



[12] 发明专利申请公开说明书

[21] 申请号 95195435.0

[43]公开日 1997年9月17日

[11] 公开号 CN 1159858A

[22]申请日 95.8.18

[30]优先权

[32]94.8.19 [33]US[31]08 / 293,073

[86]国际申请 PCT / US95 / 10605 95.8.18

[87]国际公布 WO96 / 07257 英 96.3.7

[85]进入国家阶段日期 97.4.1

[71]申请人 皮尔罗杰克公司

地址 美国加利福尼亚州

[72]发明人 丹尼尔·P·格雷格森

戴维·R·法雷利 苏尼尔·S·盖顿迪

拉廷蒂·P·阿胡贾

克里什·罗摩克里希纳恩

穆罕默德·沙菲克

扬·F·沃利斯

[74]专利代理机构 永新专利商标代理有限公司

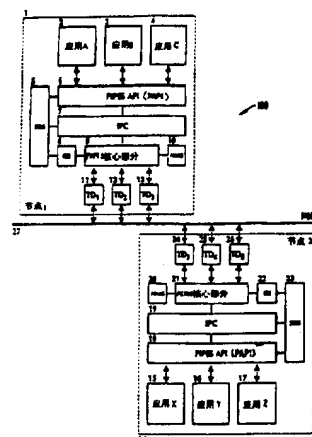
代理人 蹇 炜

权利要求书 5 页 说明书 23 页 附图页数 34 页

[54]发明名称 规模可以改变的分布式计算环境

[57]摘要

本发明涉及分布式计算系统,更具体地涉及规模可变的分布计算环境的体系结构与实现,它提供单一网络或互联网络上独立操作的节点之间的通信,它可以是同种的也可以是异种的。本发明为一动态、对称、分布式、实时、对等层对对等层系统,包括共同构成一棵逻辑树的任意数目的相同(语义上等价)的实例,即核。这些核通过网络的自配置与自维护逻辑视图在运行时呈现统一与相符的性能。各核位于具有一个或多个外与之关联的资源网络节点上。这些核动态实时地相互定位以构成与维护一个支持实际上不受限制的数目的独立运行的核层次结构。本系统维护其网络的逻辑视图与用户开发的程序资源而与传送协议的数目与组合及物理拓扑的基层混合无关。系统的通信服务利用动态桥接在可能不共用公共传送协议栈的端节点之间通信,从而允许位于不同的栈上的应用自动与透明地互相通信。



权 利 要 求 书

1、一种用于在包含多个节点的一个网络的节点中独立地执行软件成分的方法，该方法包括下述步骤：

在网络中生成节点的角色逻辑层次，其中任何节点都能担任一种或多种角色，角色的担任既不要求也不排除任何其它角色的担任；以及

在网络的配置有变化时协商节点的角色。

2、权利要求1的方法，其中具有管理角色的一个节点离开网络及其余节点中至少一个参加确定哪一个节点担任该管理角色的协商进程，协商步骤进一步包括由参加的节点执行的下述步骤：

广播表明参加的节点有兴趣担任该管理角色的报文；

在广播步骤之后收听网络上的报文；以及

如果在网络上没有表明另一节点担任该管理角色更称职的报文，便担任该管理角色。

3、权利要求1的方法，还包括由参加的节点执行的下述步骤：

在所述广播步骤之前，收听其他参加节点发送的报文一个指定的时间段；以及

当报文表明至少有一个参加节点担任该管理角色更称职时便退出该进程。

4、权利要求1的方法，其中至少两个冲突的节点要求同一管理角色及冲突的节点中至少一个参加确定担任管理角色的节点的进程，协商步骤进一步包括由各参加节点执行的下述步骤：

建立一个包含所有参加该进程的已知节点的名的数据库；

发送选举报文给含在数据库中的节点，选举报文中包含对发送节点已知的关

于参加节点的信息;

接收来自其它参加节点的选举报文;

用包含在收到的选举报文中的信息更新数据库; 以及

根据包含在更新后的数据库中的信息, 确定哪一个参加节点担任该管理角色。

5、权利要求1的方法, 其中节点之一为一双亲节点, 该方法还包括在一个节点进入网络时搜索一双亲节点的步骤。

6、权利要求5的方法, 其中的搜索步骤进一步包括由进入的节点执行的下述步骤:

收听报文一个指定的时间段;

如果收到报文, 根据收到的报文确定进入节点的双亲;

如果在指定的时间段用完时未找到双亲, 便广播用于搜索其双亲的一个报文;

收听对广播的报文的应答及判定应答中任何一个是否发自其双亲;

收不到应答时担任其自身的双亲的角色。

7、权利要求1的方法, 其中的节点之一为一双亲节点, 该方法还包括在其进入网络时登记一个子女的步骤。

8、权利要求7的方法, 其中所述登记步骤进一步包括:

收听由进入节点发送的报文;

判定报文之一是否由一个子女节点还是由一个双重子女节点所发送;

如果检测到一个双重子女节点, 通知该双重子女节点角色冲突;

如果检测到一个子女节点，发送一个确认报文给该子女节点。

9、权利要求1的方法，其中的节点之一为一双亲及其余节点之一为一子女，所述方法还包括监视双亲与子女的状态的步骤。

10、权利要求9的方法，其中该监视步骤进一步包括下述步骤：

在指定的是时间间隔上在双亲与子女之间交换状态报文；

当在一个预定的时间段内该子女收不到来自双亲的状态报文时，搜索一个新的双亲节点。

11、权利要求9的方法，还包括在一预定的时间段内双亲收不到来自该子女的状态报文时注销该子女的步骤。

12、权利要求1的方法，其中该逻辑层次包含任意数目的级。

13、权利要求12的方法，其中的级数是可变的。

14、权利要求1的方法，其中网络中节点的角色是可以随网络的要求变化的。

15、一种用于在互连数字网上的多个节点的规模可变的系统中寻找节点请求的资源的方法，各节点与一个或多个资源相关联，各资源具有它可被其它节点利用的活跃状态及不能利用的不活跃状态，该方法包括下述步骤：

自动识别通过从不活跃状态切换到活跃状态而参加网络的资源；以及

自动通知请求者所请求的资源已成为可以利用。

16、权利要求15的方法，其中所述多个节点中至少一个包含一个高速缓冲存储器用于存储一张资源请求表，所述方法还包括在高速缓冲存储器中存储资源请求的步骤。

17、权利要求16的方法，还包括当一个资源成为活跃并满足请求时从高

速缓冲存储器中清除一个请求的步骤。

1 8 、权利要求1 5 的方法，其中所述多个节点是配置在至少两级中的，及其中第一级中的节点包含关于出现在第二级中的节点中的资源的信息。

1 9 、权利要求1 8 的方法，其中第一级中的多个节点包含用于存储资源请求表的高速缓冲存储器，所述方法还包括在高速缓冲存储器中存储资源请求的步骤。

2 0 、权利要求1 9 的方法，还包括在一个资源成为活跃且满足请求时从高速缓冲存储器中清除一个请求的步骤。

2 1 、权利要求1 8 的方法，其中该第一级为双亲级而该第二级为子女级，双亲级中的各节点与子女级中的一个或多个节点关联，而子女级中的各节点与双亲级中的一个节点关联。

2 2 、一种用于能够在不同级上的具有不同通信协议的节点之间发送包的上下文桥接中确定路由路径方法，该上下文桥接是在包含多个节点的不同种类网络中的许多上下文桥接之一，该方法包括下述步骤：

建立一张上下文桥接及表中各上下文桥接处理的通信协议的表；

收听通过其它上下文桥接通知各接收者该广播上下文桥接所处理的通信协议而被周期性广播的路由信息包；以及

用包含在收到的路由信息包中的信息更新数据库。

2 3 、一种利用上下文桥接将包的路由从一个源节点确定到一个目的节点上的方法，源节点的路由协议与目的节点的路由协议在不同的级上，该上下文桥接是包含多个节点的不同种类网中的许多上下文桥接之一，该方法包括下述步骤：

判定目的节点是否有可确定路由协议；

利用包含上下文桥接的地址的源路由将包的路由从源确定到目的地上，如果目的地没有可确定路由协议，地址是通过源发送一个路由发现包且目的地回答该

发现包而被发现的；以及

如果目的地具有可确定路由协议，便通过至少一个包含可确定路由协议的上下文桥接确定该包的路由。

规模可以改变的分布式计算环境

本发明涉及分布式计算系统，更具体地针对一种使便于在单一网络或互连网络上的独立操作节点之间进行通信的同类型或不同类型的规模可以改变的分布式计算环境的体系结构与实现。

在当今的商业环境中，由于兼并与收购、全球化及与客户、供应商与金融机构的实时通信的需求的变迁与发展，联合结构正在快速重组。此外，在建立与保持竞争优势方面，对信息的快速访问及对信息的快速处理的要求已成为关键性的。这便要求以全新的方式来配置联合数据及处理这些数据的计算机程序；以分布方式而不是集中的整体的方式。

对于分布式计算，程序与数据是逻辑地定位的，从而对于与它们交互作用的用户，它们能被就近处理。理论上，通过减少通信开销及利用个人、集体与部门计算资源的未充分利用的处理能力而使公司更可靠与有效地工作。通过将工作负荷分布在许多计算机上，能为给定的个人、工作组或目的优化信息处理资源。这一方法允许分布与复制数据与进程，从而在系统的需要增加时更容易保持性能与可靠性。提高粒度及规模可变性这一特征还提供涉及软件可重复利用的重大优点，即同一部分可在若干不同应用中使用，从而降低开发及维护时间与成本。

由于这些需求，存在着面向全体企事业单位的虚拟计算的趋势，其中网络的全部资源对用户呈现为本地驻留在他或她的桌面计算机或终端上的。传统的整体式集中联合信息处理模型正在向分布式细粒方法让步。这一向虚拟动态企业计算的转变要求采用分布式体系结构来实现对任务关键的核心系统，在分布式体系结构中本地化计算资源（程序单元与数据）是用虚拟网络无缝地互相链接的。

然而，在当今的联合信息系统中，独立的应用通常存在在互不协作的不同环境中。企业面对着连接不兼容的系统同时在广阔的地理区域上保持不同操作系统与网络协议的数目不断增长的任务。团体合并与兼并又在兴起，对将已装设的不同网络集成为一个单一的全企事业网的需求不是一次而是多次地提出。再者，许多公司已成为全球实体而它们的信息系统现在必须在多个时区上工作，而要求这些系统是“与时间无关的”。再者，随着这些公司本身的动态转变，支持它们的商业操作的信息系统也是如此。从而，联合计算环境必须是“开放的”，即它必

须充分灵活以便容易地迁移到新的标准同时保持完整性及对其现有的“遗产”系统与数据的访问。遗产系统通常依靠静态表的使用来保持与连网的资源的接触。这些系统不支持动态恢复并且不能容易地将其规模扩展成在全体企业上展开的，因为在不断变化的环境中维护这些表需要极高的开销。

在现有系统中，为了使连网的一个资源发现另一资源的存在，两者都必须是“活的”。随着连网的资源总数的膨胀，具有一个与时间无关的资源发现机构并借此使网络自动地在新资源成为可利用时知道它们便变得极为重要。

现有系统还受到由任何节点能担任的诸如机器、区域、组群、地域、网络等角色或层次级的固定数据目的可获得性的限制。这一限制在合并或汇集两个或更多具有不同层次结构的现有网络时产生严重的问题。此外，在先有技术系统中，如果一个节点担任多种角色，这些角色之间的关系是规定的。这便是，为了在级1（如机器）与级3（如组群管理员）上工作，该节点还必须担任级2的功能（如区域管理员）。这一限制能严重地降低系统性能与恢复。

诸如实时通知、报文排队、远程过程调用、进程间通信以及广播/发布与预定等解决与建立坚固的高效全体企业计算环境相关的问题的原有努力至多只表示部分解决方案。因为真正的分布式计算必须有对等层对等层通信（由于主进程故障必定导致从进程故障），实现企业计算目标的基于客户服务器的方法只代表次最佳解决方案。利用静态表的现有对等层对等层系统并不允许动态恢复，并在规模可变性及维护上存在严重问题。

本发明为一种动态的、对称的、分布式、实时、对等层对等层系统，包括共同构成一个逻辑树的任意数目的相同（语义上等价的）实例，即核。这些核在通过一个自配置及自维护的网络逻辑视图的运行时间上呈现统一的与相符的性能。各核位于具有一种或多种与之关联的资源的网络节点上。这些核动态地实时互相定位以构成与维护支持虚拟的无限数量的独立运行的核的分层结构。本系统不管传送协议的数目与组合及物理拓扑的底层混合始终保持其网络的逻辑视图及用户开发的程序资源。本系统的通信服务利用动态范围桥接在不能共享公共的传送协议栈的端节点之间通信，借此允许位于不同栈上的应用自动与透明地互相通信。

本系统设计成支持包含话音、声音、静止与活动图象、大型文件传送、传统的事务处理及诸如“群件”（“groupware”）应用所需要的任何人对任何人会议在内的一切形式的数字化通信。本系统还设计成在任何类型的网络协

议与媒体上操作其中包含I S D N、X.25、T C P /I P、S N A、A P P C、A T M等。在所有情况中，本系统提供基层媒体的高百分比，通常为60 - 95 %，的理论传输容量，即带宽。

当新资源参加（或重新参加）网络时，位于各节点上的核并从而连接在该节点上的各种资源便自动与立即成为使用本系统的所有应用可以访问的。任何节点在使用的管理层（诸如区域管理员、地域管理员、网络管理员等）内所担任的角色是任意的，即任何节点都可在层内担任一种或多种角色，并且担任一种角色既不要求也不排除担任任何其它角色。再者，角色根据网络的要求，即在一个或多个节点进入或离开网络时，动态地改变。从而，单个的核动态地互相定位及协商相关节点在管理网络层中所扮演的角色，而不考虑它们的物理位置。此外，任何节点可以担任的可能角色或级别的数目是不受限制的，并可根据网络环境的具体要求选择。

通过参阅下面结合附图的较佳实施例的详细描述将会更好更全面地理解本发明的上述与其它特征与优点，其中：

图1 为展示按照本发明的分布式计算系统的图。

图2 为图1 的系统中的节点之一的详细框图。

图3 为展示按照本发明的核的结构框图。

图4 为本发明的P I P E S 逻辑网络（P L N）的流程图。

图5 为按照本发明的子女注册过程的流程图。

图6 为按照本发明的双亲注册过程的流程图。

图7 为展示按照本发明的不同节点之间的注册通信的图。

图8 为按照本发明的角色调用过程的流程图。

图9 为展示按照本发明的不同节点之间的角色调用通信的图。

图1 0 为按照本发明的子女监视过程的流程图。

图1 1 为按照本发明的双亲监视过程的流程图。

图1 2 为展示按照本发明的不同节点之间的“心博”监视通信的图。

图1 3 为按照本发明的选举进程的流程图。

图1 4 为展示按照本发明的不同节点之间的选举通信的图。

图1 5 为按照本发明的注销进程的流程图。

图1 6 为展示按照本发明的不同节点之间的注销通信的图。

图1 7 为展示关于本发明的资源的活动的图。

图1 8 为按照本发明的“增加资源”进程的流程图。

图1 9 为按照本发明的“寻找资源”进程的流程图。

图2 0 为本发明的区域管理员节点上的“寻找资源”进程的流程图。

图2 1 为按照本发明在区域管理员以上级别上的“寻找资源”进程的流程图。

图2 2 为在本发明的区域管理员节点上的“持续寻找”进程的流程图。

图2 3 为按照本发明在区域管员以上级别上的“持续寻找”进程的流程图。

图2 4 为本发明的区域管理员节点上的“清除持续寻找”进程的流程图。

图2 5 为按照本发明在区域管理员以上级别上的“清除持续寻找”进程的流程图。

图2 6 为按照本发明的在区域管理员故障时的“资源恢复”进程的流程图。

图2 7 为按照本发明在另一管理节点故障时的“资源恢复”进程的流程图。

图2 8 为按照本发明的“清除资源”进程的流程图。

图2 9 A 展示本发明的上下文桥接的组成部分。

图2 9 B 为说明上下文桥接在不同协议之间通信中的使用的实例。

图3 0 为展示按照本发明的上下文桥接路由进程的流程图。

图3 1 为按照本发明的“路由发现”进程的流程图。

图3 2 为按照本发明的“路由证实”进程的流程图。

图3 3 为按照本发明的“路由广告”进程的流程图。

图3 4 为展示本发明的P I P E S 逻辑网络中的级别数目改变中执行的步骤的流程图。

图1 示出按照本发明的分布式计算系统1 0 0 。本申请的受让人的系统1 0 0 的实现称作P I P E S 平台（“P I P E S ”）。在系统1 0 0 中，两个节点，节点1 （示出为框1 ）与节点2 （示出为框2 ），通过一条网络连接线（示出为线2 7 ）通信。对于熟悉本技术的人员显而易见连接在网络2 7 上的节点数目不限于两个。

这些节点的结构基本上是相同的。从而只详细描述了节点之一，诸如节点1 。三个应用，应用A （示出为框2 ）、应用B （示出为框3 ）及应用C （示出为框4 ），在节点1 上运行。这些应用通常是由应用开发者编写以在P I P E S 上运行的。P I P E S 软件包含用于与应用A - C 通信的一具P I P E S 应用程序员界面（“P A P I ”）（示出为框6 ）。P A P I 6 通过进程间通信（I P C ）功能调用（示出为框7 ）发送报文给在节点1 上执行的单个P I P E S 核（示出为框9 ）。核9 通过传送装置驱动器T D 1 （示出为框1 1 ）、T D 2 （示出为框1 2 ）与T D 3 （示出为框1 3 ）在网络2 7 上发送与接收报文。

类似地，节点2 具有三个应用在其上运行，应用X （示出为框1 5 ）、应用

Y (示出为框1 6) 与应用Z (示出为框1 7) , 并通过P A P I (示出为框1 8) 与I P C (示出为框1 9) 与在节点2 上运行的一个单个P I P E S 核 (示出为框2 1) 通信。节点2 支持三种不同的网络协议, 从而包含三个传送驱动器T D 3 (示出为框2 4) 、T D 4 (示出为框2 5) 及T D 5 (示出为框2 6) 。

例如, 如果节点1 上的应用A 要求与节点2 上的应用Z 通信, 便从应用A 通过P A P I 6 、I P C 7 及核9 发出一则报文。核9 利用其传送驱动器T D 3 在网络2 7 上将报文发送到节点2 上的传送驱动器T D 3 。然后将报文传递给节点2 上的核2 1 、I P C 1 9 、P A P I 1 8 并最终到应用Z 。

P I P E S 还提供其所有部件使用的通用服务。网络管理服务 (示出为框1 0 与2 0) 提供P I P E S 网络管理代理人 (未示出) 的访问来监视核的网络与系统级计数器、属性及统计。通用服务 (示出为框8 与2 2) 提供核9 与2 1 对操作系统服务的一个通用界面, 这些服务中包含散列、b 树、地址处理、缓冲器管理、排队管理、登录、定时器及任务调度。依赖于系统的服务 (示出为框5 与2 3) , 提供对节点上的操作系统、平台、环境与传送专用的服务。通用服务 (示出为框8 与2 2) 利用这些服务在给定的操作系统或平台环境内实现通用服务。

图2 示出系统1 0 0 的节点1 内的P I P E S 内部结构的更详细的框图。P I P E S 结构分成三个不同的层: 界面层 (示出为框2 8) 、核心层 (示出为框2 9) 与传送层 (示出为框3 0) 。界面层2 8 处理来自与应答正在通过P A P I 6 访问P I P E S 环境的应用的查询。界面层2 8 是以链接在访问核9 的各应用 (诸如应用A - C) 的库实施的。核心层2 9 向正的访问P I P E S 的应用提供程序资源与通信管理服务, 允许不共用一个传送协议栈的端节点之间的通信。传送层3 0 包含节点1 支持的网络协议的传送装置驱动器1 1 、1 2 与1 3 。各传送驱动器提供从核9 对其它供应商提供的网络传送协议, 诸如T C P / I P 、S N A 、I P X 或D L C , 的访问。传送层3 0 为给定的传送规定处理给定的平台上的传送专用的A P I 发布。

图3 示出核9 的内部结构。该9 包含一个A P I 界面 (示出为框3 1) , 这是对图2 的P A P I 6 的界面。A P I 界面3 1 处理来自界面层2 8 的请求并返回对这些请求的应答。它识别应用的优先级并根据这一优先级排队应用的报文。A P I 界面3 1 还处理来自资源层 (示出为框3 2) 及对话服务 (示出为框3 5) 的应答, 并确定这些应答的路由到适当的应用上。

