



(12) 发明专利申请

(10) 申请公布号 CN 115277346 A

(43) 申请公布日 2022. 11. 01

(21) 申请号 202211194673.6

H04L 67/04 (2022.01)

(22) 申请日 2022.09.28

H04L 12/40 (2006.01)

(71) 申请人 中国电子科技集团公司第十研究所
地址 610000 四川省成都市金牛区茶店子
东街48号

(72) 发明人 罗玲 张正勇 李鹏 卿宸
唐鸣跃 张伟

(74) 专利代理机构 成都九鼎天元知识产权代理
有限公司 51214
专利代理师 钱成岑

(51) Int. Cl.

H04L 41/0246 (2022.01)

H04L 41/044 (2022.01)

H04L 41/0823 (2022.01)

H04L 41/0895 (2022.01)

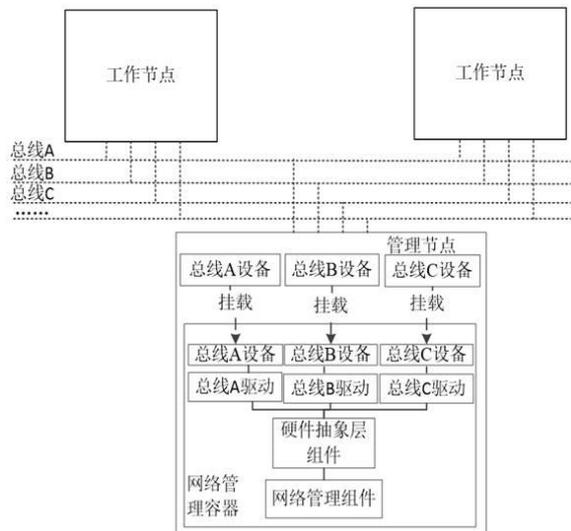
权利要求书3页 说明书7页 附图9页

(54) 发明名称

一种面向嵌入式异构总线的云原生容器网络系统

(57) 摘要

本发明公开了一种面向嵌入式异构总线的云原生容器网络的云原生容器网络系统,包括:挂载异构总线驱动并实现异构总线统一封装的硬件抽象层组件;对接应用数据并实现应用句柄及回调注册和应用数据收发的硬件抽象层接口组件;以及实现所有网络配置下发和网络信息查询的网络管理组件。本发明采用硬件抽象层组件实现对异构总线数据发送和接收的统一抽象,通过设备挂载提供对异构总线的接入支撑机制,支持容器内应用以透明方式实现通过异构总线完成数据发送和接收,解决容器化应用在嵌入式领域难以移植的问题。



1. 一种面向嵌入式异构总线的云原生容器网络系统,其特征在于,包括:挂载异构总线驱动并实现异构总线统一封装的硬件抽象层组件;对接应用数据并实现应用句柄及回调注册和应用数据收发的硬件抽象层接口组件;以及实现所有网络配置下发和网络信息查询的网络管理组件;

在嵌入式异构总线环境下,所述硬件抽象层组件部署在工作节点的硬件抽象层守护进程容器以及管理节点中,所述网络管理组件部署在管理节点中,所述硬件抽象层接口组件部署在工作节点的应用容器中;

硬件抽象层守护进程容器和应用容器分别将共享数据卷/var/run挂载进各自运行的容器中,且硬件抽象层守护进程容器将所有需要接入容器网络的异构总线设备挂载进自身运行的容器中;硬件抽象层组件在共享数据卷/var/run目录中创建Unix域套接字hal.sock;网络管理组件根据嵌入式系统建模信息完成网络配置信息到各个节点硬件抽象层组件的下发;应用容器通过该hal.sock向硬件抽象层组件进行路由表项注册和应用数据收发;硬件抽象层组件通过路由信息和注册回调信息完成容器间和节点间的数据路由和数据收发。

2. 根据权利要求1所述的面向嵌入式异构总线的云原生容器网络系统,其特征在于,所述硬件抽象层组件包括守护进程容器数据接收模块、守护进程容器数据发送模块、硬件抽象层数据发送处理模块、注册/注销模块、内部存储管理模块、路由模块、消息分发处理模块、总线数据接收处理模块、总线数据接收接口模块、总线数据发送驱动挂载接口模块。

3. 根据权利要求1所述的面向嵌入式异构总线的云原生容器网络系统,其特征在于,所述硬件抽象层接口组件包含句柄注册/注销模块、应用数据发送处理模块、应用容器数据发送模块、应用容器数据接收模块。

