链路协议信息从所述工作处理器发送到所述备用处理器的装置包含：

在所述工作处理器装置上创建隐藏的 OSPF 接口和在所述备用处理器装置上创建隐藏的 OSPF 接口的装置；

用于通过通信链路连接所述工作处理器装置的所述隐藏的 OSPF 接口到所述备用处理器装置的所述隐藏的 OSPF 接口的装置；

利用 OSPF 协议通过所述隐藏的 OSPF 接口同步 OSPF 路由数据库的装置，使得当所述工作处理器装置的所述隐藏的 OSPF 接口和所述备用处理器装置的所述隐藏的 OSPF 接口达到完全接近的状态时同步所述 OSPF 路由数据库；

通过通信链路从所述工作处理器装置的所述隐藏的 OSPF 接口发送 OSPF 协议信息到所述备用处理器装置的所述隐藏的 OSPF 接口的装置，以镜像所述工作处理器装置和备用处理器装置的状态；

除去所述工作处理器装置的所述隐藏的接口和所述备用处理器装置的所述隐藏的接口的装置。

36、如权利要求 35 所述的系统，其中所述 OSPF 协议信息是 OSPF 配置信息、OSPF 邻接信息、OSPF 接口信息和 OSPF 全局协议信息。

37、如权利要求 35 所述的系统，其中所述 OSPF 配置信息根据存储在数据库中的命令行界面命令确定。

38、如权利要求 35 所述的系统，进一步包含：

用于更新在所述工作处理器装置中的网络链路协议信息的装置；

用于发送所述更新的网络链路协议信息到所述备用处理器装置的装置。

39、如权利要求 35 所述的系统,其中所述通过通信链路从所述工作处理器装置的所述隐蔽的 OSPF 接口发送 OSPF 协议信息到所述备用处理器装置的所述隐蔽的 OSPF 接口的装置基于 OSPF 协议数据库交换过程进行处理。

40、如权利要求 35 所述的系统,进一步包含在所述工作处理器和所述备用处理器之间实现所述链路配置和链路协议状态的同步后,用于处理相同的 OSPF 包的装置。



## 实现 OSPF 冗余的方法和系统

### 技术领域

本发明涉及网络通信，更具体地，涉及路由协议，如开放式最短路径优先（OSPF）协议的冗余，和用于保护一路由器和邻近路由器的协议业务免受故障损害的设备。

### 背景技术

互联网协议（IP）是许多公用网和专用网的基础，如因特网，企业内部互联网、数据网等。语音、数据和多媒体网络的集中在很大程度上也基于 IP 协议。

数据包通过从一台计算机上发送到另一台计算机在数据网中穿行到达其目的地。路由器或其他类型的交换机用于在数据源，例如连接在数据网上的一台用户计算机到目标之间的一条或多条链路上路由数据包。路由协议如边界网关协议（BGP）、路由信息协议（RIP）和开放式最短路径优先协议（OSPF）使每台计算机知道数据包在传递到目标过程中的“下一跳”是哪台计算机。路由器使用路由协议来建立路由表。当路由器接收到一个数据包并且必须进行转发决定时，路由器在路由表中“查询”下一路程段的计算机。通常情况下，路由器使用数据包中目标 IP 地址作为索引来查询路由表。

在基本的 OSPF 算法中，路由器广播一个包含路由器自身 ID、已知的相邻路由器 ID 的广播问候数据包，同时也从其他路由器上收到这样类似的信息。如果路由器从其已知的另一个路由器接收到一个包含其自身 ID 的问候数据包，如果这两个路由器彼此知晓，那么这两个路由器通过发送路由协议数据包来交换网络链路状态信息。路由器

基于运行链路状态路由算法收集的网络链路状态信息生成路由表，所述路由算法典型地如 Dijkstra 算法。在 OSPF 中，路由表可以基于通过考虑包括作为数据包路由的网络链路带宽在内的许多因素确定的消耗指定最小消耗路径。当网络链路改变时，每个路由器为其自身计算到每个网络的最短路径并根据这些路径设置自身的路由表。路由计算单元用来生成路由表。

每个路由器在发送或接收控制数据包和网络链路状态信息时，管理该路由器所连接的网络上的其他路由器的状态和接口状态，该路由器通过此接口连接到网络上。关于路由器的状态，每个路由器管理其 ID，并检查是否每一个其他路由器都感知到它，或检查是否那些路由器的每一个都完成了发送和接收网络链路状态信息。关于接口状态，每个路由器管理接口地址和其他连接在连接有接口的网络上的路由器的地址。