资源层3 2 登记P I P E S 逻辑网络 (“P L N”) 层 (示出为框3 3) 内的应用的资源, 提供寻找P I P E S 内其它P A P I 资源的能力, 以及处理网络内资源的注销。此外, 资源层3 2 实现“坚持寻找”能力, 它启动尚未登记在P L N 3 3 中的资源的定位。

P L N 3 3 维护P I P E S 内的节点的逻辑的层次关系来强加一个动态管理框架。P L N 3 3 处理管理员选举, 作为物理网络故障的后果的管理层次的透明重建。P L N 3 3 利用“心博”报文系统来监视网络内节点的状态及识别网络故障, 这一层还处理请求及返回应答给资源层3 2 与确认的数据报服务 (“A K D G”, 示出为框3 4) 。

A K D G 3 4 在故障时用重新发送来为用户提供最大努力的数据报服务。A K D G 3 4 通过无线报文传送服务 (C L M S) 3 6 与对话服务3 5 处理报文发送与接收。

对话服务3 5 为用户分配、管理与重新分配对话。对话管理包括按顺序发送与接收用户发送的数据、保证对话的安全使用及在面向连接的报文传送服务 (C O M S) 流协议上保持报文语义。对话服务3 5 还在P A P I 应用拥有的对话上多信道广播P A P I 应用报文。对话服务3 5 与C O M S 3 7 交互作用来满足来自A K D G 3 4 与A P I 界面3 1 的请求。

C L M S 3 6 传送没有投送保证的数据。它也与上下文桥接层3 8 交互作用来满足来自A K D G 3 4 的请求。

C O M S 3 7 管理对话服务3 5 打开的连接。C O M S 3 7 提供高性能数据传送, 包括为用户分段与重组报文。C O M S 3 7 根据连接端点之间的接力段的最大报文长度修正报文长度。

上下文桥接层3 8 通过在多个网络传送上执行动态传送协议映射将P A P I 应用与基层网络隔绝, 从而即使在端对端协议不同时也能启动数据传送。

传送驱动器界面 (示出为框3 9) 处理传送专用驱动器与C L M S 3 6 及C O M S 3 7 层之间的通信。这一界面中包含所有传送驱动器的通用公共码。

P L N 层

P L N 3 3 是系统管理员强加在一组执行核的机器上的一种分层结构。这些核在运行时间统一形成一个带管理该层次结构的给定的级的动态选举的管理员的分层网络。P L N 各空间分成五个不同的级：正常、区域、组群、地域及网络。起动时所有核都具有正常特权。它们取决于它们在网络中的配置及诸如已担任的角色的数目等实时考虑而担任管理角色。从而，将管理功能平均分布在成员核中间，而导致更好的性能与较快的恢复。应理解级的数目不限于5种，本系统中能实现任何数目的级，如下面说明的。

在P L N 3 3 中，网络管理员与区域管理员之间的各种管理员（诸如地域管理员与组群管理员）所扮演的主要角色基本上是相同的：保持与其双亲及子女的通信，以及确定资源层3 2 通信量的路由。除了这些功能之外，网络管理员与区域管理员之间的任何管理员（诸如地域或组群）还提供坚持寻找源高速缓冲存储服务，如下面结合图2 2 与2 3 描述的。除了上述这些功能之外，区域管理员还提供其子女广告的资源的高速缓冲存储服务，其子女中包括该区域管理员的各空间中的所有核。因此，区域管理员对于P L N 3 3 的有秩序的工作起决定性作用，P L N 是通过在分层结构中任何其他角色之前填补区域管理员角色的基础上建立起来的。通过系统设定，任何核都能成为区域管理员。

如图4 中所示，P L N 建立与维护算法包括五个主进程：注册（示出为框1 0 0）、角色调用（示出为框2 0 0）、监视（示出为框3 0 0）、选举（示出为框4 0 0）、及注销（示出为框5 0 0）。在这一说明中，为了能够适当抽象而采用下述名词。P L N 3 3 中的级数是用M i n l e v e l 与M a x l e v e l 定义的。具有正常特权的核是配置在M i n l e v e l 上而不在管理员级上的。反之作为网络管理员的一个核是配置在M a x l e v e l 上并具有成为网络的根的潜力。配置参数M a x S t a t u s 在该核能成为管理员的最高级别上施加一个上限。级n 上的核称作级n + 1 上其双亲的子女，假定这两个核在级n 以上具有相同的名。

注册

图5 与6 描述在P L N 3 3 中的子女与双亲节点上执行的注册过程。注册是子女核 用双亲核来定位与登记的进程。图7 说明在节点N 7 （示出为圆圈3 7 并称作核N 7 ）中的核运行该注册进程来进入网络的注册进程的假想执行期间核之间所传递的报文。

一个核通过运行注册进程来定位其双亲核而进入网络。子女核首先进入等待时段（步骤1 0 1），在其中该子女收听网络上其它注册广播（步骤1 0 2）。如果在等待时段中接收到注册广播（步骤1 0 3），子女核便读取报文。报文中的信息足以使该子女确定其双亲与兄弟姐妹的身份。如果报文的始发者为兄弟姐妹（步骤1 0 4），为了防止注册广播淹没网络，子女核修改其等待时段间隔（步骤1 0 5）。如果报文的始发者为双亲（步骤1 0 6），该子女核便发送一个注册请求给双亲（步骤1 0 7）并等待确认。如果未接收到注册广播，子女核继续收听注册广播直到等待时段结束为止（步骤1 0 8）。等待时段结束时，子女核在网络上发送一个注册广播（步骤1 0 9）。

图7 中，核N 7 通过发送一则注册广播报文（用虚线a 表示）到节点N 1 中的核（用圆圈3 1 表示并称作核N 1）、节点N 2 中的核（用圆圈3 2 表示并称作核N 2）、节点N 3 中的核（用圆圈3 3 表示并称作核N 3）、节点4 中的核（用圆圈3 4 表示并称作核N 4）、节点N 5 中的核（用圆圈3 5 表示并称作核N 5）及节点N 6 中的核（用圆圈3 6 表示并称作核N 6）来试图注册到P I P E S 网络上。该子女核等待一个规定的时间来接收注册确认（步骤1 1 0）。

所有的核都在网络上收听注册广播报文（步骤1 1 6）。如果接收到一则注册广播（步骤1 1 7），双亲核判定发送该报文的核是否为其子女（步骤1 1 8）。如果始发核不是其子女，双亲核继续收听注册广播（步骤1 1 6）。然而，如果始发核是其子女，双亲核检验该核是否为双重子女（步骤1 1 9）。如果是双重子女，双亲通知其双重子女角色冲突（步骤1 2 0）。如果否，双亲便发送一个注册确认给其子女核（步骤1 2 1）。

图7 中，双亲核N 4 收到核N 7 的注册广播报文a，并发送一则用线b 表示的注册确认报文给核N 7。

如果接收到注册确认（步骤1 1 0），子女核便发送一个注册证实给发送注册确认的第一个双亲核（步骤1 1 4）。子女核忽略它可能收到的任何其它注册确认。发送了注册证实给其双亲之后，该子女核便开始与其新双亲的监视进程（步骤1 1 5）。如果双亲核收到注册证实（步骤1 2 2），双亲核便登记该子女（步骤1 2 3）并开始与其新子女的监视进程（步骤1 2 4）。如果双亲核未收到来自子女的注册证实（步骤1 2 2），双亲核便继续收听注册广播（步骤1 1 6）。

图7 中, 接收到双亲核N 4 的注册确认b 之后, 子女核N 7 便发送一个用线C 表示的注册证实报文给核N 4 并开始与其双亲核N 4 的监视进程。

如果没有双亲核发送注册确认给子女, 除非已经超过了再试阈值 (步骤1 1 1), 子女核再一次开始注册进程 (步骤1 0 1)。如果已经超过了再试阈值, 子女检验其Max Status 设定值 (步骤1 1 2)。如果子女的Max Status 大于Min level, 该子女便开始角色调用进程来担任其自己的双亲的角色。否则, 子女核再一次进入注册等待周期 (步骤1 0 1)。

角色调用

角色调用是一个核用来查询网络以寻找出名空间层次结构中的空缺的过程。该过程由已配置了具有大于Min level 的Max Status 的所有的核执行的。角色调用过程由核在启动时及随后在其名空间中存在管理空缺时调用。角色调用算法设计成使同时参加角色调用进程的核的数目最小, 减少全网络广播以及同一空缺的潜在竞争者之间的可能冲突。

角色调用过程示出在图8 中。想要参加角色调用的一个核通过一个强制的等待时段 (步骤2 0 1)。等待时段是该核已经担任的角色的数目、该核是否一个活跃的上下文桥接及该核的当前状态的函数。还将一个随机等待间隔加在公式上。

在等待时段中, 核收听来自其它核的角色调用 (步骤2 0 2)。如果收到相同层次级的角色调用广播 (步骤2 0 3), 该核便放弃角色调用过程 (步骤2 0 4)。如果未收到角色调用广播, 该核继续收听角色调用广播 (步骤2 0 2) 直到等待时段结束 (步骤2 0 5)。等待时段结束时, 核在网络上发送其本身的色色调用广播 (步骤2 0 6)。广播报文中包含所请求的角色调用的层次级。发送了角色调用广播之后, 该核起动一个定时器 (步骤2 0 7) 并在网络上收听角争调用广播 (步骤2 0 8)。作为请求的角色调用的名空间的管理员的核将用点对点角色调用确认报文来应答。如果发动角色调用的核收到这一确认 (步骤2 0 9), 该核将放弃角色调用过程 (步骤2 0 4)。如果发动角色调用的核接收到同一层次级的另一个角色调用广播 (步骤2 1 0), 该核读取这一报文。如果报文的始发者具有更高的凭证 (步骤2 1 1), 该核将放弃角色调用过程 (步骤2 0 4)。特定的核的凭证为该核已担任角色的数目、该核是否是一个活跃的上下文桥接及该核的当前状态的函数。在暂停时段结束时 (步骤2 1 2), 该核便担任它请

求角色调用的空缺管理角色（步骤2 1 3）。

图9 描述角色调用过程的一个实例。用圆圈4 4 表示的核N 4 由于物理连接问题而成为从网络上孤立的。用圆圈4 7 表示的核N 7 作为与其双亲N 4 的监视进程（下面详细描述）的结果检测到核N 4 的缺席。核N 7 进入强制等待时段并收听网络上的角色调用广播通信量。如果用圆圈4 5 表示的核N 5 已在核N 7 之前启动了其角色调用进程，核N 7 在收到用虚线i 表示的核N 5 的角色调用广播报文之后便中止其角色调用。然而，假定核N 7 先启动其角色调用，在角色调用等待时段结束时核N 7 发送出用虚线n 表示的广播报文。

如果核N 5 在核N 7 已经这样做之后发出其自己的角色调用广播报文，核N 7 将其凭证与核N 5 的比较。如果核N 5 的凭证较高，核N 7 放弃角色调用而核N 5 担任由核N 4 的消失留下空缺的管理角色。如果核N 7 的凭证较高，核N 5 放弃角色调用而核N 7 在暂停时段结束时担任核N 4 的空缺管理角色。

如果N 4 重新出现在网络上并接收到了核N 5 的广播报文i 或核N 7 的广播报文h，核N 4 便用发送一则用线j 表示的确认报文给核N 5 或用线k 表示的确认报文给核N 7。如果核N 4 未重新出现在网络上，核N 5 与核N 7 继续它们的角色要求进程。

监视

图1 0 与1 1 描述子女与双亲监视进程，它用于互相保持接触。

双亲具有设定在所有它的子女的最低心博间隔上的本身的“心博”定时器。在监视进程开始时，双亲初始复位其心博定时器（步骤3 1 2）并收听来自其子女的心博报文（步骤3 1 3）。与其双亲参加监视进程的子女首先发送一个心博报文到其双亲（步骤3 0 1）并等待确认。如果双亲收到心博报文（步骤3 1 4），双亲将发送一个心博确认给该子女（步骤3 1 5）并在其子女表中检验该子女（步骤3 1 6）。确认报文中包含一个心博偏移值以在其子女间散布心博间隔。如果该子女收到心博确认（步骤3 0 2），该子女便修正其心博间隔（步骤3 0 6）并进入等待时段（步骤3 0 7）。如果在这一时间上收到心博确认（步骤3 0 4），则该子女修正其心博间隔（步骤3 0 6）并进入等待时段（步骤3 0 7）。如果该子女仍未收到心博确认，该子女假定它已成为孤儿而起动注册进程（步骤3 0 5）。

当双亲的心博定时器到时 (步骤3 1 7), 该双亲在其子女表中检验缺少的心博报文 (步骤3 1 8)。如果双亲检测到缺少的心博, 该双亲发送心博报文给该缺失的子女 (步骤3 1 9)。如果该双亲没有收到来自缺失的子女的心博确认 (步骤3 2 0), 该双亲便注销这一子女 (步骤3 2 1)。

在其等待时段中 (步骤3 0 7), 子女收听来自其双亲的心博报文 (步骤3 0 8)。如果子女收到心博报文 (步骤3 0 9), 该子女发送一个心博确认给其双亲 (步骤3 1 0), 修正其心博间隔 (步骤3 0 6) 并再度进入等待时段 (步骤3 0 7)。等待时段结束时 (步骤3 1 1), 子女再一次启动监视进程 (步骤3 0 1)。

图1 2 示出在监视进程中双亲与子女之间传递的周期性报到报文或“心博”。图1 2 中, 核N 3 与N 4 (分别用圆圈5 3 与5 4 表示) 为核N 2 (用圆圈5 2 表示) 的子女。核N 2 又依次为核N 1 (用圆圈5 1 表示) 的子女。报文d 1 至d 3 表示从子女到双亲的心博报文, 而报文e 1 至e 3 表示双亲到子女的心博确认。报文f 1 至f 3 表示从双亲到子女的心博报文, 而报文g 1 至g 3 表示从子女到双亲的心博确认。

选举

发生角色冲突时, P I P E S 核进行分布式选举 (图1 3) 来确定获胜者。当在基层物理连接中存在导致网络分裂的问题时, 两个或更多管理员可能要求同一空间上的管理职能。名空间中的冲突主要是通过上述角色要求或注册广播之一检测到的。当一个核检测到名空间冲突时, 它将通知负责人, 后者便执行选举进程。新的参加者可以参加已在进行中的选举。因为选举完全是分布式的, 各核独立地进行选举并得出结果。

当一个核检测到角色冲突或接收角色冲突的通知时, 该核便通过启动选举定时器及打开选举数据库来启动选举进程 (步骤4 0 1)。该核存储到目前为止已知的选举参加者, 并向各参加者发送选举请求 (步骤4 0 2)。这一报文中包括所有已知的参加选举的核。然后该核收听网络上的任何选举通信量 (步骤4 0 3)。如果该核收到包含已知参加者的表的选举应答 (步骤4 0 4), 该核便将任何新的选举参加者存储在数据库中并发送给各人一个选举请求 (步骤4 0 2)。如果收到另一选举请求 (步骤4 0 5), 该核发送一个选举应答给始发者 (步骤4

06), 更新选举数据库, 并发送选举请求给新参加者(步骤402)。在选举定时器到时(步骤407), 该核查询其选举数据库来确定获胜者(步骤408)。一次选举的获胜者取决于各参加的核已担任的角色的数目、参加的核是否活跃的上下文桥接及各核的当前状态。如果该核是选举的获胜者(步骤409), 该核便发送选举结果报文给所有选举参加者(步骤410)。如果该核选举失败, 该核便辞去其作为管理员的职位(步骤411), 将它们的新双亲通知其所有子女。所有选举参加者验证选举结果并最终关闭它们的选择数据库(步骤412)。

图14说明选举进程的一个实例。假定核A与B(分别用圆61与62表示)已独立检测到角色冲突。核A将发送一个选举请求报文(箭头L)给核B。这一报文中将包含核A已知的参加者, 在这一点上正好是核A与B。当核B收到这一报文时, 核B将一个选举应答报文发送给核A(箭头m)。此后, 核C检测到与核B的角色冲突。这时核C将发送一个选举请求报文(箭头n)给核B。核B将用新加入的核C更新其选举数据库并送回一个选举应答报文(箭头o)给核C。这一报文中将包含这一点上核B已知的选举参加者, 即核A、B与C。当C收到这一报文时, 它检测到新的竞争对手核A, 更新其选举数据库, 并发送一个选举请求报文(箭头p)给核A。在这一点上, 核A将知道新的竞争对手(从其观察), 用核C的凭证更新其数据库, 并回答核C的请求(箭头q)。以相同的方式, 当只知核A的核D进入选举时, 它将立即知道核B与C。

注销

注销(图15与16)为核用来从其双亲注销的过程。注销可以作为核关闭逻辑的一部分或作为辞去作为层次的特定级别的管理员的结果来启动。一个子女核(示出为图16中的核N2)发送一个注销请求(用箭头x表示)给其双亲, 在图16中示出为核N1(步骤501)。当双亲收到来自其子女的注销请求时(步骤506), 它便发送一个注销确认(图16中示出为箭头y)给该子女(步骤507)并注销该子女(步骤508)。如果该子女为管理员(步骤503), 该子女将发送报文(在图16中用报文Z1至Z3表示)通知其所有子女(即图16中的核N3、N4与N5)它不再是它们的双亲(步骤504)。此外, 双亲核将通过指定它在其子女上执行的选举进程的获胜者而在其子女中指定一个接班人(步骤505)。

资源层

资源层（图3 中框3 2 ）负责管理分布在整个P I P E S 网络层次上的所有资源。资源是使网络上其它节点上执行的其它P I P E S 应用可以利用的一个P I P E S 的功能子集。可将P I P E S 资源想象为严格定义的服务元素，其中当将一个或多个元素作为整体考虑时，它们便组合成一个完整的服务。

图1 7 描述P I P E S 中的资源的生命周期。一个资源通过增加资源进程（框6 0 0 ）进入网络。为了利用一个资源所提供的服务，应用必须执行寻找资源进程（框7 0 0 ）来确定其在P I P E S 地址空间内的位置。例如，在执行寻找查询及得到一个可利用的资源的地址之后，应用可以通过对话服务3 5 试图建立与该资源的对话。

如果在应用执行寻找查询时不能得到资源，该应用可替代执行坚持寻找查询，它在一个符合搜索标准的资源通过增加资源进程进入网络时将会立即通知应用资源的可利用性。在这一情况中，P I P E S 的区域管理员保持高速缓冲存储未决的坚持寻找查询以便立即响应这一查询。如果通过物理网络故障一个区域管理员要从P I P E S 层次的其余部分上断开，便利用恢复机制（框8 0 0 ）在接替断开的管理员的职责的新的区域管理员上重建坚持寻找高速缓冲存储器。

在其网络上的生命期间，可利用资源向网络上的应用提供服务。如果拥有该资源的应用将该资源从网络上取走，资源层便执行消除资源进程（框9 0 0 ）。

增加资源进程

图1 8 说明用来将应用的资源引导进P L N 3 3 中的增加资源进程。资源的始发节点首先检验其本地资源数据库来判定带有相同名称的资源是否已经存在（步骤6 0 1 ）。如果是存在这一资源，始发节点返回一个出错给用户应用（步骤6 0 2 ）。如果不存在该资源，始发节点在其本地数据库中为该资源增加一个项目（步骤6 0 3 ）。然后该资源检验其坚持寻找查询高速缓冲存储器来判定正在该节点上执行的应用是否正在等待资源（步骤6 0 4 ）。如果新资源与坚持寻找高速缓冲存储器中的任何搜索标准匹配，则始发节点将新资源的属性发送给启动坚持寻找查询的始发用户的应用（步骤6 0 5 ）。这时始发节点从高速缓冲存储器中消除搜索标准与新资源匹配的坚持寻找查询（步骤6 0 6 ）。如果新消除的坚持寻找查询的范围大于机器级（步骤6 0 7 ）则始发节点发送一个清除坚持寻找查询到其双亲节点（步骤6 0 8 ）。在坚持寻找处理结束时，或者如果没有新资源匹配的坚持寻找查询，始发节点发送一个增加资源请求到其双亲区域管

理员 (步骤6 0 9) 。

如果区域管理员收到来自其子女之一的一个增加资源请求 (步骤6 1 0) , 该区域管理员便将该资源增加到其本身的独立资源高速缓冲存储器中 (步骤6 1 1) 。然后区域管理员检验其本身的坚持寻找高速缓冲存储器来判定这一新资源是否与高速缓冲存储器中任何查询标准匹配 (步骤6 1 2) 。如果是, 区域管理员发送这一资源的属性给始发坚持寻找查询的节点 (步骤6 1 3) 并从其坚持寻找高速缓冲存储器中清除该查询 (步骤6 1 4) 如果查询范围大于区域级 (步骤6 1 5) , 则区域管理员发送一个清除支持寻找查询给其双亲组群管理员 (步骤6 1 6) 。