4. 根据权利要求1所述的面向嵌入式异构总线的云原生容器网络系统,其特征在于,系统运行流程为:

P1:在管理节点上部署运行网络管理组件的网络管理容器;

P2:在每个节点上以守护进程方式部署运行硬件抽象层组件的守护进程容器;

P3:根据系统配置,在工作节点上部署运行应用组件的应用容器;

P4:硬件抽象层守护进程容器和应用容器分别将共享数据卷/var/run挂载进各自运行的容器中;

P5:硬件抽象层守护进程容器将所有需要接入容器网络的异构总线设备挂载进自身运行的容器中;

P6:硬件抽象层组件在共享数据卷/var/run目录中创建Unix域套接字hal.sock;

P7:网络管理组件根据嵌入式系统建模信息将网络配置信息下发到各个节点硬件抽象层组件;

P8:应用组件向网络管理组件获取与自己相关的网络配置信息;

P9:应用组件的使用者端口和提供者端口分别向硬件抽象层接口组件进行句柄和句柄回调注册;

P10:硬件抽象层接口组件向硬件抽象层组件进行路由表项添加;

P11:应用组件的使用者端口通过硬件抽象层接口组件进行数据发送;

P12:硬件抽象层接口组件通过/var/run/hal.sock向硬件抽象层组件进行数据发送;

P13:硬件抽象层组件接收到来自应用容器中的硬件抽象层接口组件发送的数据;

P14:硬件抽象层组件根据PORT路由表和IP路由表进行路由判断,如果是发送到本节点的数据,就跳转到P15;如果是发送到其它节点的数据,就跳转到P17;

P15:硬件抽象层组件将发送到本节点的数据,根据PORT路由表注册的套接字地址,通过守护进程容器数据发送模块发送到本节点对应的容器中;

P16:本节点的应用容器数据接收模块接收到数据,根据应用组件在硬件抽象层接口组件的句柄注册表中注册的回调函数,将数据推送给应用组件进行应用逻辑处理,并结束本流程;

P17:硬件抽象层组件将数据通过挂接的总线驱动发送接口发送到总线上;

P18:对端节点的总线驱动接收到数据后,根据硬件抽象层组件挂接的驱动接收回调函数将数据推送给总线数据接收处理模块;

P19:总线数据接收处理模块接收到来自总线驱动推送的数据,继续通过P14的路由模块进行数据路由判断。

5.根据权利要求4所述的面向嵌入式异构总线的云原生容器网络系统,其特征在于,在数据发送端,所述应用组件向硬件抽象层接口组件进行句柄注册的处理流程为:

Q1:应用组件的使用者端口调用硬件抽象层接口组件的句柄注册接口进行句柄注册;

Q2:硬件抽象层接口组件首先根据输入的端口名称进行参数校验,如果校验不成功直接跳转至结束,如果校验成功则进入Q3;

Q3:硬件抽象层接口组件根据端口名称查找网管配置下发信息中对应的端口句柄配置信息,如果查找失败直接跳转至结束,如果查找成功则进入Q4;

Q4:硬件抽象层接口组件将端口注册信息封包,并将封包数据通过Unix域套接字/var/run/hal.sock发往硬件抽象层组件中;

Q5:硬件抽象层组件将端口注册信息添加到PORT路由配置表和IP路由配置表中,并记录该端口对应的共享套接字地址。

6.根据权利要求4所述的面向嵌入式异构总线的云原生容器网络系统,其特征在于,在数据接收端,所述应用组件向硬件抽象层接口组件进行句柄回调注册的处理流程为:

R1:应用组件的提供者端口调用硬件抽象层接口组件的句柄回调注册接口进行回调注册;

R2:硬件抽象层接口组件首先根据输入信息进行参数校验,如果校验不成功直接跳转至结束,如果校验成功则进入R3;

R3:硬件抽象层接口组件进行句柄类型判断,如果句柄类型不是提供者类型直接跳转至结束,如果句柄类型是提供者类型则进入R4;

R4:硬件抽象层接口组件判断句柄是否已经进行回调注册,如果已完成注册则直接跳转至结束,如果未进行注册则进入R5;

R5:硬件抽象层接口组件将注册回调信息添加到句柄注册表中;

R6:硬件抽象层接口组件将端口注册信息封包,并将封包数据通过Unix域套接字/var/run/hal.sock发往硬件抽象层组件;