当常用的 IP 边缘路由器丢失其主电路，并且操作退回到冗余控制器时，在路由器释放路由状态和数据包转发表，产生 5 到 15 分钟的运行中断。为了提高路由器设备的可靠性，最重要的是要使上面提到的路由计算单元多路复用。多路复用路由器设备包括多个路由计算单元，且总是使其中一个路由器计算单元处于工作模式 (active mode)，用于处理常规的程序，其他路由计算单元则处于备用模式。当处于工作模式的路由计算单元出现故障，多路复用路由器设备使其中一个备用模式的路由器计算单元进入工作模式 (这被称作路由器计算单元的系统切换)，另一个路由器接管并继续处理前面出现故障的路由器计算单元执行的处理。

美国专利 No. 6049524 描述了一种多路复用路由器设备，它减少了从一个工作的路由计算单元到另一个备用路由计算单元间传送的信息。工作模式下的路由计算单元通过内部总线与备用路由计算单元连接。处于工作模式的路由计算单元存储表示该路由器和其它路由器

与网络的连接的网络链路状态信息，表示邻近路由器状态的邻近路由器状态以及表示网络接口的状态的接口状态，以将该多路复用路由器装置连结至该网络。工作模式的路由计算单元仅向备用模式的路由计算单元发送网络链路状态信息。在备用模式的路由计算单元，接收链路状态信息的数据库集成模块将其内容记录于链路状态数据库。当工作模式下的路由计算单元出现故障时，该路由计算单元通过使用存储的链路状态数据库进行路由协议处理，因此有必要与其他路由器交换信息以再次搜集网络链路状态信息。在切换至工作模式后的一小段时间，该路由计算单元没有关于该邻近路由状态和接口状态的信息。从进入工作模式的路由计算单元发出问候数据包。进入现用状态的路由计算逐渐累积有关邻近路由器状态和接口状态的信息，以便逐渐产生其他路由器的 ID 的列表，该列表包含在该路由计算单元之后发出的问候数据包中。

期望通过改进的冗余性提供高的网络可用率，这可以实现为在具有与现用链路层处理完全实时同步的备份链路层处理的 IP 上运行链路层协议，使得在该现用控制卡出现故障时能够进行快速的切换。

### 发明内容

本发明涉及用于实现路由器中链路层协议冗余的方法和系统。具体的，本发明涉及提供开放式最短路径优先 (OSPF) 路由协议的冗余。工作处理器提供 OSPF 操作。在本发明中，一备用处理器被连结至该工作处理器。在初始的同步期间，来自该工作处理器的所有的网络链路协议信息均被转发至该备用处理器。该网络链路信息可包括 OSPF 状态信息，OSPF 配置信息，OSPF 邻接信息，OSPF 接口信息和 OSPF 全球协议信息。之后，对该网络链路信息的任何更新均被以有序的和受控的方式立即转发至该备用处理器。一旦工作的处理器产生故障，路由器切换到备用处理器上，并且所有的 OSPF 协议操作由备用处理

器操作。在本发明中，链路协议的所有状态立即开始工作，就好像没有发生故障一样。相邻的路由器将不会注意到在切换后有任何的不同，并且在切换后也不需要来自相邻路由器的任何附加信息。因此，路由器的传送能力不会受到影响，邻近的路由器也不会注意到系统发生了故障。

在本发明的一个具体装置中，在同步初始阶段，在工作处理器和备用处理器中确定一个隐藏的 OSPF 接口用于该路由器的每个区域。此隐藏接口是一个点对点的无编号接口，在外部看不到。工作处理器的链路状态数据库与备用处理器通过此隐藏 OSPF 接口同步。链路协议信息也通过此隐藏 OSPF 接口从工作处理器传送到备用处理器。一旦两个处理器的同步建立，每个区域的隐藏 OSPF 接口将被删除。