寻找资源进程

在P L N 3 3 内搜索资源的应用可为它发送给P I P E S 核的寻找查询指定三种不同选择之一: 寻找、寻找下一个或坚持寻找。寻找查询将在本地机器上开始搜索资源, 如果在机器级上找不到资源便移到区域级上, 以此类推直到P I P E S 网络层次。如果在一个特定级上找到资源, 便将该资源的属性发送到请求资源的应用。如果应用稍后发布寻找下一个查询, 便从P I P E S 层次内前一搜索停止处继续搜索。

如果用户发布一个坚持寻找查询, 始发节点首先将其转换成正常寻找查询, 它象任何其它寻找查询一样通过网络。如果将任何资源返回给该用户, 则寻找查询不在网络内继续进行; 然而如果在P I P E S 层次内未找到资源, 则将坚持寻找查询存储在P I P E S 层次内的区域管理员的坚持寻找高速缓冲存储器中。

图1 9 描述在始发节点上执行的寻找资源进程。如果启动了寻找或坚持寻找查询, 始发节点便清除用作缓冲器的资源高速缓冲存储器来存储满足该查询的搜索标准的资源属性 (步骤7 0 1) 。由于寻找查询是完全由该查询的始发者调整的, 而且在任何中间节点上不维护状态, 各查询数据包必须携带充分的信息使中间节点能进行它们的搜索。最重要的信息中的一些为始发节点在网络内的位置、始发节点要求的最大匹配数 (M a x M a t c h e s) 、已返回给始发节点的当前匹配数 (C u r r M a t c h e s) 、搜索范围 (S c o p e) 、最后进行搜索的级别 (L e v e l) 、以及在该级别是最后搜索的状态 (L e v e l S t a t u s) 。当以寻找查询或坚持寻找查询开始搜索时, 始发节点初始化这些变量中的一些以在机器级上开始搜索 (步骤7 0 2) 。由于寻找下一个查询是设计成

在上一搜索停止处开始下一个搜索的，寻找下一个查询导致始发节点跳过这些初始化步骤。

始发节点将CurrMatches与MaxMatches进行比较来判定用户是否已收到它所要求的最大数目的匹配（步骤703）。如果CurrMatches不等于MaxMatches（CurrMatches永远不能超过MaxMatches），则始发节点检验其资源来判定是否还有任何更多的可利用资源可以返回给该用户（步骤704）。由于虽然分布式寻找查询可能返回一个以上的资源给始发节点，但始发节点是一次一个地返回资源给用户的，所以在本地高速缓冲存储器中可能剩有资源。如果在本地高速缓冲存储器中剩有资源，始发节点将第一个资源返回给用户（步骤705）。如果资源高速缓冲存储器是空的，始发节点检验LevelStatus来判定最后搜索停止的地方（步骤707）。如果在该级别上没有可利用的资源，便将LevelStatus设置成EOF（即寻找结束）。如果LevelStatus为EOF，始发节点将CurrLevel增加一而在层次的下一级上继续搜索（步骤710）。如果LevelStatus不是EOF，始发节点检验CurrLevel来判定在开始分布式搜索之前是否先在本地机器上开始搜索（步骤708）。如果CurrLevel设定为机器，始发节点搜索其本地资源数据库来判定本地资源是否有可能与搜索标准匹配（步骤709）。如果得到一个本地资源，始发节点将到MaxMatches资源的属性复制到查询的资源高速缓冲存储器中，并将CurrMatches设定为找到与复制进高速缓冲存储器中的匹配数（步骤706）。然后始发节点将高速缓冲存储器中的第一个资源返回给请求资料的用户（步骤705）。如果未找到本地资源，始发节点将LevelStatus设定为EOF（步骤711），然后将CurrLevel增加一，在下一级上继续搜索（步骤707）。

如果CurrLevel超过了MaxLevel（步骤712）或Scope（步骤716），则搜索或者已通过整个PIPES层次工作过或者超过了原始查询的范围。从而，如果符合了这些条件之一，搜索便完成了。如果否，始发节点将寻找查询发送给其双亲，区域管理员，以开始分布式搜索（步骤713）。如果在应答中返回了资源属性（步骤714），始发节点复制这些资源属性到查询的资源高速缓冲存储器中（步骤718）并返回第一个给用户（步骤717）。如果搜索不成功地结束，始发节点检验CurrMatches来判定是否已将任何资源返回给用户（步骤715）。如果CurrMatches大于零，则用户已收到其全部资源，而始发节点返回一个EOF给用户（步骤723）。

如果CurMatches 为零, 并且在网络上未找到资源, 如果用户已指定了坚持寻找查询, 始发节点分发一个坚持寻找查询 (步骤7 1 9)。这是为了跟踪坚持寻找查询的资源而必须将该查询加到该节点上的未决坚持寻找查询表中 (步骤7 2 0)。如果存在着本地机器上的一个资源有可能与查询的搜索标准匹配 (步骤7 2 1), 始发节点将查询加到其坚持寻找高速缓冲存储器中 (步骤7 2 2), 它是用来跟踪搜索标准的, 以便在满足这些标准的资源加入PIPES 中时便立即将它们返回。如果查询范围大于机器级 (步骤7 2 4), 则将坚持寻找查询送至区域管理员 (步骤7 2 5)。

图2 0 与2 1 说明资源层如何在整个PLN3 3 中确定寻找查询的路由。图2 0 示出在区域管理员级上执行的进程。当区域管理员收到一个寻找查询时 (步骤7 2 6), 区域管理员检验CurLevel 来确定请求搜索的级 (步骤7 2 7)。如果CurLevel 小于区域 (步骤7 2 8), 则如果区域管理员错误接收该查询而返回一个出错给发送寻找查询的节点 (步骤7 2 9)。如果CurLevel 大于区域 (步骤7 2 8), 则如果区域管理员从其子女之一接收该寻找查询 (步骤7 3 1), 便将该寻找查询转交给其双亲 (步骤7 3 2)。从而, 区域管理员只是向上传递寻找查询, 因为搜索应在层次的较高级别上继续。如果搜索应在这一级上继续, 区域管理员分析搜索标准来判定在本区域中是否有资源能满足这些标准 (步骤7 3 0)。如果否, 区域管理员返回寻找查询给发送者 (步骤7 3 8)。此外, 如果CurMatches 已经等于MaxMatches (步骤7 3 3), 区域管理员也将寻找查询返回给发送者 (步骤7 3 8)。否则, 区域管理员搜索其资源数据库寻找始发节点可见的匹配 (步骤7 3 4)。增加资源到PIPES 的用户能指定哪些应用能利用其服务, 或其在PIPES 内的“能见度”。如果找到可见的匹配, 便将最多的MaxMatches 资源属性复制到寻找查询 (步骤7 3 5)。如果找到比MaxMatches 多的资源 (步骤7 3 7), 区域管理员将LevelStatus 设定为OK (步骤7 3 9), 从而在下一一次发布寻找下一个查询时能在这一级上继续搜索。否则, 区域管理员将LevelStatus 设定为EOF 来通知始发节点在这一级上得不到更多的资源 (步骤7 3 6)。最后, 区域管理员将寻找查询返回给发送者 (步骤7 3 8)。

高于PLN 层次中区域管理员的管理级上的寻找查询进程 (图2 1) 类似于区域管理员级上的, 只是不出现搜索, 因为只有机器与区域管理员拥有资源数据库。图2 1 中步骤7 4 0 至7 4 7 与图2 0 中步骤7 2 6 至7 3 3 相同。在各种情况中, 节点判定搜索应在这一级还是较高级上继续。在这一情况中, 在这一级

上的搜索包含依次将寻找查询转交给管理员的各个子女。如果还有未曾见到寻找查询的子女（步骤7 4 8），管理员将寻找查询发送给下一子女（步骤7 4 9）。当没有剩下子女时，管理员将Level Status 设定为E O F（步骤7 5 1）并返回寻找查询给发送者（步骤7 5 0）。

图2 2 与2 3 说明在整个网络上增加一个坚持寻找查询的进程，而图2 4 与2 6 描述用来从网络上消除一个坚持寻找查询的类似的“清除”进程。图2 2 中，区域管理员节点处理在P L N 3 3 上收到的一个坚持寻找查询（步骤7 5 2）。首先，如果区域管理员收到来自其子女之一的查询（步骤7 5 3），便将该查询加在其未决坚持寻找的源表上（步骤7 5 4）。如果这一区域中的一个资源能满足坚持寻找查询的搜索标准（步骤7 5 5），区域管理员便将该查询加入其坚持寻找高速缓冲存储器。如果查询的范围大于区域级（步骤7 5 7），区域管理员将坚持寻找查询发送給其双亲（步骤7 5 8）。类似地，在图2 3 中，在高于区域的级上的管理员收到一个坚持寻找查询（步骤7 5 9）。如果发送者是该管理员的子女之一（步骤7 6 0），管理员便将查询加在其未决支持寻找的源表上（步骤7 6 1）。如果这一级是在查询中指定的搜索标准之内（步骤7 6 2），管理员将查询转交给其子女（除外发送该查询的子女）（步骤7 6 3）。如果查询的范围大于这一级（步骤7 6 4），则管理员将坚持寻找查询发送給其双亲（步骤7 6 5）。

通过从未决坚持寻找的源表中清除它们（步骤7 6 8 与7 7 5）及从区域管理员的坚持寻找高速缓冲存储器中清除它们（步骤7 7 0）来“清除”坚持寻找查询的类似进程示出在图2 4 与2 5 中。

坚持寻找恢复进程

由于有关分布式坚持寻找查询的重要信息保存在区域管理员节点上，而在其它管理节点上范围较小，当这些节点之一破坏或从P L N 层次的其它部分上断开时必须使用恢复进程。图2 6 与2 7 表示当区域管理员（图2 6）或另一管理节点（图2 7）关闭时用来提供恢复的进程。

当一台机器注册到用选举进程选择的其新的双亲区域管理员上时，该子女机器将其未决坚持寻找的源表发送給其新双亲（步骤8 0 0）。新区域管理员接收这一表（步骤8 0 1）并利用从其子女接收的信息更新其自己的未决坚持寻找源表（步骤8 0 2）。然后新区域管理员发送一个添加高速缓冲存储器请求給其双

亲（步骤8 0 3）。其它管理员收到这一请求（步骤8 0 5）并将其发送给未决坚持寻找的管理员的源表中的其所有子女（步骤8 0 6）。如果发送者是该管理员的子女（步骤8 0 7），该管理员在P L N 层次中将请求向上送到其双亲（步骤8 0 8）。最终，P L N 3 3 中的其它区域管理员接收到该添加高速缓冲存储器请求（步骤8 0 9），如果新区域管理员的坚持寻找高速缓冲存储器中有一个查询（步骤8 1 0），接收的区域管理员用来自其坚持寻找高速缓冲存储器的匹配查询来回答该新区域管理员（步骤8 1 1）。然后新区域管理员用来自P L N 3 3 中的其它区域管理员的回答来更新其自身的坚持寻找高速缓冲存储器（步骤8 0 4）。

图2 7 描述当区域管理员以外的管理员关闭时出现的情况。新管理员的子女发送它们的未决坚持寻找源表给该新管理员（步骤8 1 2）。新管理员接收这些表（步骤8 1 3）并用其子女发送的信息更新其未决支持寻找表（步骤8 1 4）。如果任何查询的范围高于这一级（步骤8 1 5），将这些查询在P L N 层次中向上发送到新管理员的双亲（步骤8 1 6）。新管理员的双亲用从其新子女获得的信息验证其未决坚持寻找源表（步骤8 1 7）。

清除资源进程

当一个应用从P L N 层次中撤出其资源时，资源层3 3 便执行图2 8 中所示的清除资源进程。资源的始发节点首先进行检验来查明该资源是否存在在其资源数据库中（步骤9 0 1）。如果该资源存在，始发节点从数据库中清除该资源（步骤9 0 3）。并发送清除资源请求给其双亲区域管理员（步骤9 0 4）。如果否，始发节点返回一个出错给用户（步骤9 0 2）。区域管理员接收该清除资源请求（步骤9 0 5）并从其区域管理员资源高速缓冲存储器中清除该资源（步骤9 0 6）。

上下文桥接层

图2 9 A 说明上下文桥接层3 8 的组成部分。上下文桥接层的主要功能为将协议数据单元（“P D U”）的路由从源节点确定到目的节点的路由进程（框1 0 0 0）。源节点与目的节点可共用一个可确定路由的协议。可确定路由的协议定义为为了到达其目的地允许单独从目的地地址作出关于必须向何处发送P D U 的决定的一种协议。源节点只是将P D U 传送给可确定路由的协议，而由可确定路由的协议自己去通过分析目的地地址确定如何将P D U 送到目的地。从而，不

必要知道用来将P D U从源转送到目的地的中间节点。在P I P E S 内, T C P / I P 与S N A 为可确定路由协议, 而I P X、 N e t B i o s 与D L C 则不是可确定路由协议。

如果源节点与目的节点共用一个不可确定路由协议或者如果源与目的地根本不共用任何协议, 则必须用中间节点来“桥接”源与目的节点。在本例中, 路由进程利用路由信息数据库 (“R I D B”, 示出为框1 4 0 0) 来确定如何确定P D U从源到目的地的路由。R I D B 中包含将P D U的路由确定到一个不可确定路由协议或源节点不支持的协议上所必需的信息。R I D B 中包含两个高速缓冲存储器: 用于不可确定路由协议的一个源路由高速缓冲存储器 (框1 4 0 1), 及用于不同类协议桥接的一个下一跳跃路由高速缓冲存储器 (框1 4 0 2)。源路由高速缓冲存储器通过路由发现进程 (框1 1 0 0) 繁衍并通过路由证实进程 (框1 2 0 0) 证实。下一跳跃路由高速缓冲存储器通过路由广告进程 (框1 3 0 0) 繁衍。

图2 9 B 示出能有利地利用本发明的上下文桥接的系统1 6 0 0。上下文桥接可用来确定由使用不同级别的协议的节点生成的包的路由, 如在国际标准化组织 (“I S O”) 参考模型中所定义的。例如, 系统1 6 0 0 包含分别采用S N A (A P P C) 与D L C 协议的两个节点1 6 1 0 与1 6 3 0。这两种协议在不同的I S O 级上: S N A 在表示级上而D L C 则在数据链路级上。为了确定通过网络1 6 4 0 从节点1 6 1 0 到节点1 6 3 0 的包的路由, 必须使用包含能桥接S N A (A P P C) 与D L C 协议的上下文桥接的节点1 6 2 0。从而, 首先通过路径1 6 4 2 将节点1 6 1 0 生成的包的路由确定到节点1 6 2 0 上, 然后通过路径1 6 4 3 将该包的路由确定到节点1 6 3 0 上。

类似地, 如果希望将节点1 6 1 0 生成的报文的路由确定到采用U D P 协议 (I S O 传送级上) 的节点1 6 5 0 上, 则必须使用包含能桥接S N A 与U D P 协议的上下文桥接的节点1 6 6 0。从而, 首先将节点1 6 1 0 生成的包的路由通过路径1 6 4 5 确定到节点1 6 6 0 上, 然后将包的路由通过路径1 6 4 6 确定到节点1 6 5 0 上。

路由进程

图3 0 描述上下文桥接路由进程的流程图。当源节点的上下文桥接层接收到送到给定的目的节点的P D U时, 源节点观察目的地址来判定目的地是否具有可

确定路由协议（步骤1 0 0 1）。

如果目的地具有可确定路由协议，源节点判定它是否支持与目的地相同的可确定路由协议（步骤1 0 0 2）。如果源与目的地共用同一可确定路由协议，源便利用共用的可确定路由协议的传送驱动器发送P D U到目的地（步骤1 0 0 3）。如果源与目的地不共用同一可确定路由协议，源便在其R I D B下一跳跃路由高速缓冲存储器中搜索到达目的地的路由（步骤1 0 0 4）。然后源节点检验判明R I D B中是否存在一条路由（步骤1 0 0 6）。如果找到路由，源节点将P D U发送到在R I D B中找到的路由指定的中间节点（步骤1 0 0 7）。如果未找到路由，源返回一个出错说明不能到达目的地（步骤1 0 0 9）。

如果目的地具有不可确定路由协议，源在其R I D B源路由高速缓冲存储器中搜索到达目的地的路由（步骤1 0 0 5）。然后源节点检验判明R I D B中是否存在路由（步骤1 0 0 8）。如果找到路由，源节点将P D U发送到在R I D B中找到的路由所指定的中间节点（步骤1 0 0 7）。如果未找到路由，源执行路由发现进程来寻找到达目的地的路由（步骤1 0 1 1）。然后源节点确认路由发现进程是否找到了路由（步骤1 0 1 2）。如果路由发现找到了路由，源节点更新其R I D B源路由高速缓冲存储器（步骤1 0 1 0），并将P D U发送到该路由指定的中间节点（步骤1 0 0 7）。如果未找到路由，源节点返回出错说明目的地不能到达（步骤1 0 0 9）。

路由发现进程

图3 1 描述用到达个别目的地的源路由来更新R I D B源路由高速缓冲存储器时使用的路由发现进程。在需要寻找到达具有不可确定路由协议的目的地路由时，源节点起动路由发现进程。首先，源节点发送一个路由发现包给所有它具有关于它们的信息的活跃上下文桥接（步骤1 1 0 1）。一个节点如果支持一个以上的协议便是一个活跃的上下文桥接；该节点作为在该节点上找到的协议之间的桥工作。网络中的所有节点通过路由广告进程找出活跃的上下文桥接。

收到源节点的路由发现包的上下文桥接首先判定它是否回答包（步骤1 1 0 7）。如果它是回答包，中间节点利用回答包中指定的路由将该包转送回源节点（步骤1 1 1 2）。如果不是回答包，接收该路由发现包的节点将其本身的地址插入包中（步骤1 1 0 8）。然后该节点检验判明它是否该包的最终目的地（步骤1 1 0 9）。如果该节点是该包的最终目的地，端节点便将包的类型改变成回

答（步骤1 1 1 1），并利用路由发现包中指定的路由将包转送回源（步骤1 1 1 2）。如果接收节点不是目的地，中间节点将包转送给它所连接的所有上下文桥接，但除外它原先从其接收该包的上下文桥接（步骤1 1 1 0）。

源节点正在等待是否能收到回答（步骤1 1 0 2）。如果在指定的时间段内收不到回答，源便返回一个出错说明不能到达目的地（步骤1 1 0 3）。如果收到回答，源节点检验是否已经有到达目的地的一条有效路由（步骤1 1 0 4）。如果已有有效路由，源便丢弃回答包（步骤1 1 0 5）。否则，源节点用回答包中指定的路由更新其R I D B 源路由高速缓冲存储器（步骤1 1 0 6）。

路由证实进程

图3 2 示出用来检验包含在R I D B 源路由高速缓冲存储器中的路由的有效性的路由证实进程。源节点发送一个路由证实包给其R I D B 源路由高速缓冲存储器中尚未标记为有效的所有目的节点（步骤1 2 0 1）。然后该源设定一个定时器（步骤1 2 0 2）并收听证实回答（步骤1 2 0 3）。

端节点也收听路由证实包（步骤1 2 0 9）并检验是否收到证实包（步骤1 2 1 0）。如果在指定的时间段内收不到证实包，端节点继续收听路由证实包（步骤1 2 0 9）。如果收到证实包，端节点证实路由证实包中指定的路由（步骤1 2 1 1）并将该包返回给发送者（步骤1 2 1 2）。

源节点检验判定是否已收到证实回答（步骤1 2 0 4）。如果收到证实回答，源节点便在R I D B 源路由高速缓冲存储器中将到达目的地的源路由标记为有效（步骤1 2 0 5）。如果未收到证实回答，源节点检验定时器（步骤1 2 0 6）。如果定时器尚未到时，源节点继续收听证实回答（步骤1 2 0 3）。如果定时器到时，且如果再试阈值尚未超过（步骤1 2 0 7），则源节点复位定时器（步骤1 2 0 2）。如果再试阈值已经超过，源节点便从R I D B 源路由高速缓冲存储器中清除无效的源路由（步骤1 2 0 8）。

路由广告进程

图3 3 示出在每一个上下文桥接与端节点上周期性执行的路由广告进程。各上下文桥接周期性地发送称作路由广告包（“R A P”）的广播报文（步骤1 3 0 1），及各端节点收听R A P 广播（步骤1 3 0 5）。R A P 中最好包含以下