R7:硬件抽象层组件将端口注册信息添加到PORT路由配置表中,并记录该端口对应的共享sock地址。

7. 根据权利要求4所述的面向嵌入式异构总线的云原生容器网络系统,其特征在于,所述应用组件通过硬件抽象层组件进行数据发送的处理流程为:

S1:应用组件调用硬件抽象层接口组件的数据发送接口进行数据发送;

S2:硬件抽象层接口组件进行数据参数合法性校验,如果校验失败则直接跳转至结束,如果校验成功则继续;

S3:硬件抽象层接口组件根据句柄值获取对应的IP和PORT,完成数据包封装并存储至缓存;

S4:硬件抽象层接口组件将数据通过共享Unix域套接字发送至硬件抽象层组件;

S5:硬件抽象层组件通过共享Unix域套接字完成数据接收;

S6:硬件抽象层组件的路由机构根据PORT值判断目的地址是否为本地应用组件,如果是本地组件跳转至S12,如果不是本地组件则进入S7;

S7:硬件抽象层组件的消息分发处理模块进行发送方式判断,如果是阻塞式发送,就跳转至S8,如果是非阻塞式发送,就跳转至S9;

S8:硬件抽象层组件的消息分发处理模块根据IP在IP/物理地址映射表中查找对应的物理地址信息,如果查找失败,则直接跳转至结束,如果查找成功,就跳转至S11;

S9:硬件抽象层组件的消息分发处理模块将数据包发送至缓存队列等待发送;

S10:硬件抽象层组件的消息分发处理模块的发送任务从缓存队列中取出待发送的数据,根据IP在IP/物理地址映射表中查找对应的物理地址信息,如果查找失败,就直接跳转至结束,如果查找成功,就跳转至S11;

S11:根据物理地址信息,消息分发处理模块调用总线数据发送驱动挂接表中挂接的总线驱动发送接口进行数据发送,并结束本流程;

S12:硬件抽象层组件的守护进程容器数据发送模块,根据发送目的地的PORT值查找对应目的容器注册的共享Unix域套接字地址,并通过该套接字地址将数据路由给本地应用容器。

8. 根据权利要求4所述的面向嵌入式异构总线的云原生容器网络系统,其特征在于,所述应用组件通过硬件抽象层组件进行数据接收的处理流程为:

T1:硬件抽象层组件的总线数据接收处理模块,根据在总线驱动上挂接的接收回调注册函数,从异构总线接收到数据,并将数据发送至缓存队列;

T2:硬件抽象层组件的总线数据接收处理模块的接收任务从缓存队列中取出数据,并根据IP判断数据目的地是否为当前节点,如果是当前节点,就跳转至T3,如果不是当前节点,就跳转至T4;

T3:硬件抽象层组件的消息分发处理模块,根据PORT值将数据包通过守护进程容器数据发送模块路由给本地应用组件;

T4:硬件抽象层组件的消息分发处理模块,根据IP查找IP/物理地址映射表,如果查找失败就直接跳转至结束,如果查找成功,就跳转至T5;

T5:硬件抽象层组件的总线数据发送驱动挂接接口模块,根据物理地址信息,调用总线数据发送驱动挂接表中挂接的总线驱动发送接口进行数据发送。

一种面向嵌入式异构总线的云原生容器网络系统

技术领域

[0001] 本发明涉及容器网络技术领域,具体涉及一种面向嵌入式异构总线的云原生容器网络系统。

背景技术

[0002] 作为诞生于云计算时代的新技术理念,云原生拥有传统计算技术无法比拟的优势,它不仅能够提高应用的弹性、可扩展性、敏捷性、可靠性、容错性、可观测性以及持续更新能力,还显著降低了系统管理运维的难度,同时兼具安全性,这些优势大大降低了系统运行的技术成本,提升了应用开发的交付效率,缩短了应用的上线时间,有效提升业务的创新能力,成为驱动产业发展和业务增长的重要引擎。如今,云原生已经从最初的领先企业尝试变成了主流企业必备,到进一步渗透至各行各业的广泛应用。

[0003] 在云原生涉及的众多技术中,最耀眼的当属以Kubernetes为代表的容器编排引擎,其它所有技术或多或少都需遵循kubernetes建立的标准,云原生容器网络便是其中之一。遵循Kubernetes的云原生网络设计的前提是每台服务器上的容器都有自己独立的IP段,各个服务器之间的容器可以根据目标容器的IP地址进行访问,而该IP地址与主机IP地址不在相同网段。为实现上述目标,云原生网络需要重点解决两个问题:(1)各处理器节点上的容器IP段不能重叠,因此需要有特定的IP段分配机制;(2)从某个pod发出的流量达到其所在节点后,节点需要将流量转发到不在同一网段的目标节点上。目前存在众多的容器网络插件能够实现上述技术目标,代表性的有flannel、calico、weave、canal、contiv、cilium等。但这些网络插件实现的基础都是基于以太网通信的。