本发的工作和备用 OSPF 处理器处于高度同步的状态，也可称为热备用状态。因此，当工作的处理器故障时可以非常立即的切换到备用处理器。

### 附图说明

下面的图示将更加详细地反映本发明。

图 1 是一个实现 OSPF 冗余的系统的示意图。

图 2 是冗余软件实现的示意图。

图 3 是每个 OSPF 区域的隐藏接口执行的示意图。

图 4 是在现用 OSPF 控制卡上运行的 OSPF 处理的状态的示意图。

图 5 是从现用处理到备用处理传送网络链路状态信息的步骤的流程图。

### 具体实施方式

现在将详细参考本发明的一个具体实施例进行说明，附图中示有其实例。在任何可能的地方，同样的参考编号将在整个附图和说明书

用于值代相同或类似的部分。

图 1 是依据本发明的在路由器 10 中实现链路协议冗余的系统的示意图。路由器 11 包括现用 OSPF 控制卡 12。现用 OSPF 控制卡 12 执行 OSPF 运算。OSPF 运算包括建立保持和检验到一个或多个相邻的路由器 15 的一个或多个邻接（邻接）14，与相邻路由器交换网络信息并更新本地路由表为最好的网络路由。当两个相邻路由器的链路状态数据库同步后，两个路由器就称为邻接的。邻接控制仅在邻接处发送和接收的路由协议数据包的分配。

与路由器 11 连接的备用 OSPF 控制卡 18 是可移除的。没有备用 OSPF 控制卡时，现用 OSPF 控制卡 12 工作在无冗余模式。现用 OSPF 控制卡 12 通过通信信道 16 将网络链路协议信息 15 传送到备用 OSPF 控制卡 18 中。通信信道 16 最好是快速、可靠的通信信道。例如，通信信道 16 可以是双工以太网。网络链路协议信息 15 可以过程间控制（IPC）消息的形式传输。现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 上运行有相同的冗余软件用于 OSPF 运算 19。用于 OSPF 运算 19 的冗余软件控制现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 之间网络链路协议信息 15 的更新，并且使用系统状态信息来辨别工作模式和备用模式，下面有更详细的叙述。

本发明的一个实施例利用运行在 Amber Network ASR2000 路由器（也可以是 ASR2020）上的 OSPF 协议。这里完全引用结合 Amber Network ASR2000 和 ASR2020 技术手册。现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 是连接于线路卡和路由器 ASIC 驱动器的处理程序。可以理解，虽然系统 10 是根据 OSPF 协议运行的，本发明还能够与其他常规的链路协议共同使用。

在 OSPF 控制卡 18 连接到路由器 11 上后，使用用于 OSPF 运算 19 的冗余软件，从正在运行的现用 OSPF 控制卡 12 到备用 OSPF 控制卡 18 进行为网络链路信息 15 的成批更新的初始同步，。网络链路信

息 15 包括配置、状态和学习信息。

完成初始同步后，OSPF 现用和备用处理充分冗余，以用于 OSPF 运算 19 的冗余软件运行的 OSPF 程序以进一步的更新模式进行操作。更新可以传递到现用 OSPF 控制卡 12 中。所有的更新都传送到备用的 OSPF 控制卡 18 中。备用 OSPF 控制卡 18 接收所有的 OSPF 消息并进行更新，以保持与现用 OSPF 控制卡 12 的完全实时同步。因此，通过冗余，备用 OSPF 控制卡 18 完全反映现用 OSPF 控制卡 12 状态。这种情况，称为热备用，现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 保持完全同步的状态。因此，如果现用 OSPF 控制卡 12 发生故障，备用 OSPF 控制卡 18 将开始工作并立即接管现用 OSPF 控制卡 12 以前执行的所有操作。

图 2 是现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 的 OSPF 运算 19 的冗余软件的详细示意图。冗余卡管理器 (RCM) 20 是保持每个任务的同步状态机的任务。用于现用 OSPF 控制卡 12 的 OSPF 运算 19 的冗余软件的所有任务与 RCM 20 相互作用以向备用 OSPF 控制卡 18 发出网络链路信息 15。OSPF 任务 21 是用于确定运行在现用 OSPF 控制卡 12 上的 OSPF 程序状态的任务。软件冗余管理器 22 是与 RCM 20 相互作用的一个模块，用于确定从工作状态到备用状态的切换，在工作状态下，现用 OSPF 控制卡 12 执行 OSPF 操作，在备用状态下，备用 OSPF 控制卡 18 接管 OSPF 操作。