信息：上下文桥接能处理的协议及所需的跳跃数。然后所有上下文桥接及端节点等待到收到RAP广播为止（步骤1302与1306）。如果收到RAP广播，接收广播的节点通过将RAP广播与其RIDB下一跳跃路由高速缓冲存储器进行比较而判定路由信息中是否有变化（步骤1303与1307）。如果需要改变，接收节点便更新其RIDB下一跳跃路由高速缓冲存储器（步骤1304与1308）。

不受限制的级

在本发明的较佳实施例中，PLN层次的级数是不受限制的。图34示出系统100的开发人员（系统开发者）、应用开发者及最终用户为实现大于省缺级数（诸如5）的级数而最好采取的步骤。在编译PIPES核心部分及PAPI库代码时设定某一实现的最大级数。如果希望在它们的PIPES中具有更大的灵活性及层次中更多的级数，则必须定制PIPES核心部分与PAPI库。

系统开发者改变硬编码在软件的标题文件中的Minlevel与Maxlevel参数（步骤1501）。重新编译PAPI库（步骤1502）及PIPES核心部分（步骤1503），并将新的PAPI库与PIPES核心部分分发给应用开发者（步骤1504）。

应用开发者从系统开发者接收这些部分（步骤1505）并对他们自己的PIPES应用进行任何必要的修改（步骤1506）。然后应用开发者与新的PAPI库一起重新编译自己的PIPES应用（步骤1507）并分发新的PIPES应用及PIPES核心部分给最终用户（步骤1508）。

最终用户从应用开发者接收这些部分（步骤1509）并将它们装在PLN中所有节点上（步骤1510）。对其PIPES配置进行任何必要的修改之后（步骤1511），最终用户通过加载PIPES核心部分（步骤1512）及PIPES应用（步骤1513）而最终重新启动系统。在这一点上，最终用户便能实现PLN层次中所要求的级数。

虽然已经用当前认为的较佳实施例描述了本发明，应理解所附的权利要求书不局限于公开的实施例，正好相反，它们旨在覆盖保持本发明的任何新颖特征与优点的修改、变化与等价配置。