[0004] 如今,云原生技术已经迁移到嵌入式边缘系统,嵌入式系统一个显著的特点就是存在大量异构数据传输总线,其目的是为满足综合化嵌入式系统中各个功能对信息共享的不同需求。当前广泛应用的嵌入式数据总线主要有RapidIO、AFDX、光纤FC、ARINC429、1553B、1394B、RS422以及RS485等。各种总线的技术特点、拓扑结构、传输协议、总线带宽以及物理接口方式等均不相同。不同类型嵌入式设备采用的总线类型不尽相同,同一类型设备的不同系统采用的总线类型也未必相同,因此,多种总线混合是普遍存在的现象。

[0005] 现有云原生架构实现技术及云原生网络技术在应用于采用异构总线的嵌入式系统时,在容器网络交互方面还存在诸多不足,包括:

1、缺乏对异构总线的接入支撑机制:目前云原生网络的网络插件设计、服务交互、服务网络设计都是基于以太网通信的,所采用的网络虚拟化技术、网络加入机制、网络退出机制,无法直接应用到异构总线上,实现对采用异构总线的容器的网络接入。

[0006] 2、缺乏对异构总线的网络配置机制:目前云原生网络的网络地址分配由IP地址管理(IPAM)插件解决,该插件主要通过host-local和dhcp(动态主机配置协议)来进行网络配置,而这两种方式也主要完成对以太网的地址分配,无法实现对异构总线的地址分配。

[0007] 3、缺乏对异构总线的容器通信机制:目前云原生网络插件主要解决基于以太网的节点间通信、容器间通信和容器到主机间的通信,通信所采用的地址和协议都是基于以太

网的,无法直接应用到异构总线上,实现对采用异构总线的容器间的通信。

发明内容

[0008] 针对现有技术中的上述不足,本发明提供的一种面向嵌入式异构总线的云原生容器网络系统解决了现有云原生架构实现技术及云原生网络技术缺乏对异构总线的接入支撑机制、缺乏对异构总线的网络配置机制以及缺乏对异构总线的容器通信机制的问题。

[0009] 为了达到上述发明目的,本发明采用的技术方案为:一种面向嵌入式异构总线的云原生容器网络系统,包括:挂载异构总线驱动并实现异构总线统一封装的硬件抽象层组件;对接应用数据并实现应用句柄及回调注册和应用数据收发的硬件抽象层接口组件;以及实现所有网络配置下发和网络信息查询的网络管理组件;

在嵌入式异构总线环境下,所述硬件抽象层组件部署在工作节点的硬件抽象层守护进程容器以及管理节点中,所述网络管理组件部署在管理节点中,所述硬件抽象层接口组件部署在工作节点的应用容器中;

硬件抽象层守护进程容器和应用容器分别将共享数据卷/var/run挂载进各自运行的容器中,且硬件抽象层守护进程容器将所有需要接入容器网络的异构总线设备挂载进自身运行的容器中;硬件抽象层组件在共享数据卷/var/run目录中创建Unix域套接字hal.sock;网络管理组件根据嵌入式系统建模信息完成网络配置信息到各个节点硬件抽象层组件的下发;应用容器通过该hal.sock向硬件抽象层组件进行路由表项注册和应用数据收发;硬件抽象层组件通过路由信息和注册回调信息完成容器间和节点间的数据路由和数据收发。

[0010] 本发明的有益效果为:本发明采用硬件抽象层组件实现对异构总线数据发送和接收的统一抽象,通过设备挂载提供对异构总线的接入支撑机制,支持容器内应用以透明方式实现通过异构总线完成数据发送和接收,解决容器化应用在嵌入式领域难以移植的问题。

[0011] 本发明的硬件抽象层守护进程容器和应用容器间通过共享数据卷和Unix域套接字实现总线数据在容器间的高效发送和接收,基于Unix域套接字的高效性,解决了传统容器网络采用隧道和vxlan等方式需要经过复杂的协议栈处理带来的通信效率低的问题。