在初始同步期间，备用 OSPF 控制卡 18 中的冗余卡管理器 (RCM) 20 与现用 OSPF 控制卡 12 中的 OSPF 任务 21 进行联系以获取任务信息。现用 OSPF 控制卡 12 中的 OSPF 任务 21 自动地处理 OSPF 信息，并计算存储在路由表管理器 (RTM) 34 中的路由。现用 OSPF 控制卡 12 标记相应的内部状态并通过 RCM 20 将链路状态数据库信息 23、OSPF 状态信息 24 与 OSPF 配置信息 25、OSPF 邻接点信息 26、OSPF 接口信息 27 和 OSPF 全局协议信息 28 传送到备用 OSPF 控制卡 18 中。

在初始同步期间,可对运行在现用 OSPF 控制卡 12 上的现用 OSPF 程序进行锁定。例如,在现用 OSPF 控制卡 12 上,在建立 OSPF 邻接时可以保持锁定,使得在初始同步期间无法建立新的 OSPF 邻接。

在初始同步阶段,现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 的每个区域都建立了隐藏的 OSPF 接口 30。区域是指一组连续的网络和附属的主机。隐藏的 OSPF 接口 30 是一个点对点的无编号接口,它与系统 10 一起使用,在外部看不到。由于发现相邻 OSPF,隐藏的 OSPF 邻接 32 会在 OSPF 接口 30 上自动产生。数据库 33 在整个隐藏的 OSPF 邻接 32 中是同步的。因此,在现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 间每个区域有一个隐藏的 OSPF 邻接 32。隐藏的 OSPF 邻接 32 可用于同步存储于数据库 33 中的链路状态数据库信息 23。

图 3 图示了一个隐藏 OSPF 接口的实现。路由器 11 有两个接口,接口 14a 属于连接在路由器 15a 上的区域 0,接口 14b 属于连接在路由器 15b 上的区域 2。在路由器 11 中,建立了两个隐藏的 OSPF 接口,隐藏接口 30a 用于区域 0,隐藏接口 30b 用于区域 2。隐藏 OSPF 邻接 32a 在隐藏 OSPF 接口 30a 上运行,隐藏 OSPF 邻接 32b 在隐藏 OSPF 接口 30b 上运行。外部链路状态广播数据包 (LSA) 只通过区域 0 的隐藏接口 30a 同步。

图 2 中,现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 处理 OSPF 数据包并先计算最短路径,以确定从路由器到目标网络的最短路径(通过考虑耗费)。现用 OSPF 控制卡 12 能够发送 OSPF 数据包到线路卡上用于传送到相邻路由器。备用 OSPF 控制卡 18 不发送任何 OSPF 数据包到线路卡上用于传送到相邻路由器。现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 对路由表管理器 (RTM) 34 进行路由更新,如图 2 所示。备用 OSPF 控制卡 18 的 RTM 34 可更新到现用 OSPF 控制卡 12 的再分配路由。IP 接口管理器 35 将系统 10 连接到互联网协议 (IP) 上。命令行接口 (CLI) 的命令使用数据存储 36 提供 OSPF 配置。

数据存储器 36 是一个能提供在存储器 38 中可靠存储的任务。例如，存储器 38 可以是一个压缩的闪存盘。因此，所有由备用 OSPF 控制卡 18 得到的信息可以直接从现用 OSPF 控制卡 12、IP 接口管理器 35 或数据存储器 36 中获得。

工作状态与现用 OSPF 控制卡 12 相关，备用状态与备用 OSPF 控制卡 18 相关。从现用 OSPF 控制卡 12 到备用 OSPF 控制卡 18 的切换能够在现用 OSPF 控制卡 12 发生故障时进行。当切换发生时，备用 OSPF 控制卡 18 将其状态转换为工作并接管所有的 OSPF 操作。备用 OSPF 控制卡 19 恢复所有取消的 OSPF 动作并开始向线路卡发送 OSPF 数据包。

图 4 是一个在现用 OSPF 控制卡 12 上运行的现用 OSPF 程序 40 的状态示意图。OSPF\_FAULT\_INIT 状态 41 是现用 OSPF 程序 40 的初始状态。如果系统 10 只使用现用 OSPF 控制卡 12 工作，那么系统 10 会保持在 OSPF\_FAULT\_INIT 状态 41 等待备用 OSPF 控制卡 18 启动。