说明书附图

图1

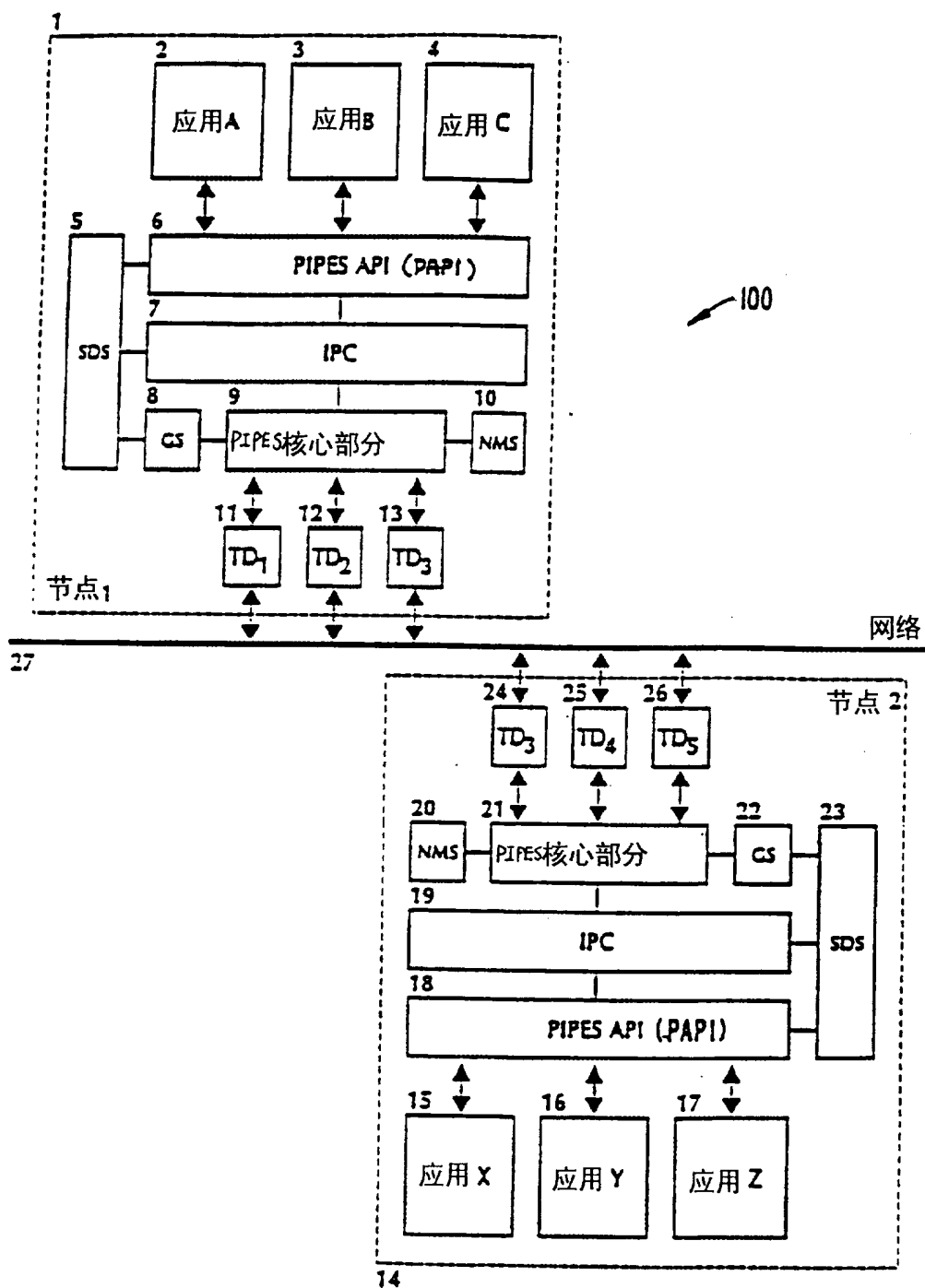


图2

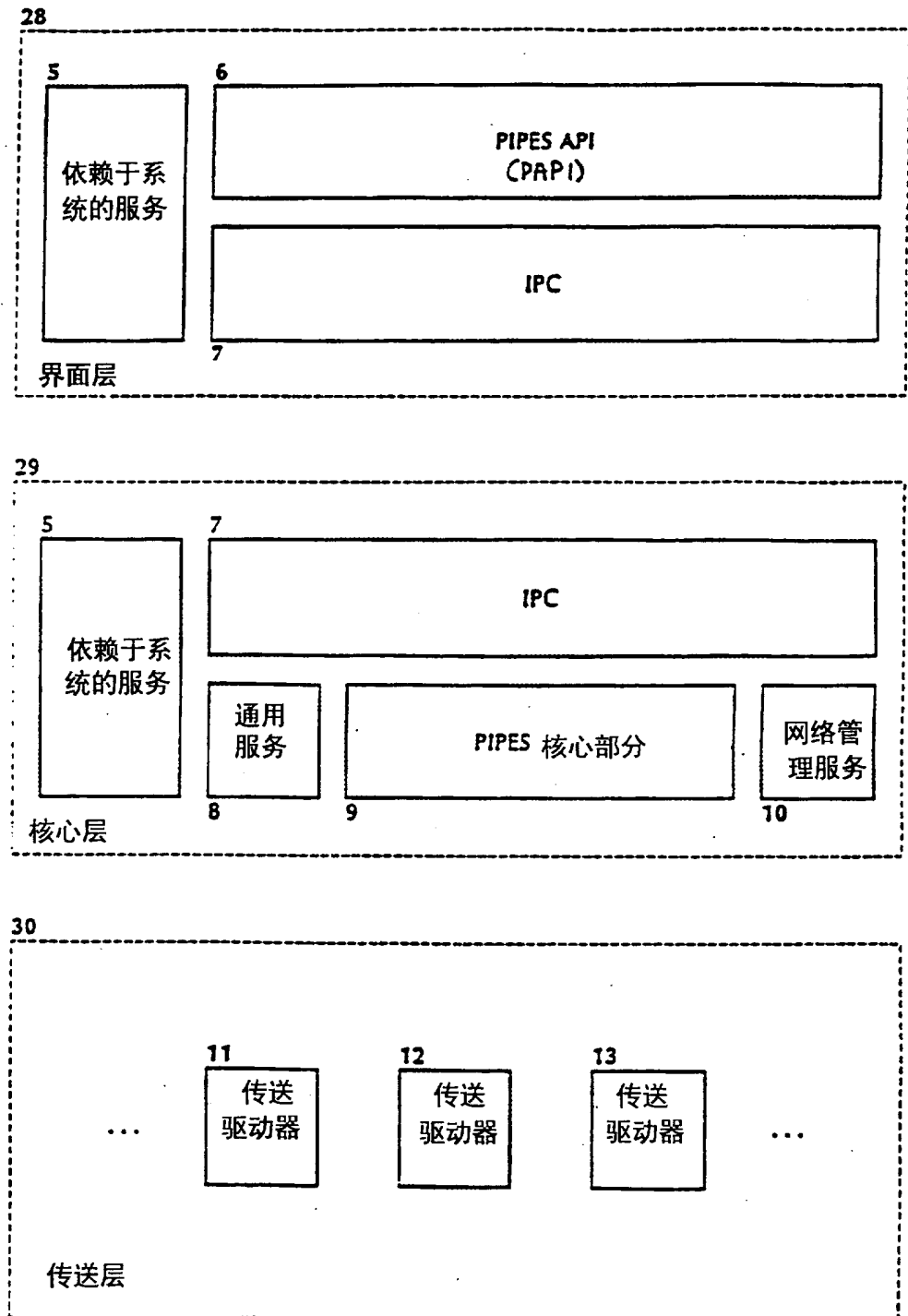


图3

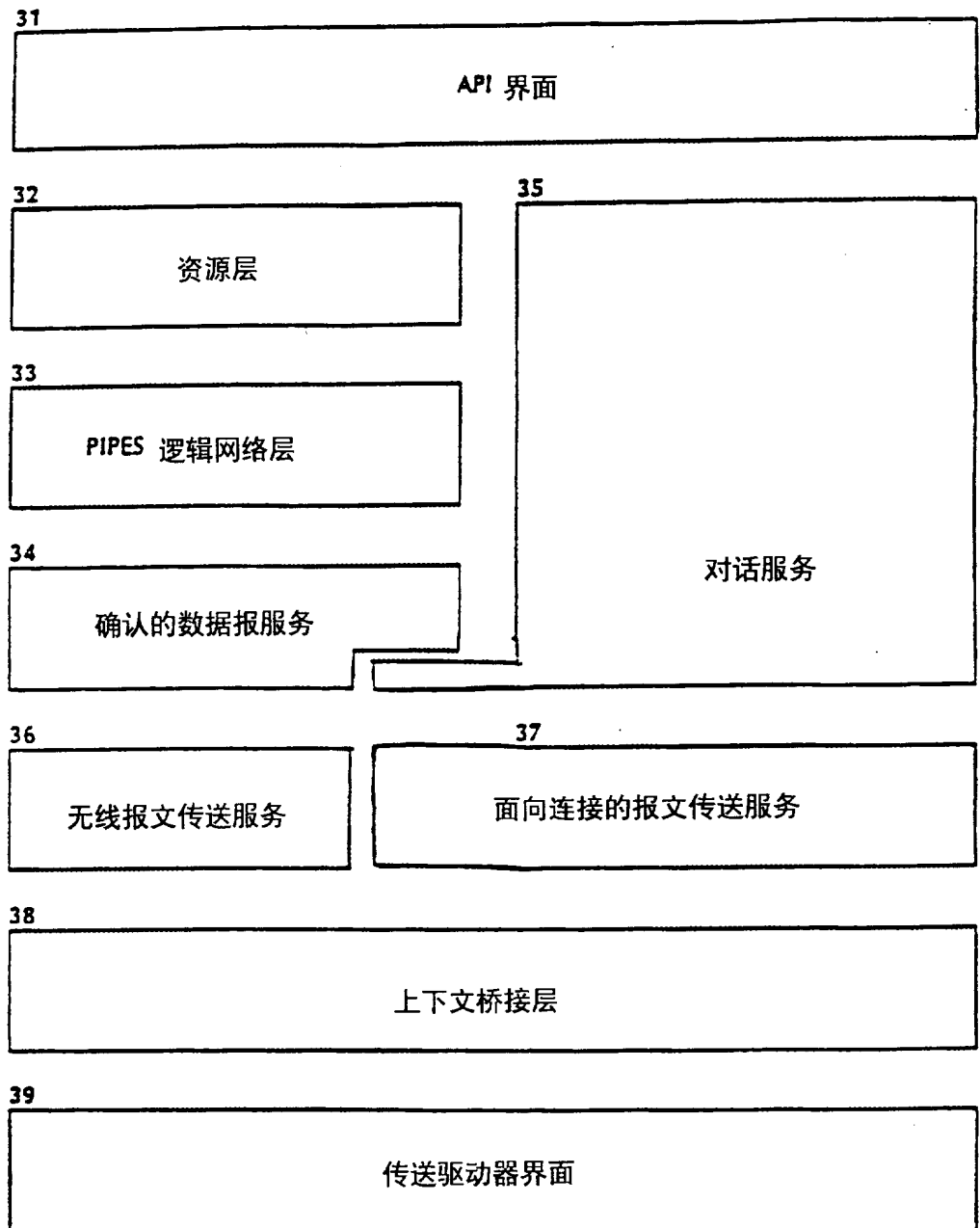


图4

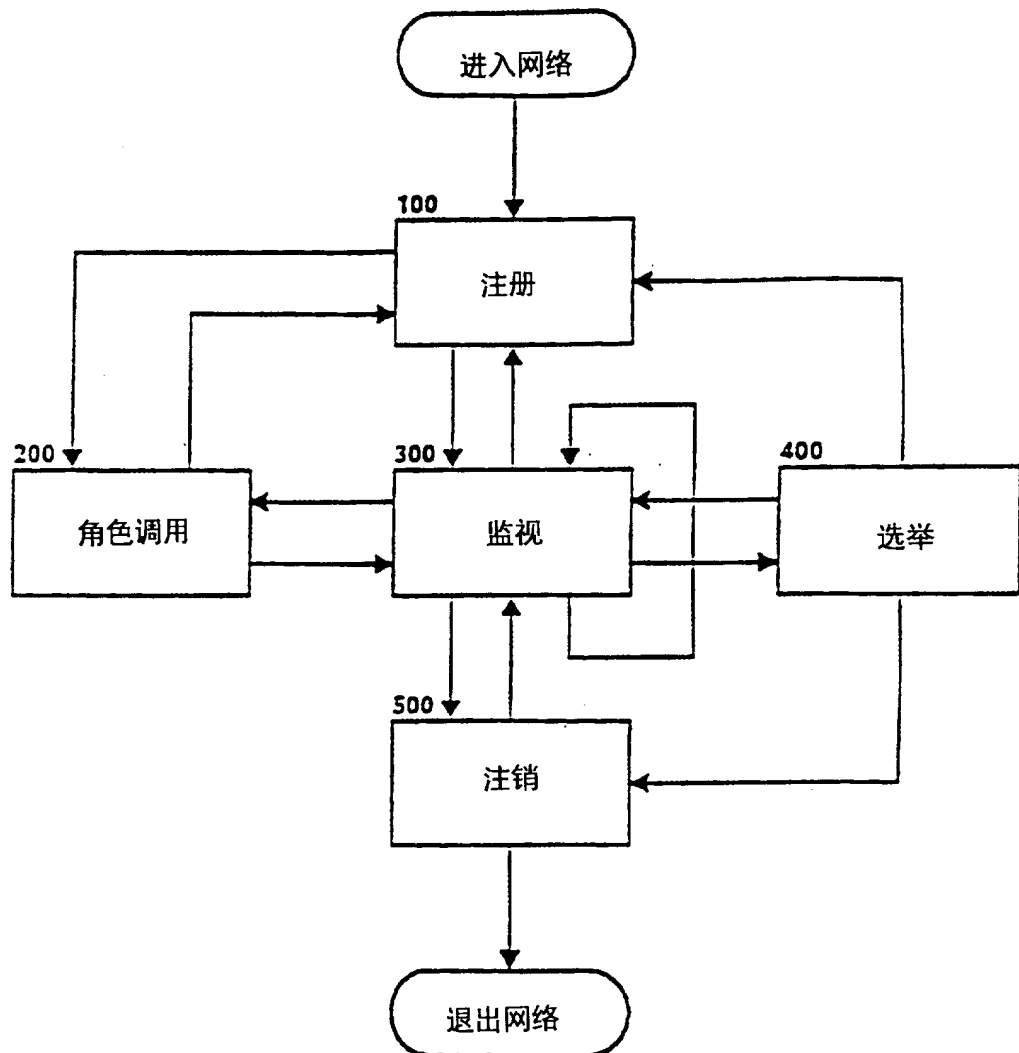


图5

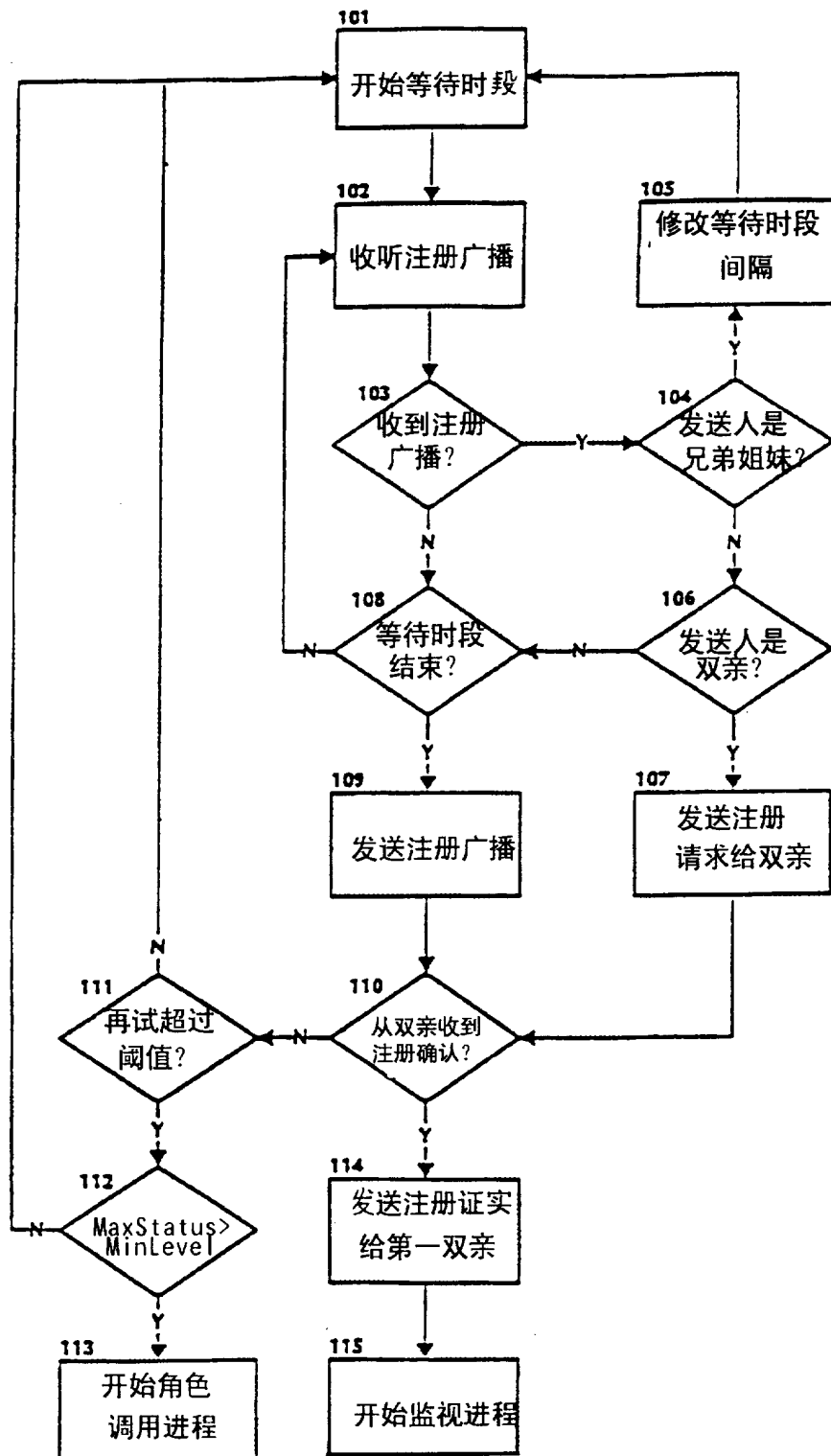


图6

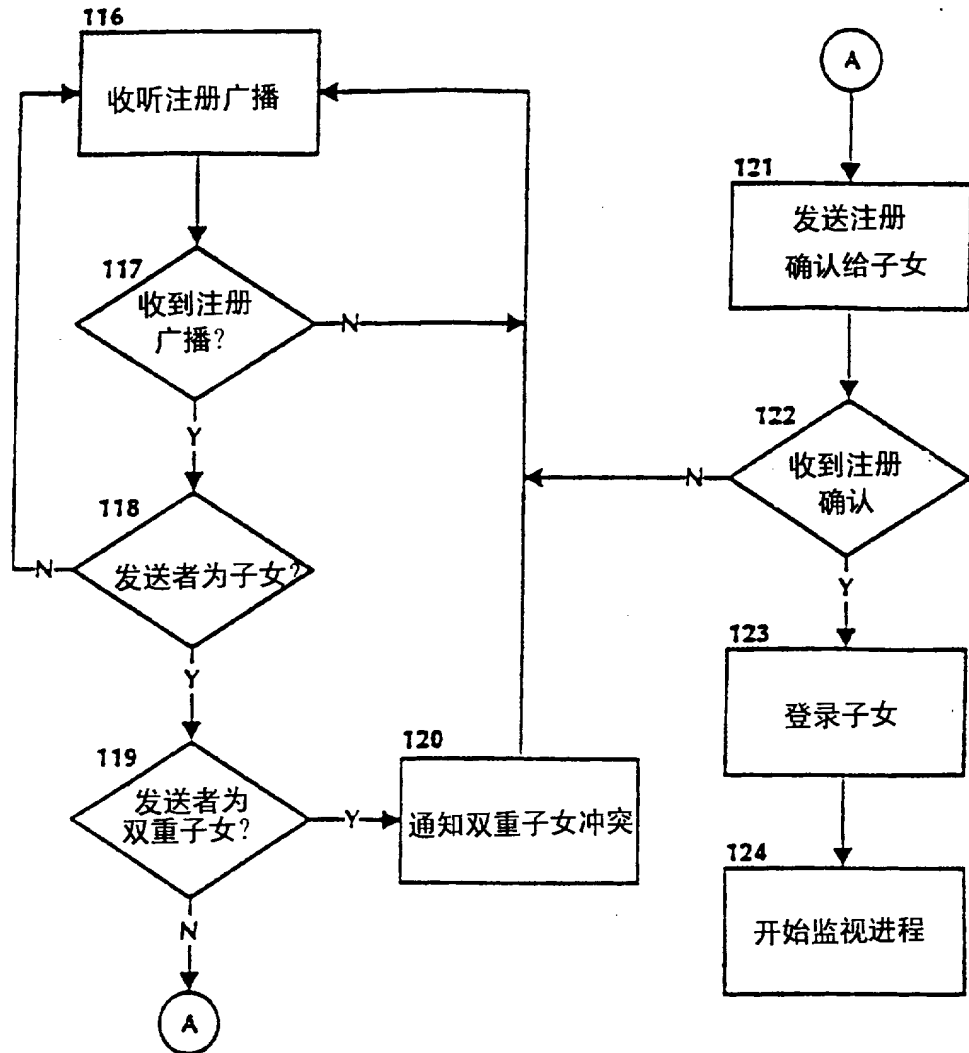


图7

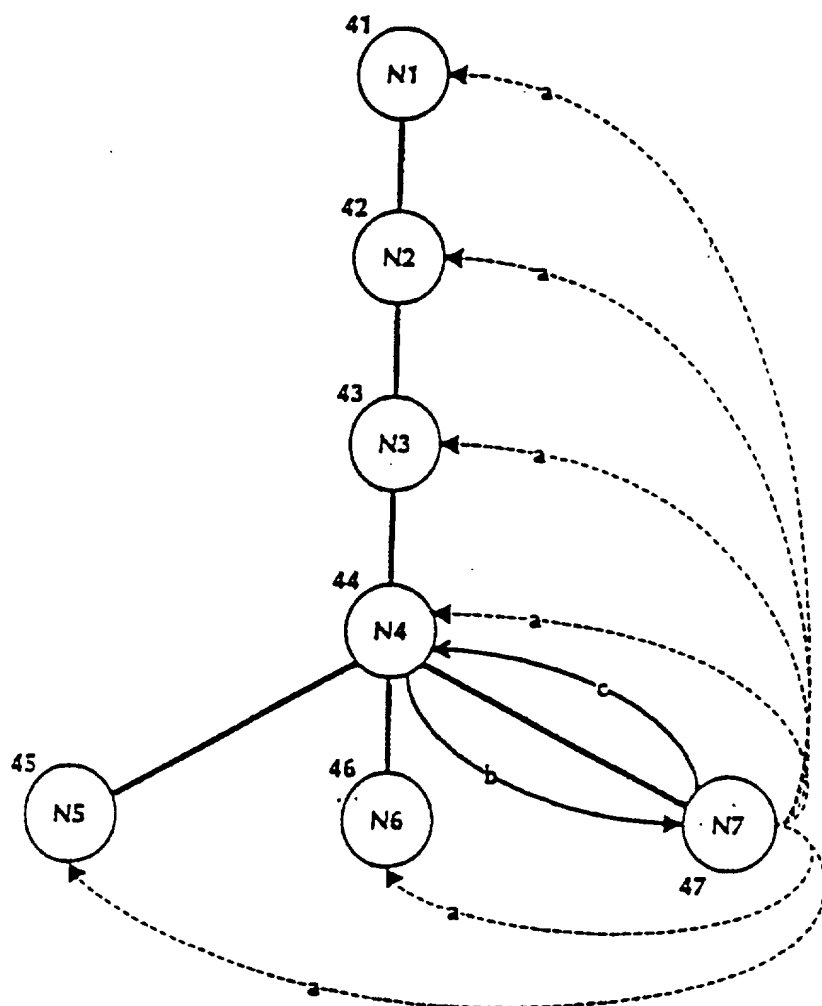


图8

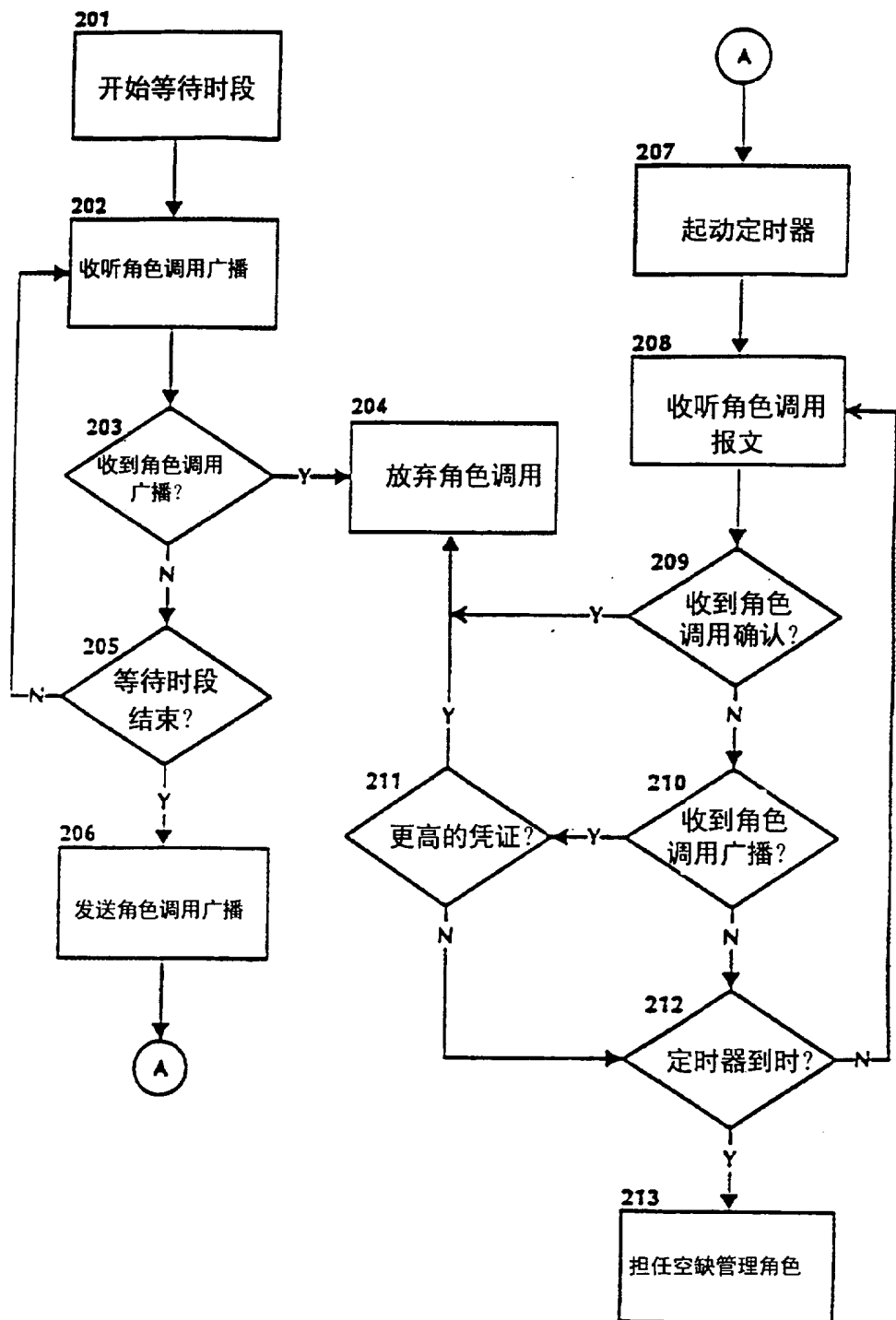