一旦备用 OSPF 控制卡 18 开始工作，OSPF\_FAULT\_VERIFY 状态 42 被输入，此过程中备用 OSPF 控制卡 18 安装来自数据存储器 36（在现用 OSPF 控制卡 12 中）的 OSPF 配置信息 25，同时 OSPF 配置在现用 OSPF 控制卡 12 上被激活，见图 2。此时，现用 OSPF 控制卡 12 上的配置禁用。来自数据存储器 36 中备用 OSPF 控制卡 18 的 OSPF 配置被同步并以现用 OSPF 程序 40 进行检验。现用 OSPF 程序 40 检验运行在备用 OSPF 控制卡 18 上的备用 OSPF 处理 44 是否完全同步配置以及数据存储器 36 的系统信息。例如，现用 OSPF 控制卡 12 可以检验接口数量和参数。如果检验失败，现用 OSPF 程序 40 可以在预先确定的时间间隔后再试，如几秒钟后。

在检验 OSPF 配置后，现用 OSPF 程序 40 和备用 OSPF 处理 44 输入 OSPF\_FAULT\_SYNC 状态 45。在 45 中，相邻信息通过位于现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 间的通信链路 16 传输，如图 5 块



50 所示。相邻信息可以作为 IPC 消息由现用 OSPF 程序 40 传送。多数 IPC 消息可以被用于传送大量的邻接点。备用 OSPF 处理 44 确认所接收的 IPC 消息，并向现用 OSPF 控制卡 12 发送一个确认 IPC 消息，如块 52 所示。

在传递相邻信息时，现用 OSPF 控制卡 12 通过忽略来自未知个体的问候数据包来拒绝接受任何新的邻接点。当从现用 OSPF 控制卡 12 到备用 OSPF 控制卡 18 的所有相邻信息传递以后，现用 OSPF 控制卡 12 将发送一个结束消息，如块 53 所示。

随后，备用 OSPF 处理 44 从块 54 的现用 OSPF 控制卡 12 下载链路状态数据库信息。链路状态数据库信息可以使用由 OSPF 提供的内部数据库同步方式被同步，如 RFC 2328 描述，可作为本应用的参考。每个路由器在使用“数据库交换程序”的数据库同步中通过向相邻路由器发送一系列数据库说明数据包描述其数据库。这两个路由器成为主/从关系。每个数据库说明数据包描述一组属于路由器数据库的 LSA。当相邻路由器发现一个 LSA 比其自身数据库备份中的更新，它会记录下应请求比较新的 LSA。每个数据库说明数据包具有一序列号。由主路由器发送的数据库说明数据包（查询）被从路由器通过回应该序列号进行确认。查询和回应都包含链路状态数据摘要。只有主路由器被允许再次发送数据库说明数据包，此数据包经过一个固定的时间间隔后发出。当数据库说明程序结束，数据库被认为已经同步并且路由器标记为是完成接近的。这时此接近完全可以起作用，并且在两个路由器 LSA 中进行了通告。在现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 之间确定隐藏的 OSPF 相邻 32 用于下载链路状态数据库信息 23。当现用 OSPF 控制卡 12 收到备用 OSPF 控制卡 18 的数据库需求消息时，现用 OSPF 控制卡 12 注意到备用 OSPF 控制卡 18 正在开始下载链路状态数据库信息 23。下载链路状态信息会一直持续，直到现用 OSPF 控制卡 12 和备用 OSPF 控制卡 18 中出现同步链路状态数据库。

当备用 OSPF 控制卡 18 具有与现用 OSPF 控制卡 12 同步的链路状态数据库时，现用 OSPF 控制卡 12 与备用 OSPF 控制卡 18 一起进入 OSPF\_FAULT\_FULL 状态 46。OSPF\_FAULT\_FULL 状态 46 是一种热备份状态，此时的备用 OSPF 控制卡 18 在现用 OSPF 控制卡 12 发生故障时能够立即接管其所有的操作。在 OSPF\_FAULT\_FULL 状态 46，隐藏的 OSPF 接口 30 与隐藏的邻接 32 被删除。现用 OSPF 程序 40 通过使用 IPC 消息经 RCM 20 立即向备用 OSPF 控制卡 18 发送更新的 OSPF 状态信息 24、OSPF 配置信息 25、OSPF 邻接信息 26、OSPF 接口信息 27 和 OSPF 全局协议信息 28，来不断更新对备用 OSPF 处理 44 的任何变化。任何现用 OSPF 控制卡 12 相邻状态的改变或相邻邻接的丢失都会通过通信链路 18 立刻传递到备用 OSPF 控制卡 18 中。任何链路状态数据的变化都要以常规的 OSPF 同步方式通过通信链路 15 传送到备用 OSPF 控制卡 18。