图 9

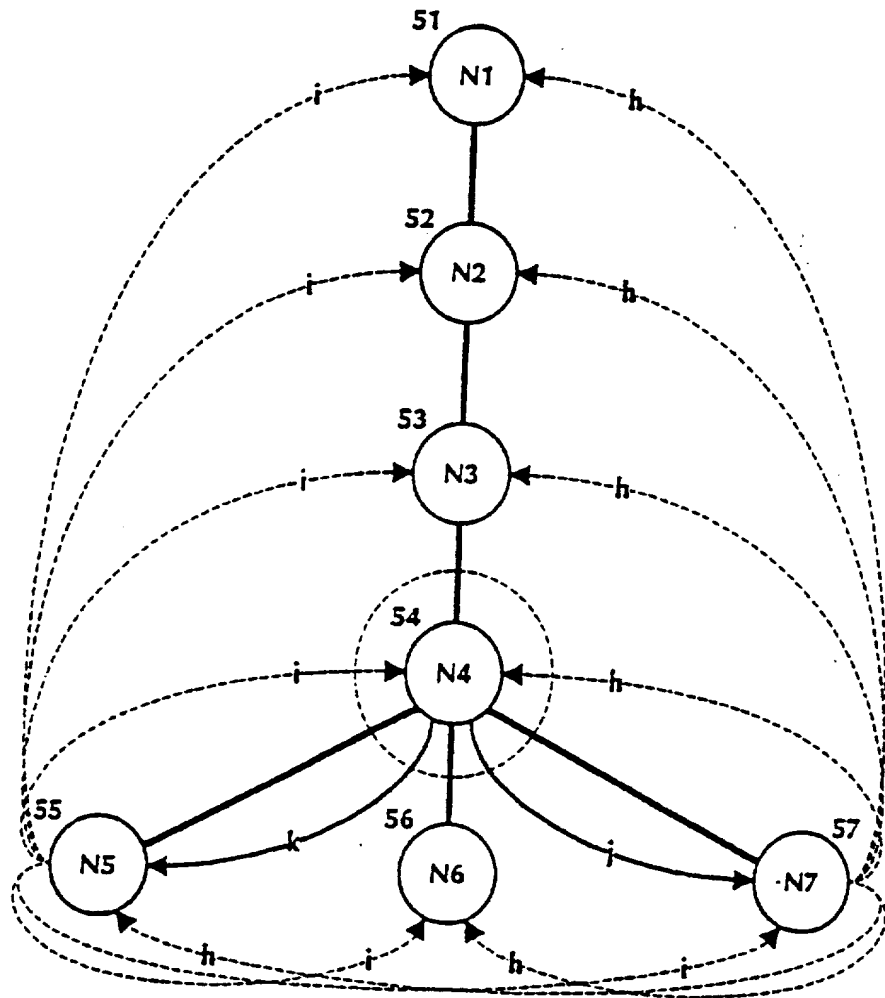


图 10

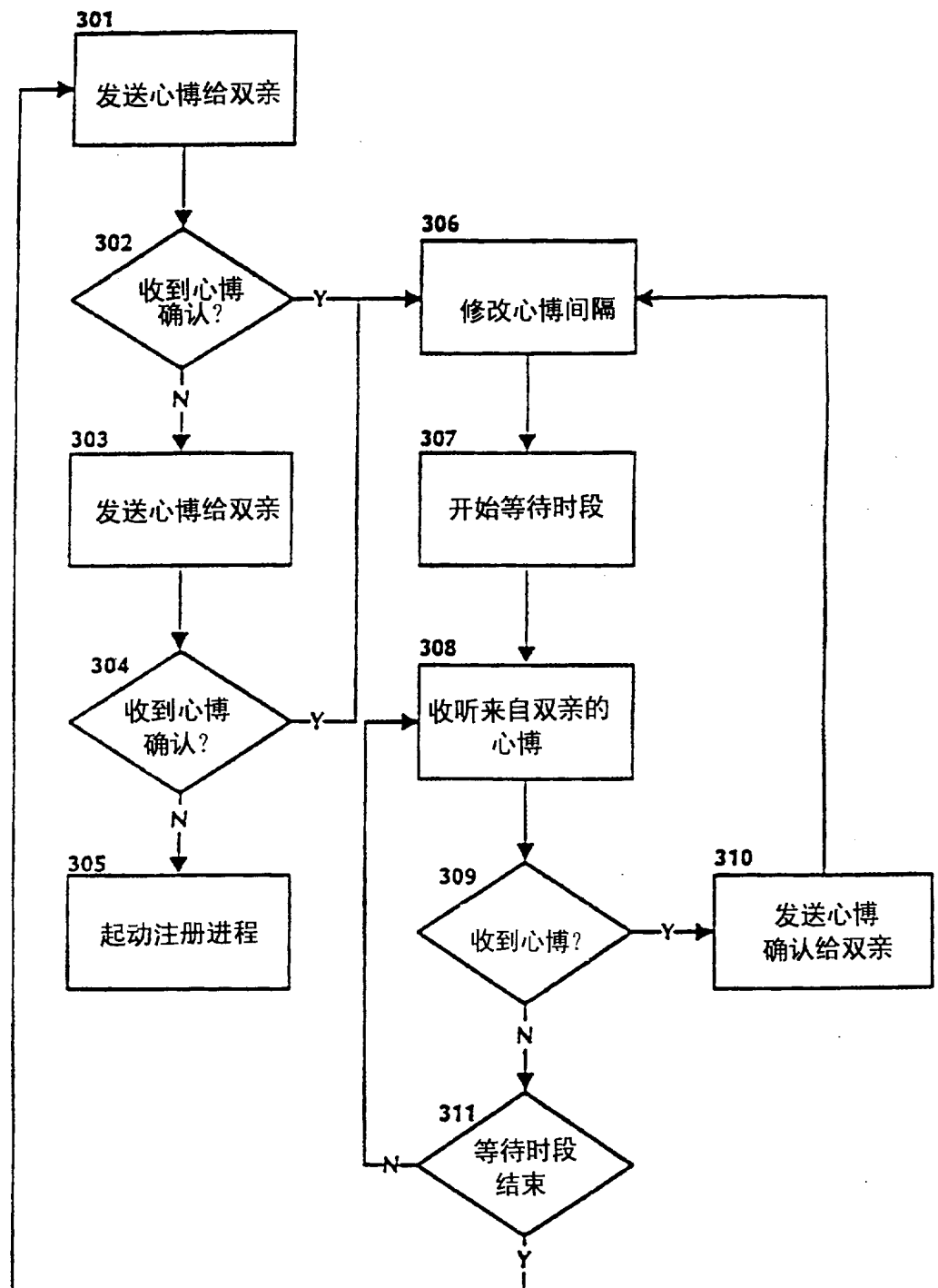


图 11

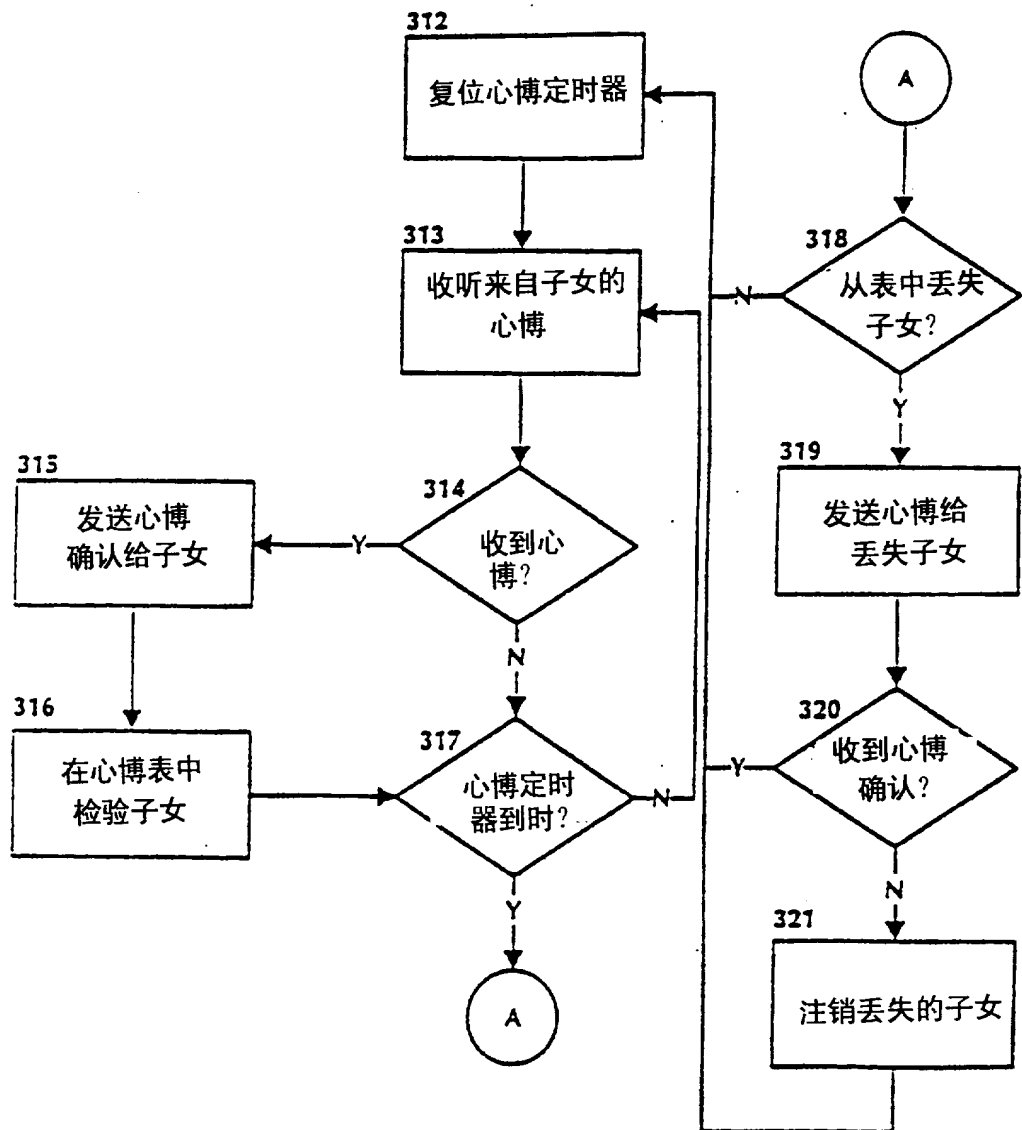


图12

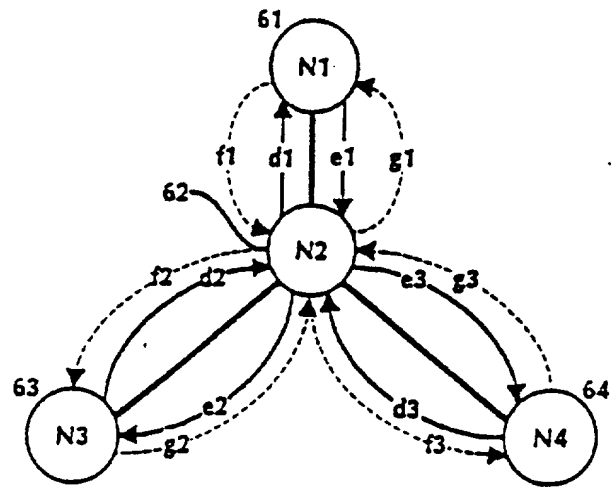


图 13

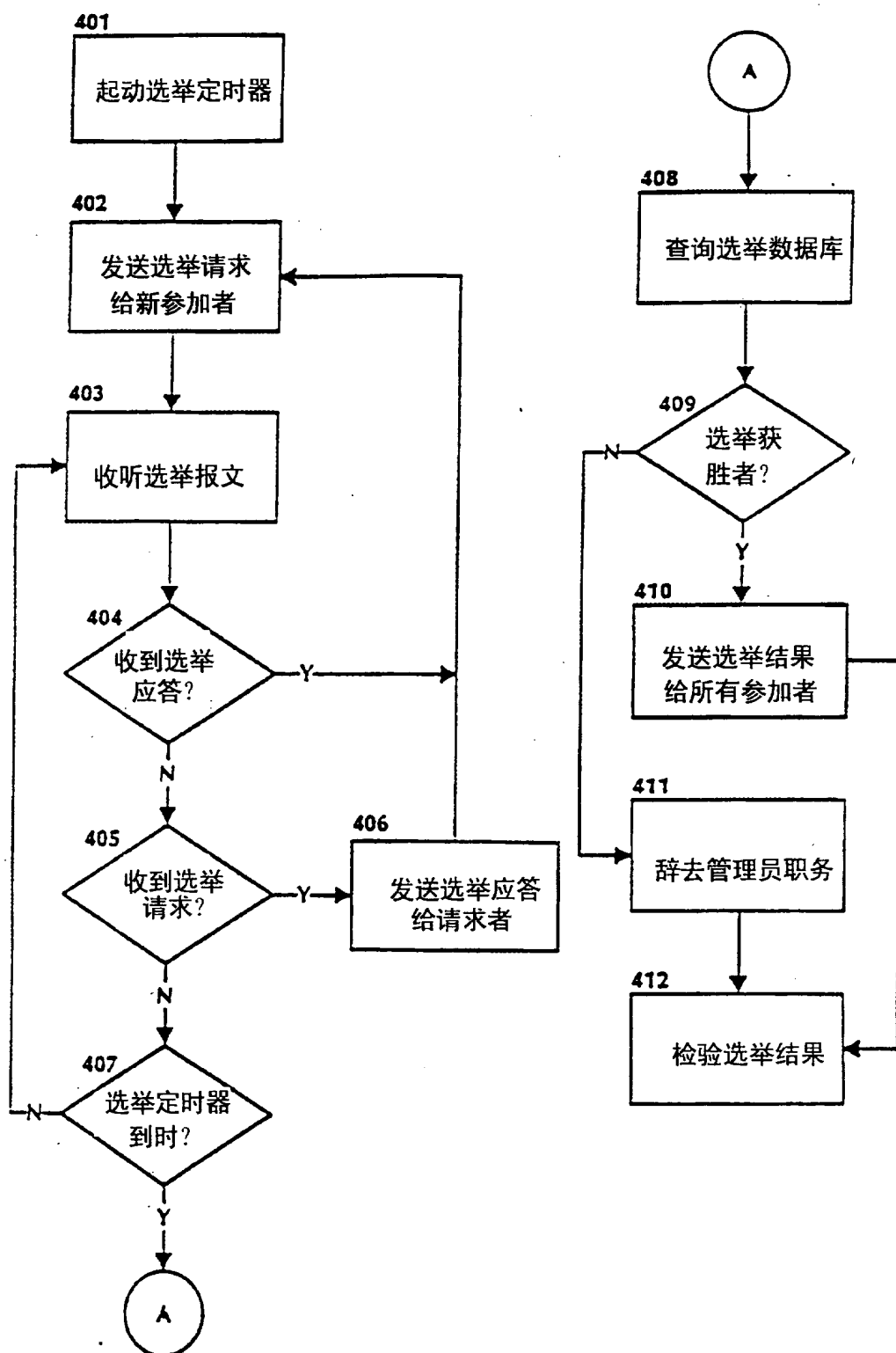


图14

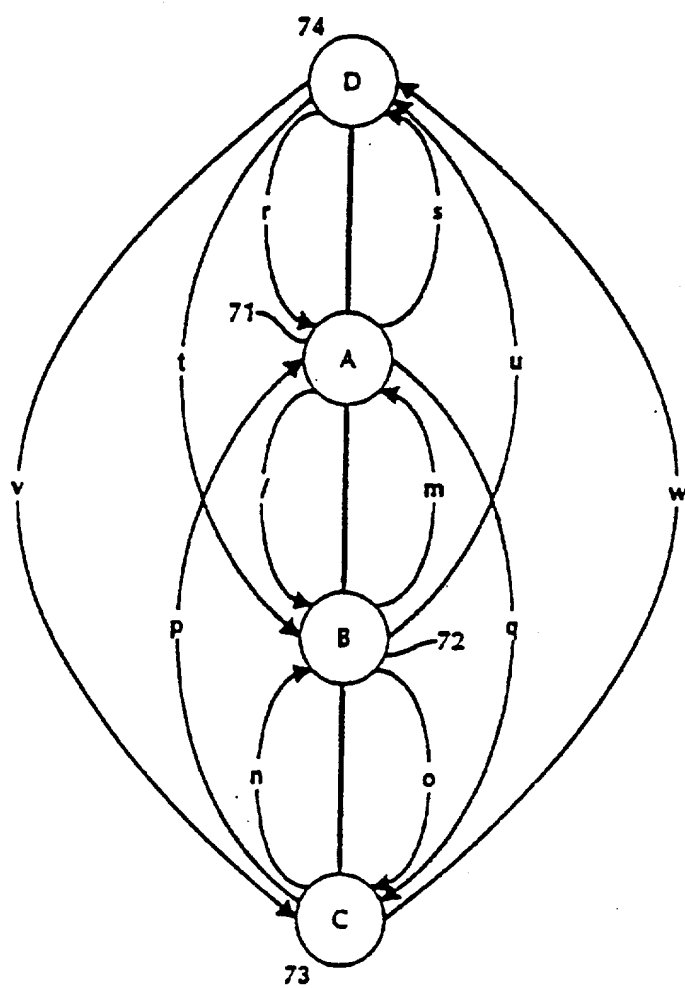


图15

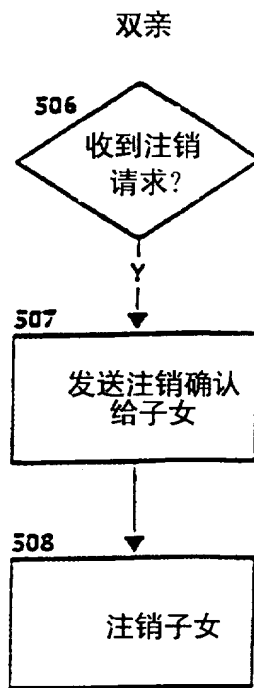
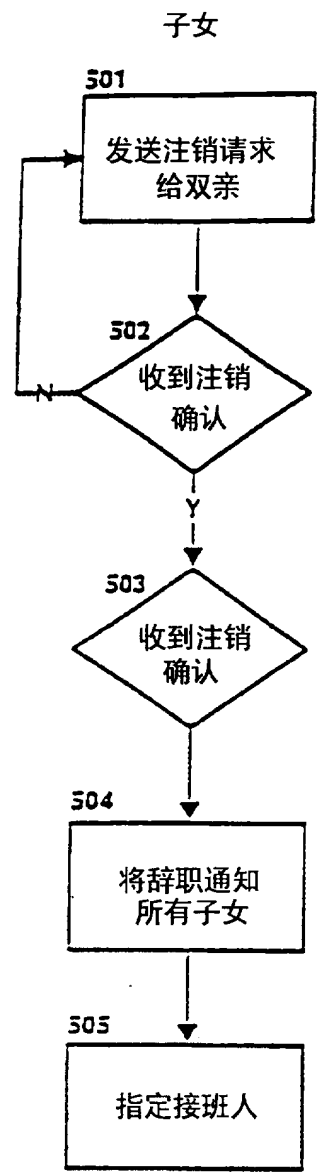


图16

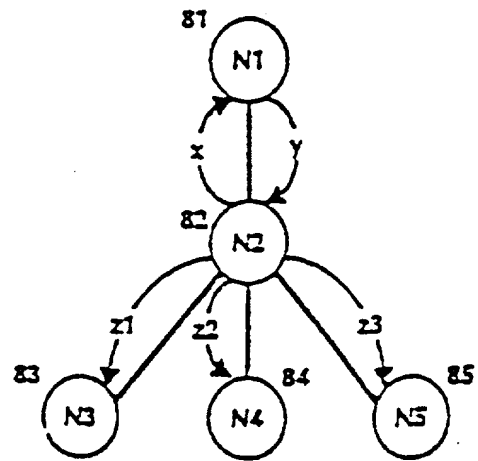


图17

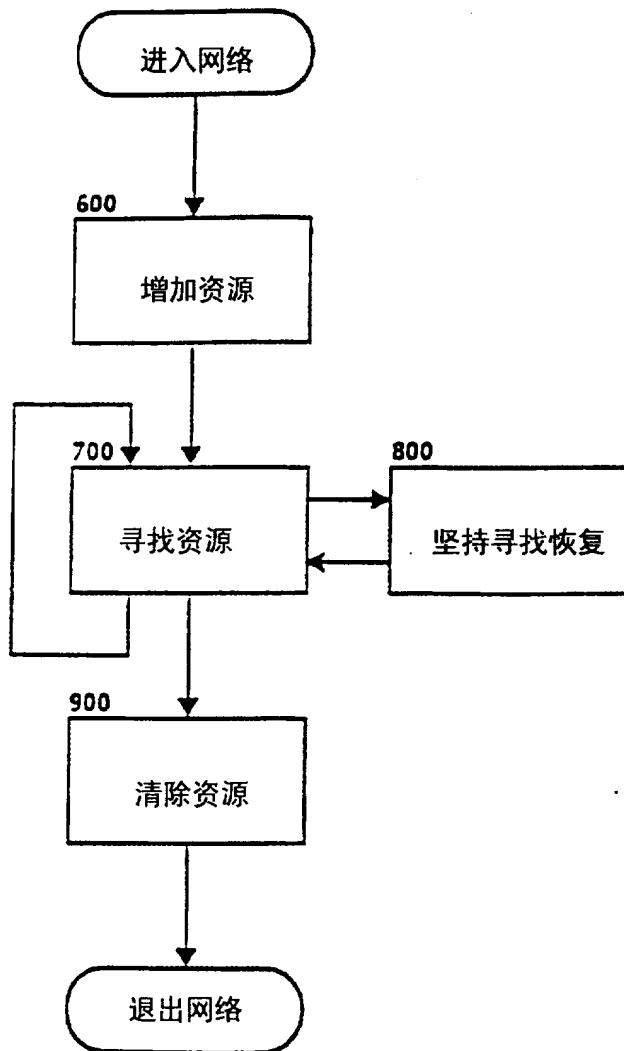


图18

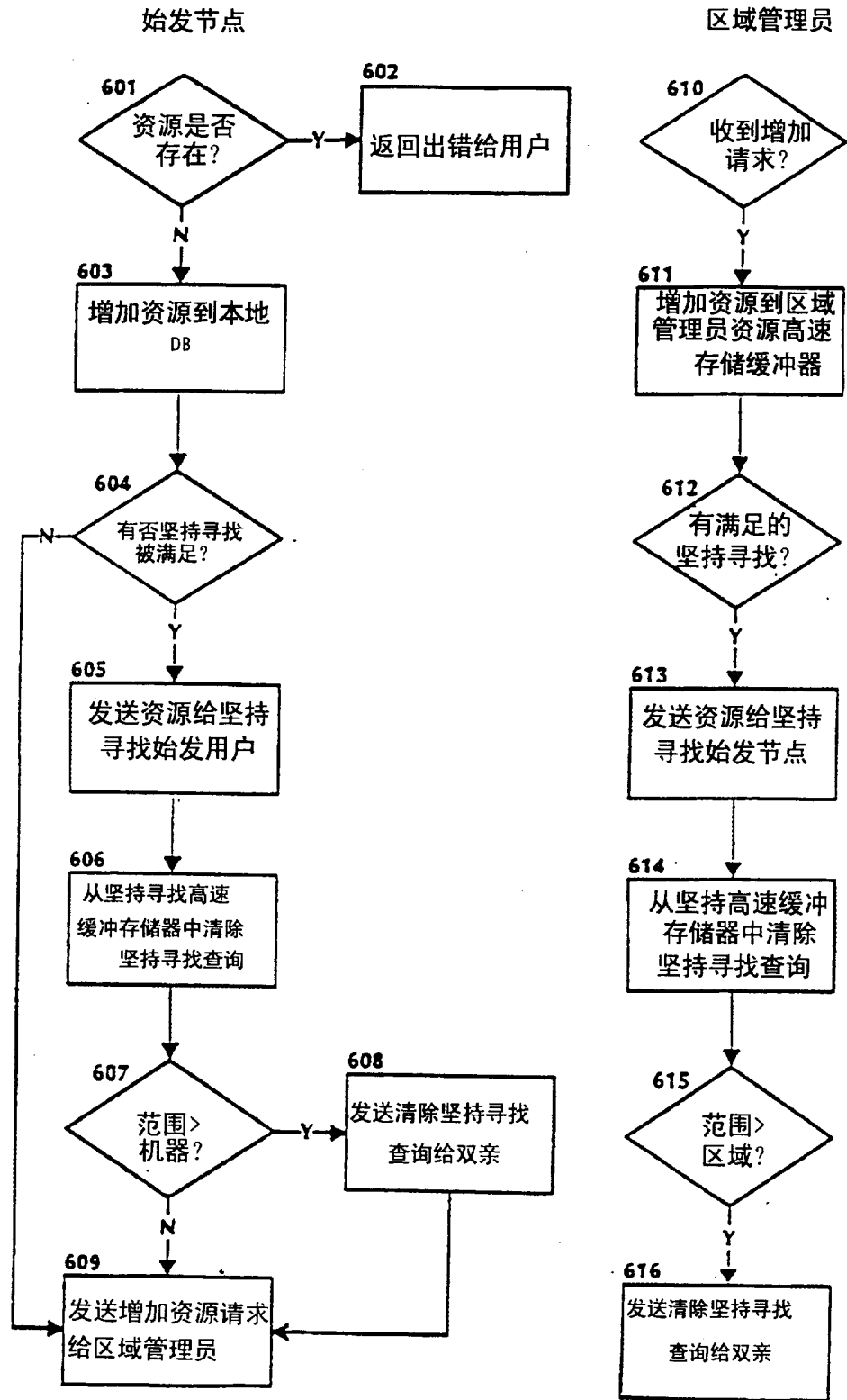


图 19

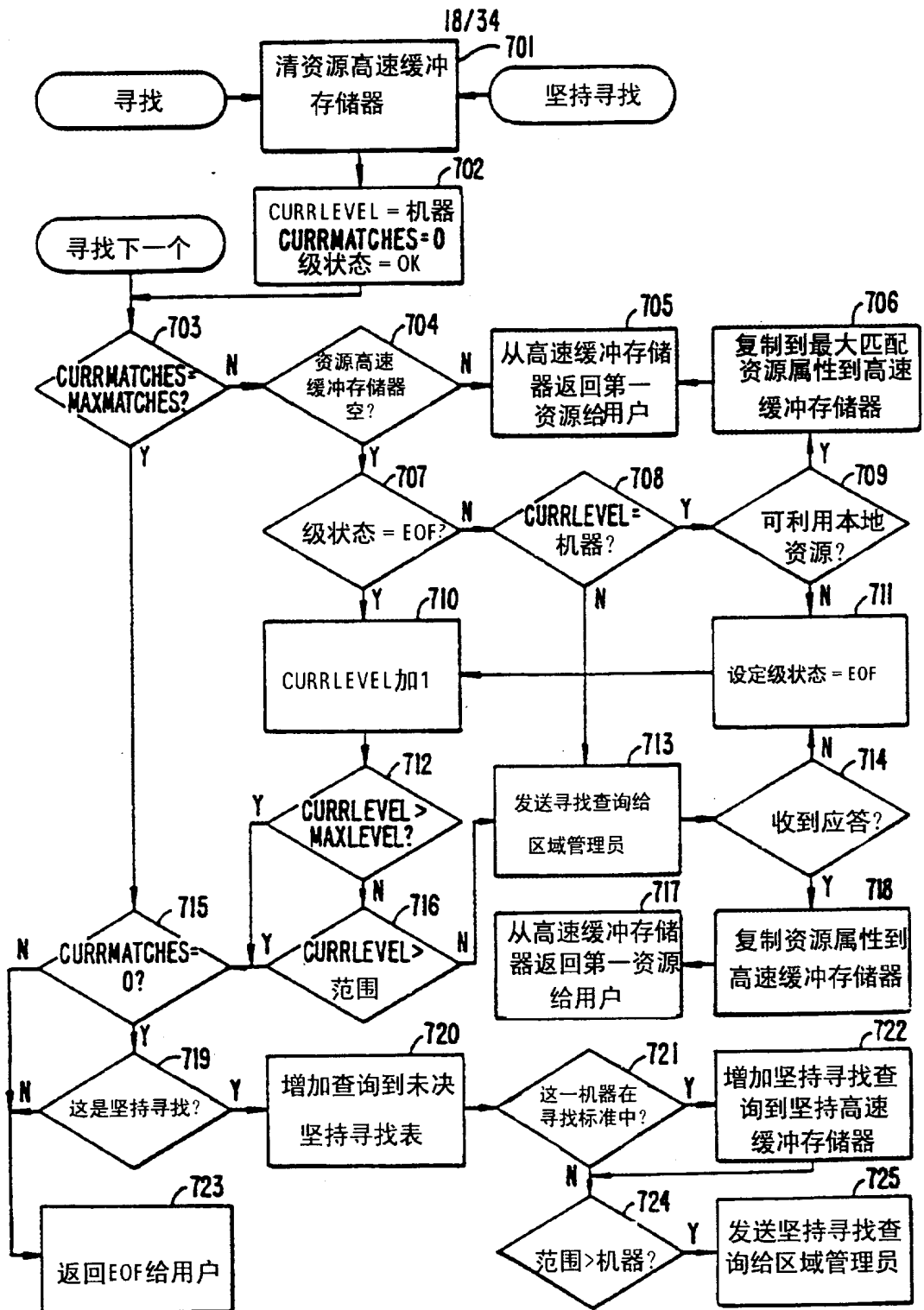


图20

在区域管理员节点上的寻找查询进程

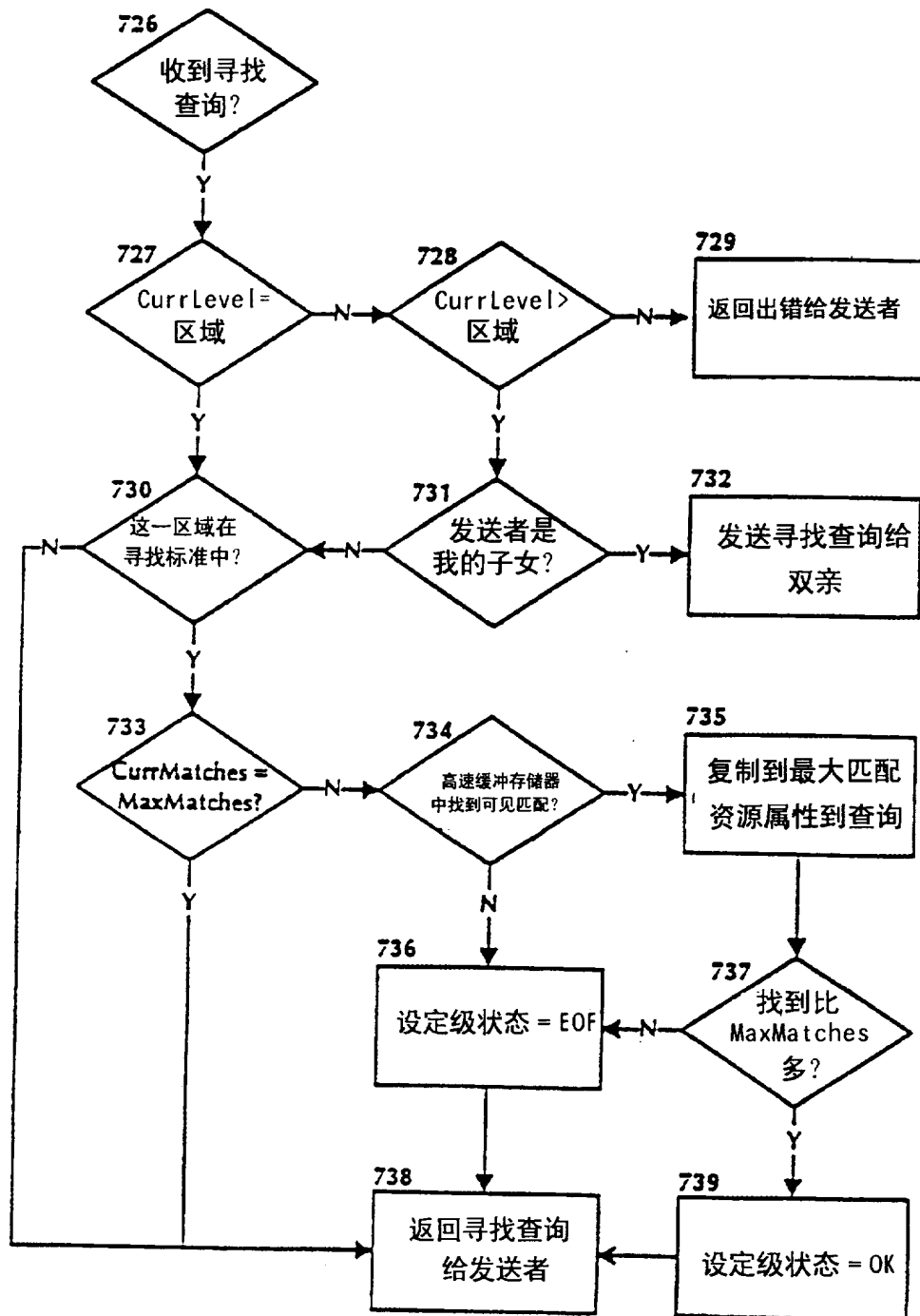


图21

在（管理员>区域管理员）节点上的寻找查询进程

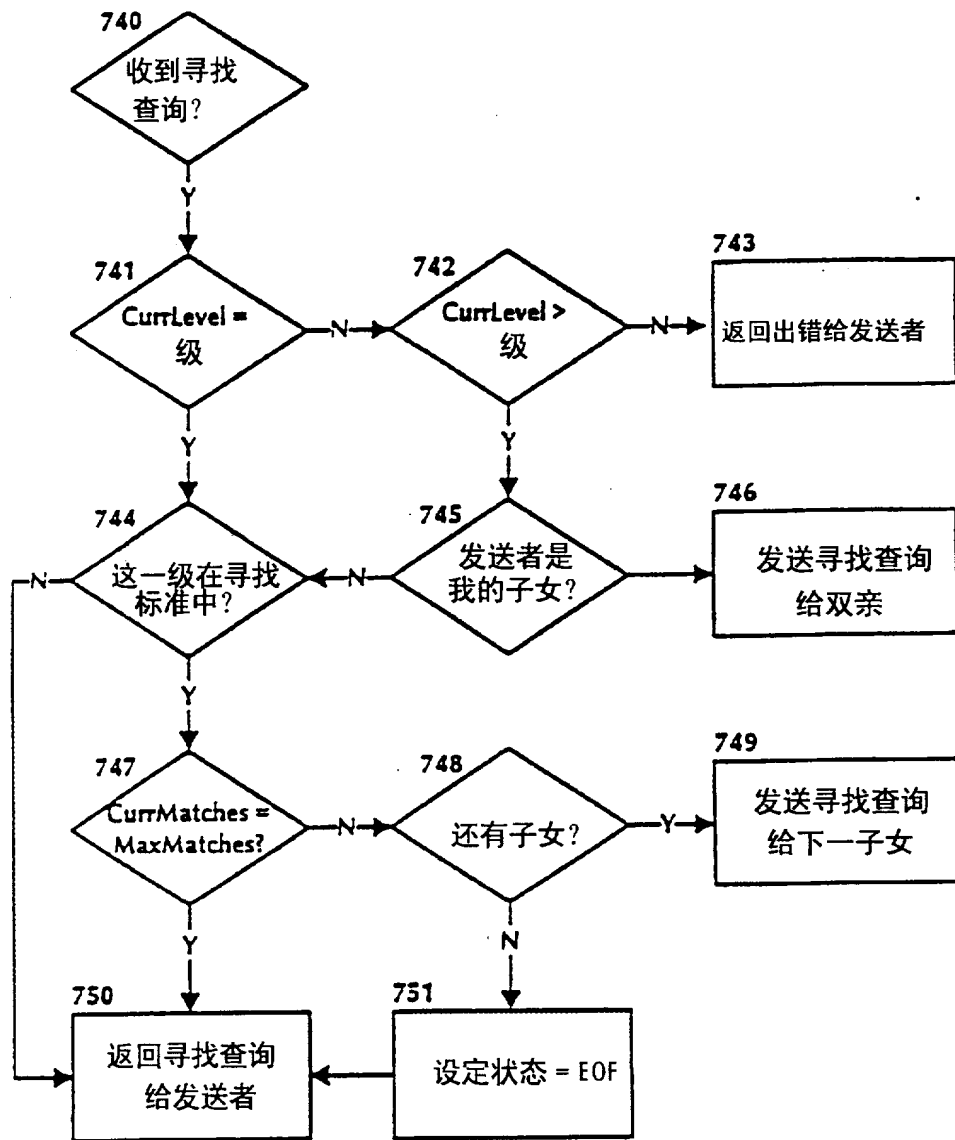


图22

在区域管理员节点上的坚持寻找查询进程

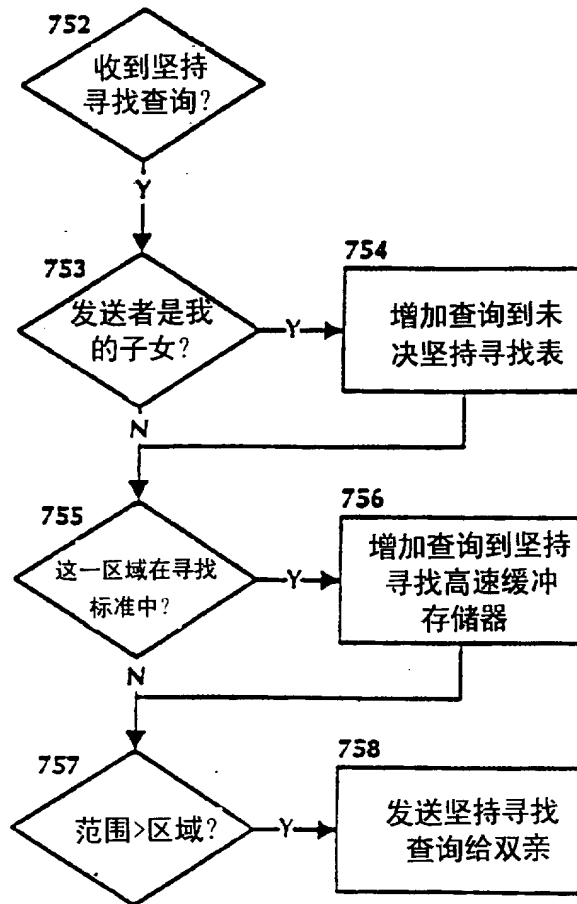


图23

在（管理员>区域管理员）节点上的坚持寻找查询进程

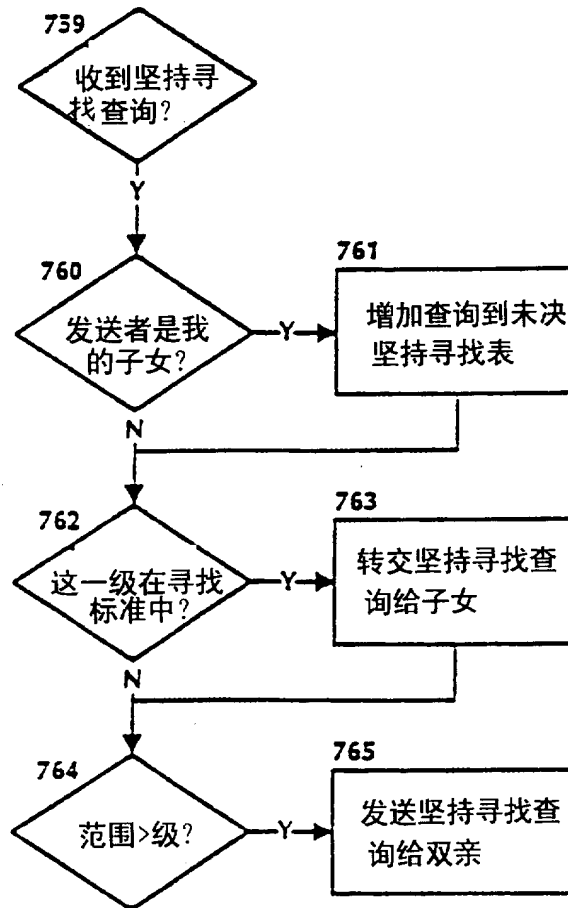


图24

在区域管理员节点上的清除坚持寻找查询进程

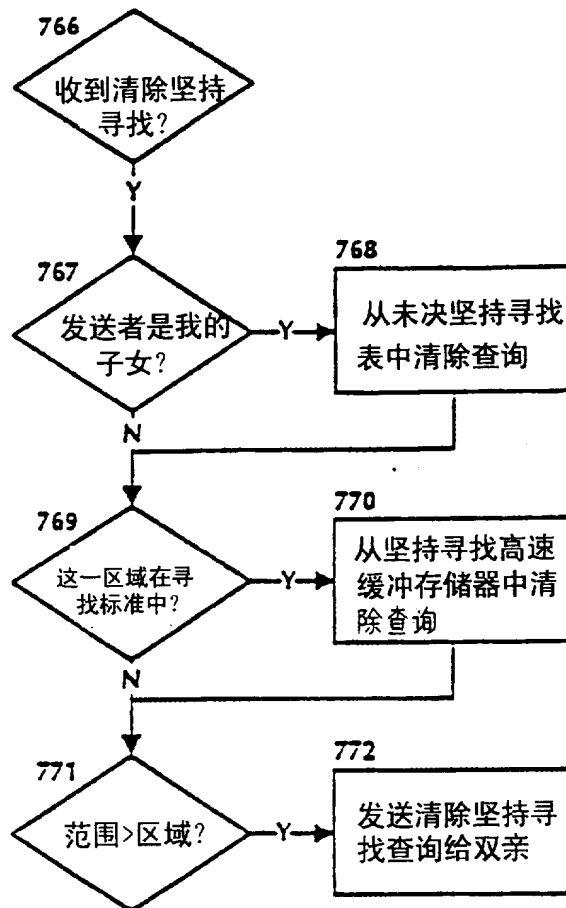


图25

在（管理员>区域管理员）节点上的清除坚持寻找查询进程

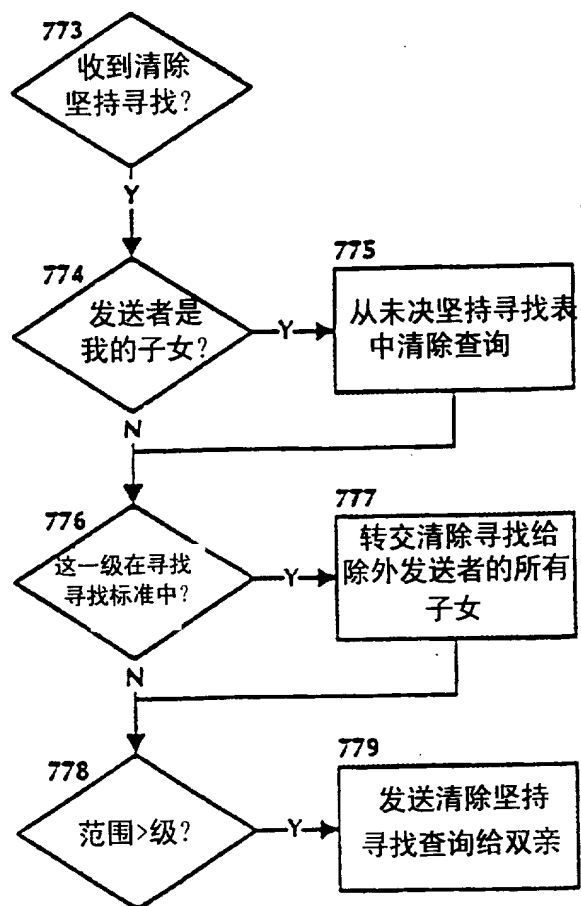


图26

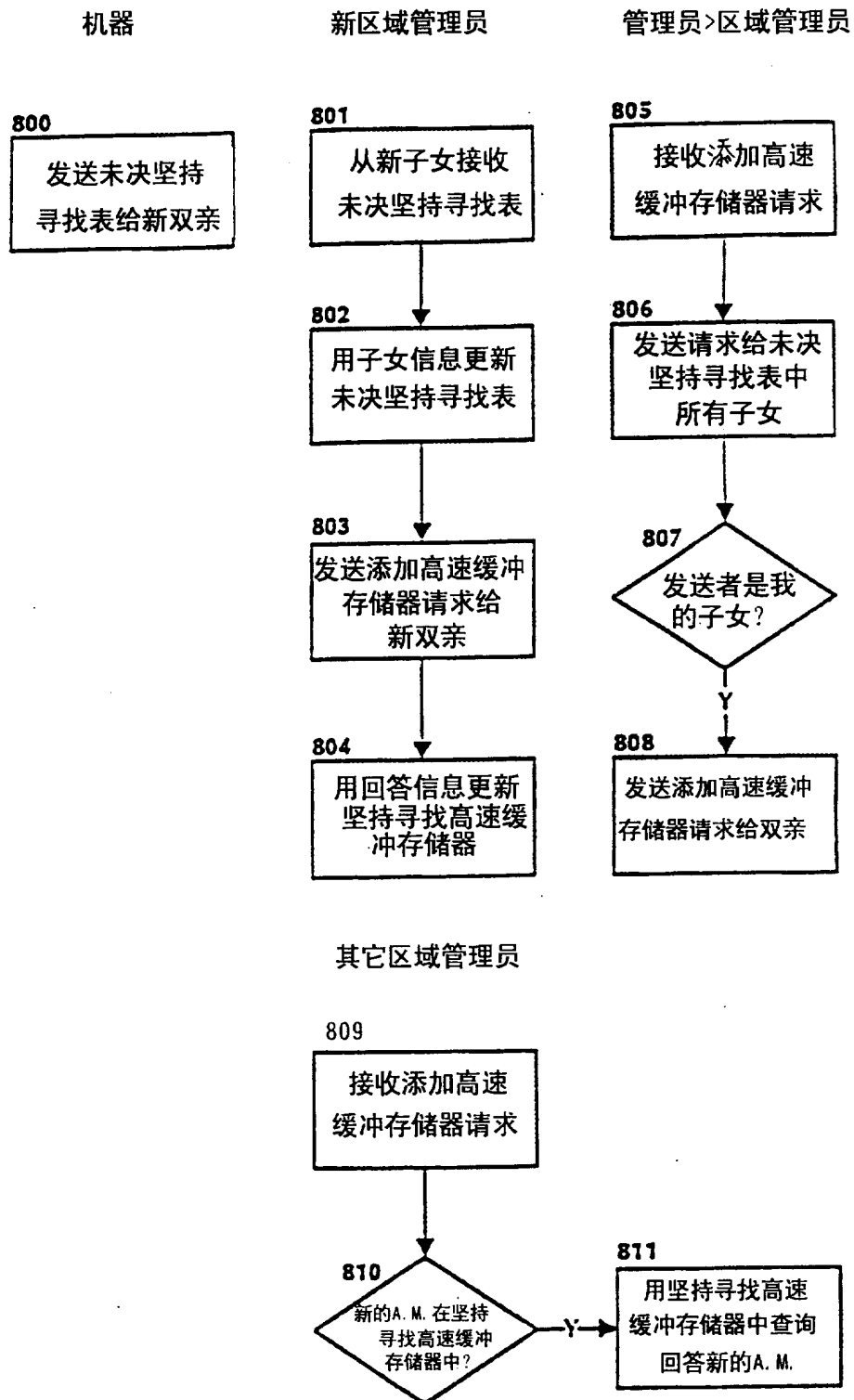


图27

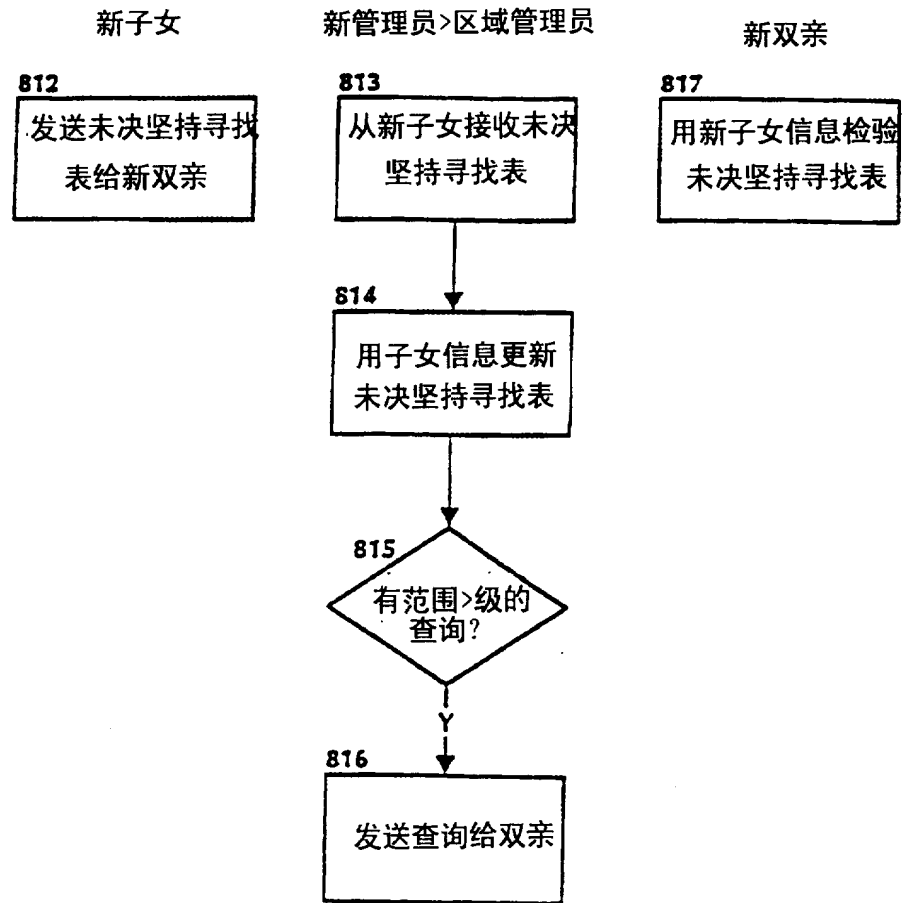


图28

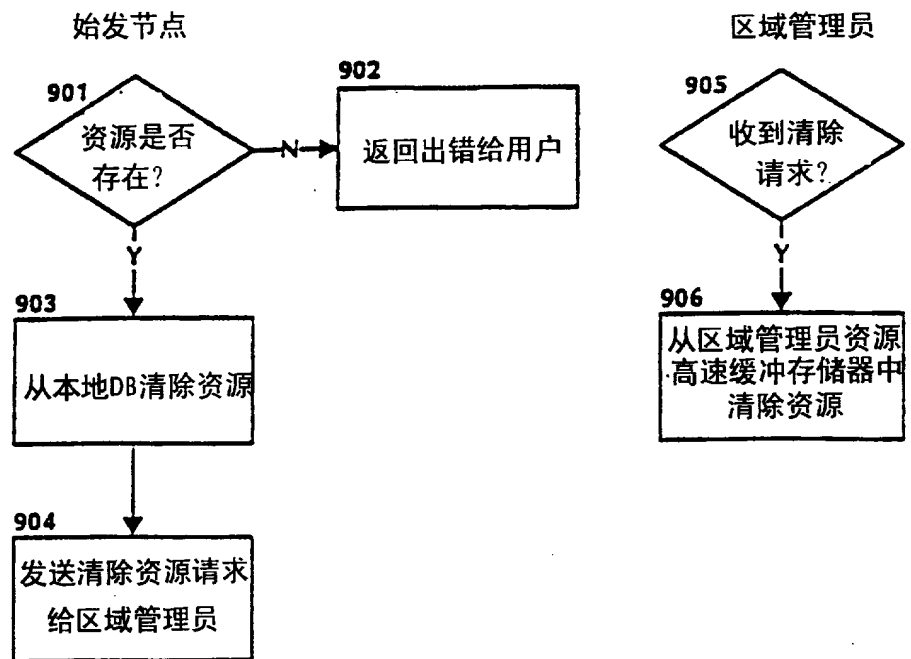
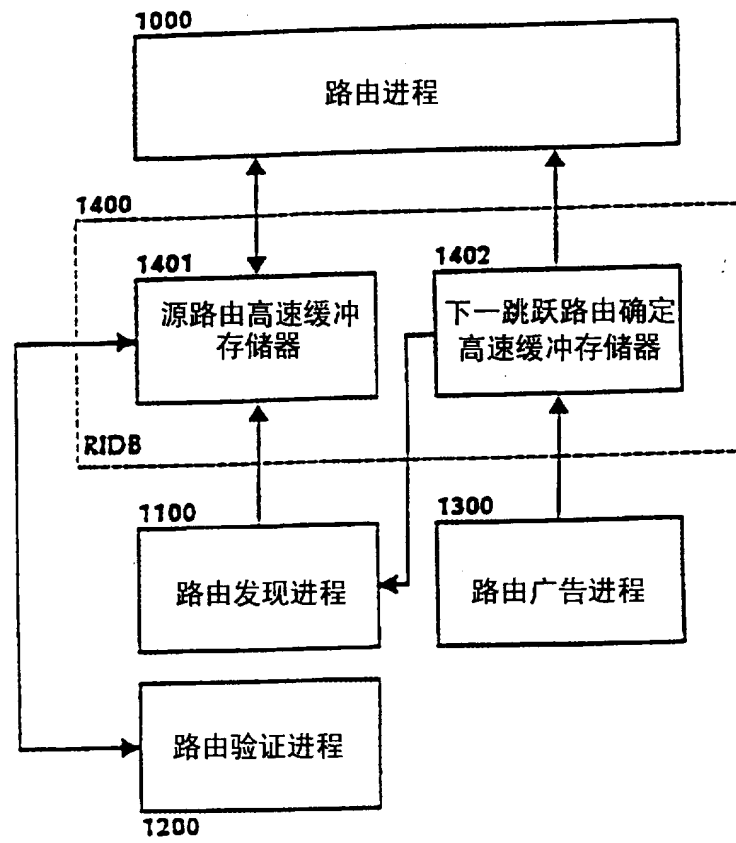