现用 OSPF 控制卡的配置变化可以作为 IPC 消息传递到备用 OSPF 控制卡 18 中，以触发备用 OSPF 控制卡 18 读取数据存储器 36 中的更新信息。或者，配置命令能够从 CLI 传递到备用 OSPF 控制卡 18 中。

如果当备用 OSPF 控制卡 18 处于 OSPF\_FAULT\_FULL 状态时现用 OSPF 控制卡 12 发生故障，备用 OSPF 控制卡 18 会立即接管所有的 OSPF 操作。如果当备用 OSPF 控制卡 18 处于 OSPF\_FAULT\_INIT 状态 41、OSPF\_FAULT\_VERIFY 状态 42 或 OSPF\_FAULT\_SYNC 状态 45 之一时现用 OSPF 控制卡 12 发生故障，说明备用 OSPF 控制卡没有处于完全的冗余状态，备用卡将复位。因为系统没有达到冗余状态，现用卡的故障将使运行中断。

应当理解，上述实施例只是许多能够体现本发明原理的可能的具体实施例中的少数。本专业技术人员在不脱离本发明的精神和范围的情况下，根据这些原理，可很容易地设计出其他多种不同的结构。

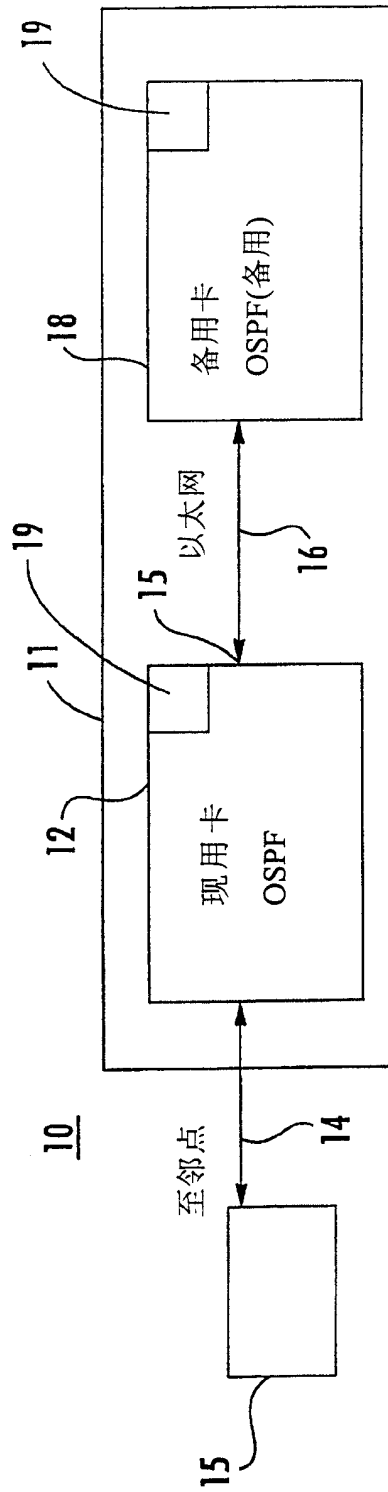


图 1

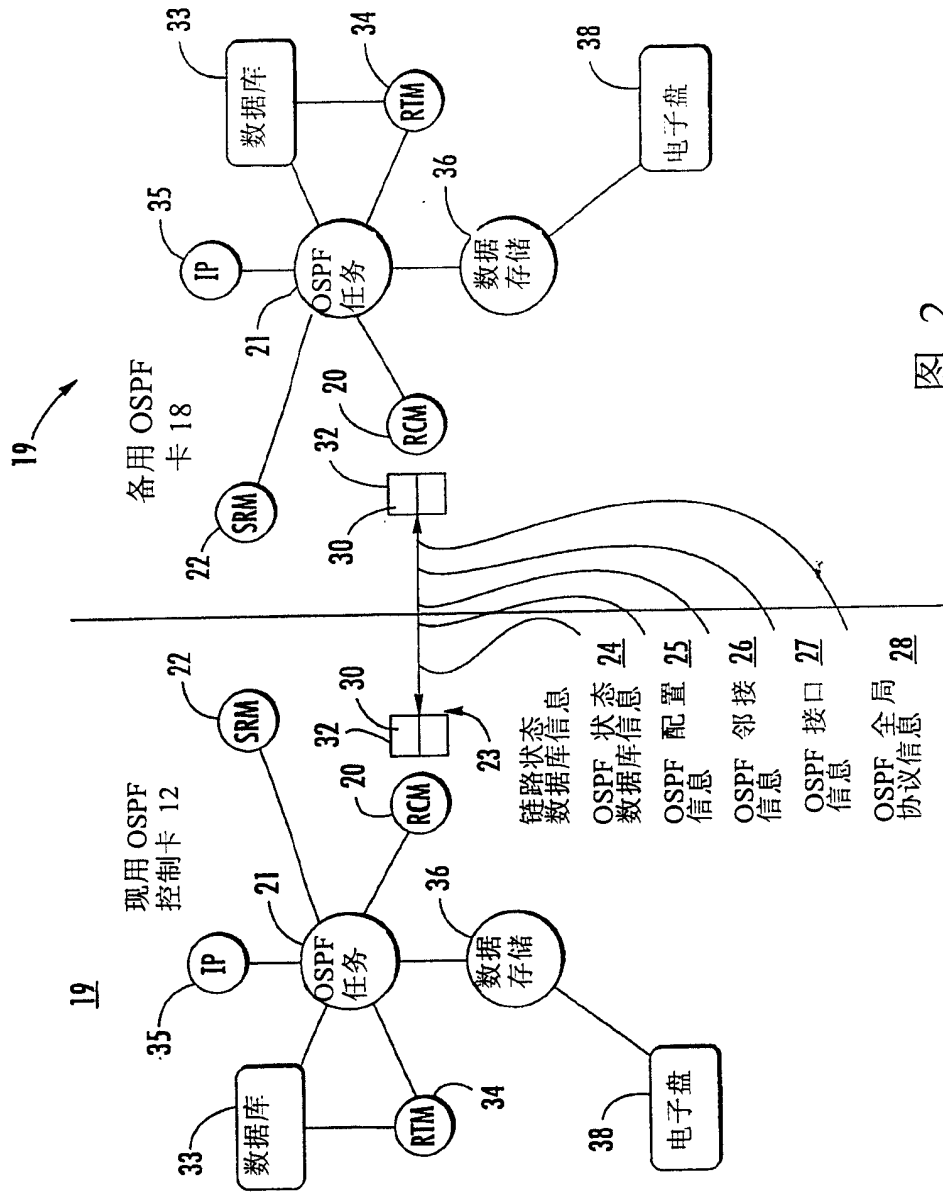


图 2

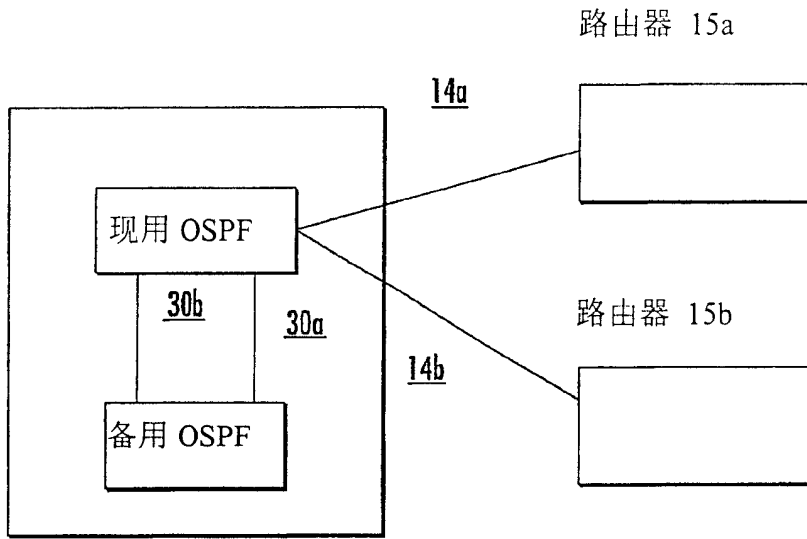


图 3

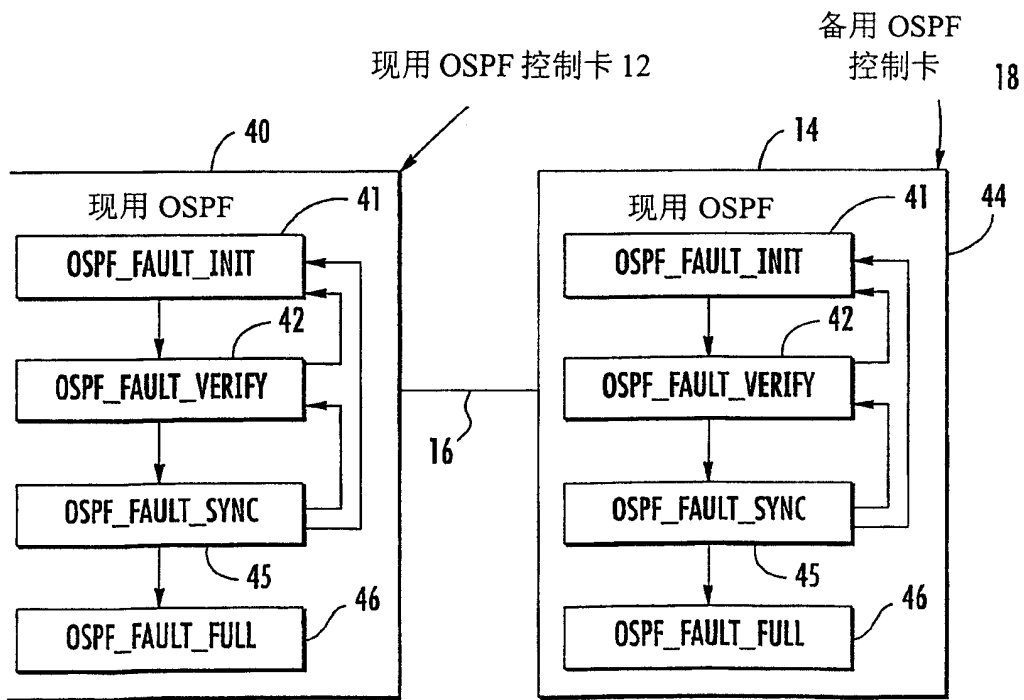


图 4

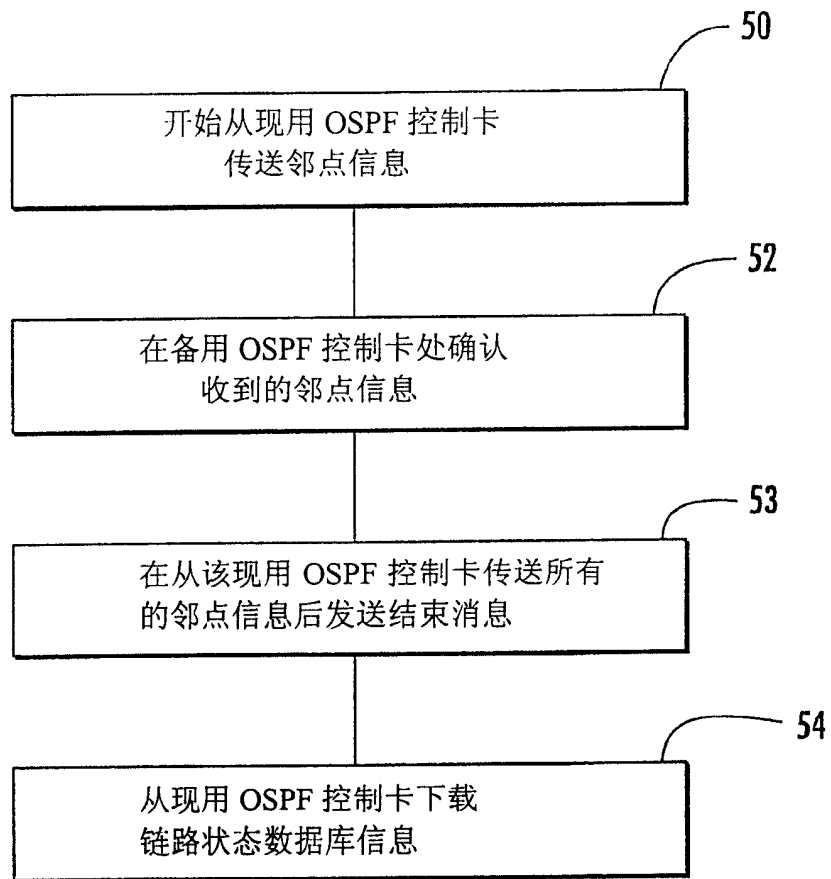


图 5