图 29A



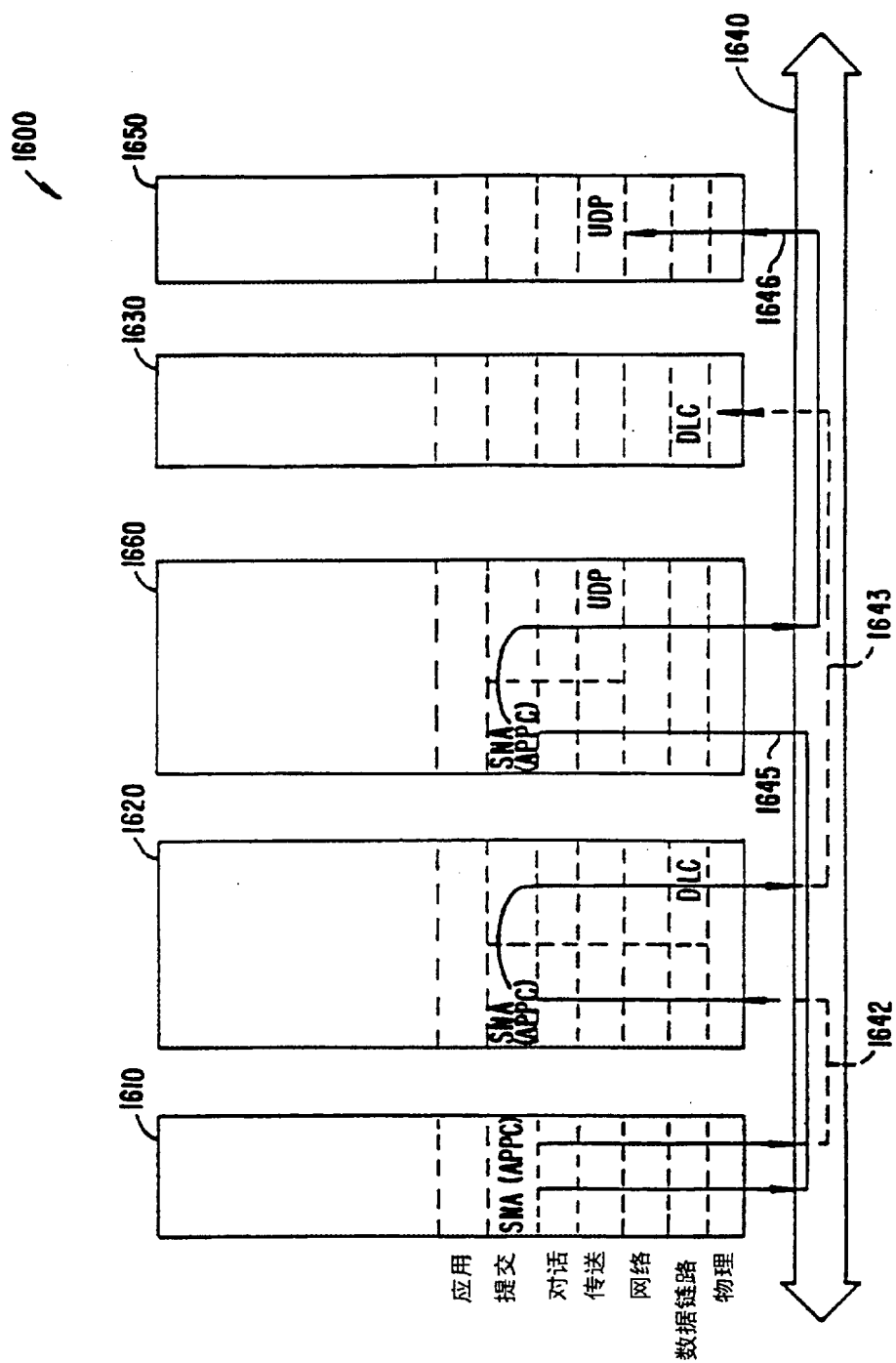


图 29B

图30

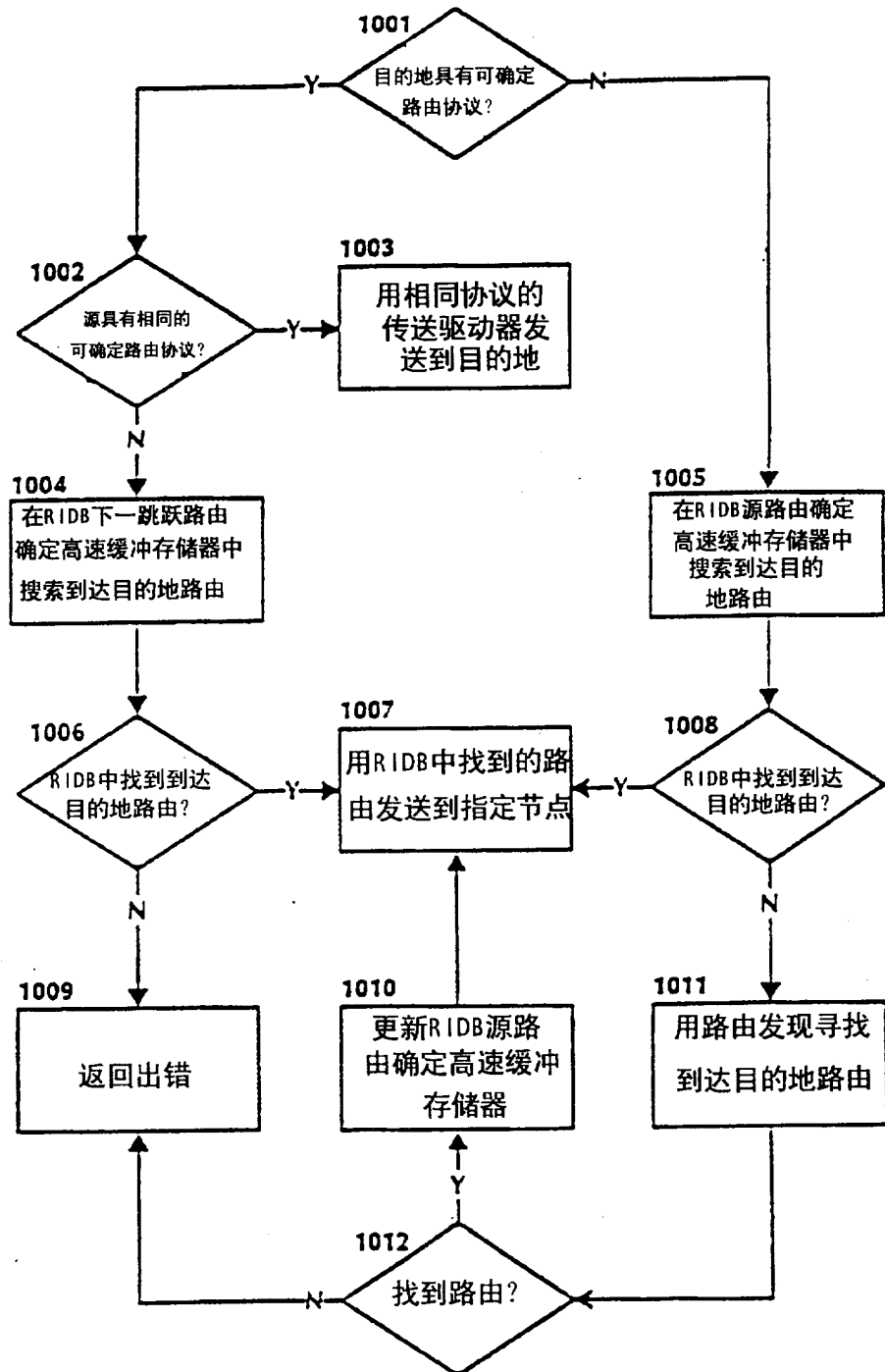


图31

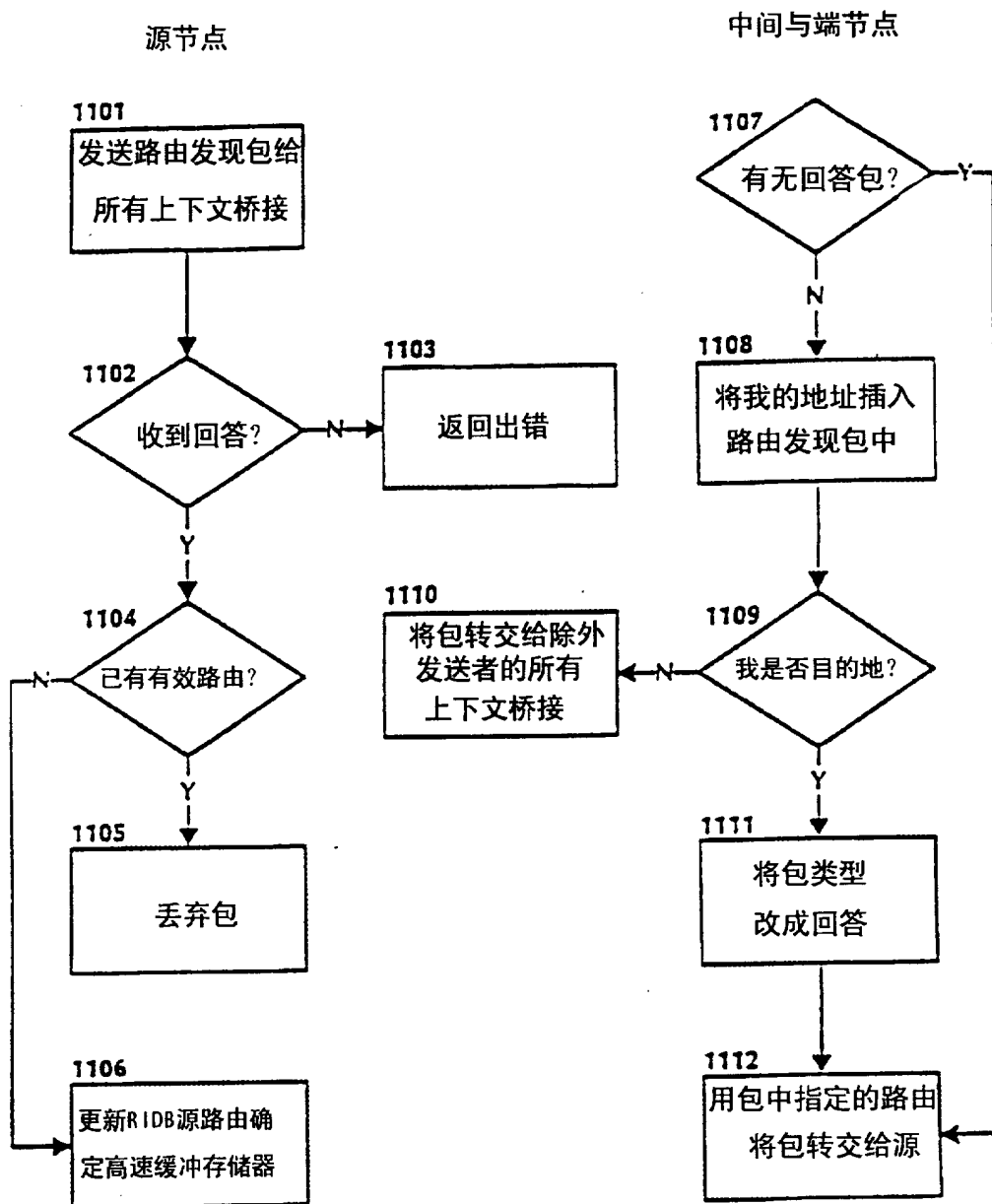


图32

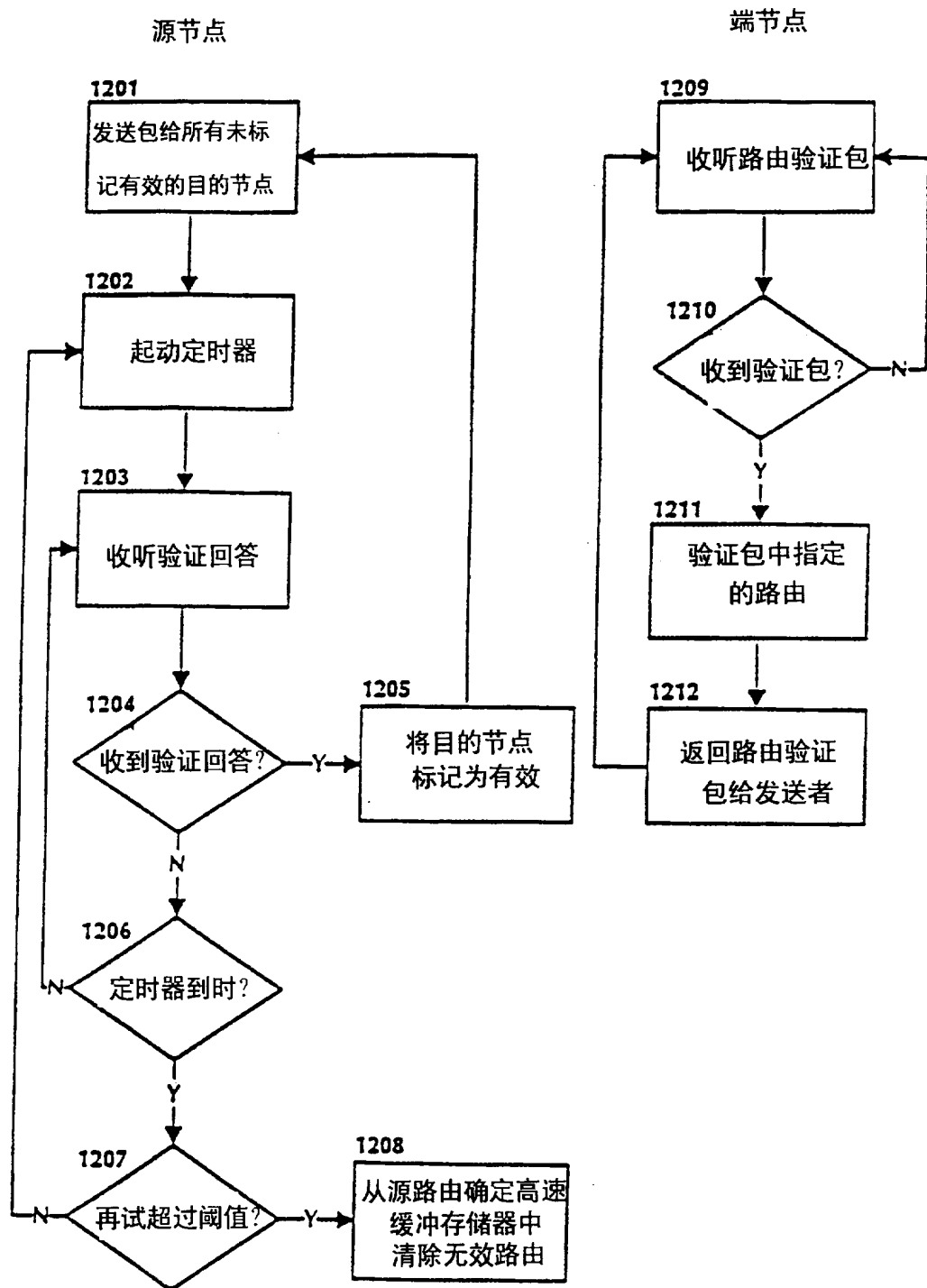


图33

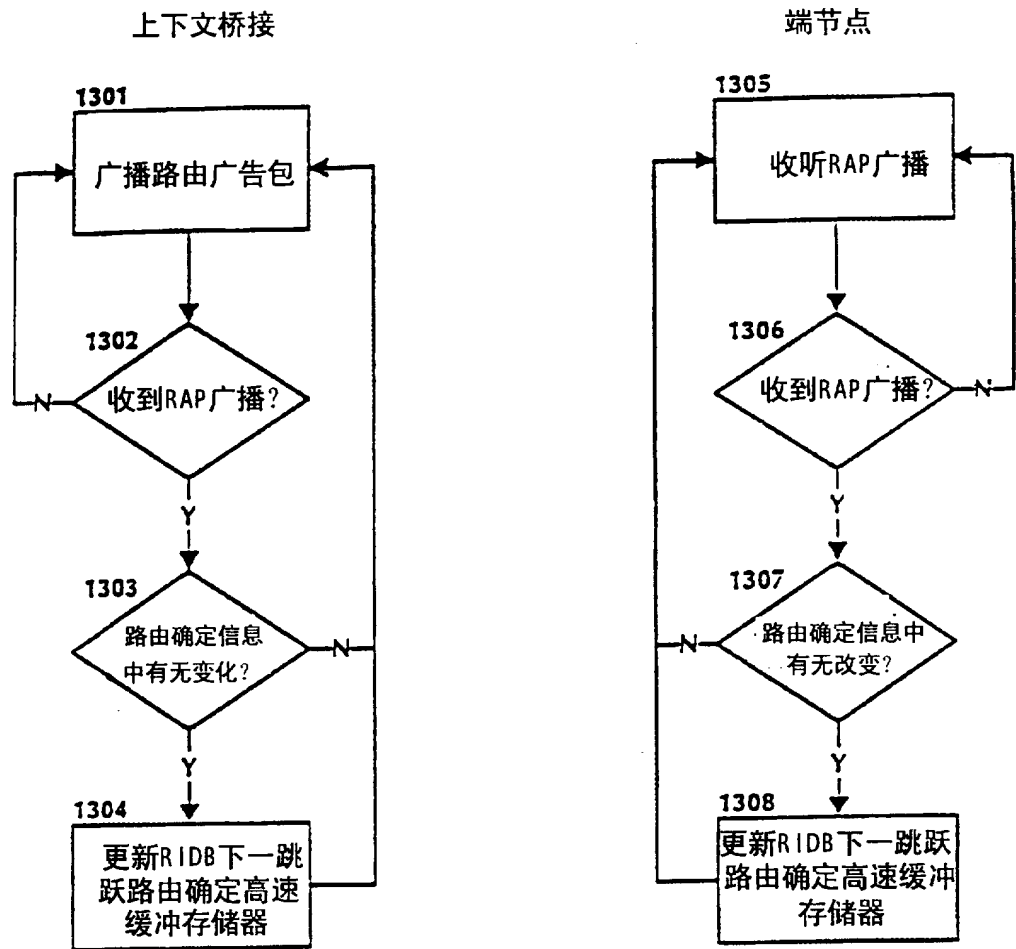


图34