[0012] 本发明采用网络管理组件实现对基于异构总线的容器网络配置信息的下发和管理,该网络管理组件结合嵌入式领域应用先验信息,实现对容器网络的高效配置管理,解决现有容器网络需要通过路由学习等方式带来的网络启动慢,配置时间长的问题。

[0013] 本发明以硬件抽象层组件为主体,结合网络管理组件和应用接口组件,为跨异构总线的容器网络通信和信息共享提供接入支撑、网络配置和容器通信等机制,提升容器间通过异构总线通信的适应性和高效性。

[0014] 本发明适用于实现嵌入式环境下,系统中涉及多异构总线的云原生架构的容器网络产品,可以广泛的应用于如嵌入式边缘平台、嵌入式无人平台的云原生架构的设计和开发领域。

附图说明

[0015] 图1为本发明面向嵌入式异构总线的容器网络系统的原理示意图;

图2为本发明中工作节点的原理图；

图3为本发明的模块组成图；

图4为本发明中应用容器的模块图；

图5为本发明的运行流程图；

图6为本发明的应用组件在数据发送端的注册流程图；

图7为本发明的应用组件在数据接收端的注册流程图；

图8为本发明的应用组件通过面向异构总线的容器网络进行数据发送的处理流程图；

图9为本发明的应用组件通过面向异构总线的容器网络进行数据接收的处理流程图。

具体实施方式

[0016] 下面对本发明的具体实施方式进行描述,以便于本技术领域的技术人员理解本发明,但应该清楚,本发明不限于具体实施方式的范围,对本技术领域的普通技术人员来讲,只要各种变化在所附的权利要求限定和确定的本发明的精神和范围内,这些变化是显而易见的,一切利用本发明构思的发明创造均在保护之列。

[0017] 如图1和图2所示,一种面向嵌入式异构总线的容器网络系统,包括:挂载异构总线驱动并实现异构总线统一封装的硬件抽象层组件,对接应用数据并实现应用句柄及回调注册和应用数据收发的硬件抽象层接口组件,以及实现所有网络配置下发和网络信息查询的网络管理组件。在嵌入式异构总线环境下,所述硬件抽象层组件部署在工作节点的硬件抽象层守护进程容器以及管理节点中,所述网络管理组件部署在管理节点中,所述硬件抽象层接口组件部署在工作节点的应用容器中。首先,硬件抽象层守护进程容器和应用容器分别将共享数据卷/var/run挂载进各自运行的容器中,且硬件抽象层守护进程容器将所有需要接入容器网络的异构总线设备挂载进自身运行的容器中;其次,硬件抽象层组件在共享数据卷/var/run目录中创建Unix域套接字hal.sock;继而,网络管理组件根据嵌入式系统建模信息完成网络配置信息到各个节点硬件抽象层组件的下发;再次,应用容器通过该hal.sock向硬件抽象层组件进行路由表项注册和应用数据收发;最后,硬件抽象层组件通过路由信息和注册回调信息完成容器间和节点间的数据路由和数据收发。

[0018] 工作节点是用于部署运行应用业务容器的节点,管理节点是用于部署运行网络管理组件,进行网配置下发和网络信息查询的节点。

[0019] 应用容器是以普通方式部署的运行应用组件和硬件抽象层接口组件的容器。

[0020] 应用组件是嵌入式系统中包含的各种应用层功能模块,完成特定的系统应用功能。

[0021] 硬件抽象层接口组件为应用提供总线统一封装和跨异构总线的数据分发与接收接口,该组件是硬件抽象层组件在各个应用容器中的接口代理,需要与硬件抽象层组件通信,才能完成跨异构总线的通信。

[0022] 共享Unix域套接字/var/run/hal.sock是在共享数据卷/var/run目录下创建的Unix域套接字,由于共享数据卷/var/run是通过数据卷方式挂载到硬件抽象层守护进程容器和应用容器中的共享文件目录,在该目录下创建的所有Unix域套接字能够被守护进程容

器和应用容器共享。

[0023] 硬件抽象层守护进程容器是以守护进程方式部署的运行硬件抽象层组件的容器。

[0024] 硬件抽象层组件是完成异构总线统一封装和跨异构总线数据分发与接收功能的主体部分,该组件用于对各种异构总线驱动进行挂载,并为上层应用提供统一的抽象接口,使应用以无感方式通过异构总线进行数据发送和接收。

[0025] 网络管理组件实现对基于异构总线通信的网络配置信息的下发和管理。在嵌入式综合化系统中,系统运行涉及的组件,组件之间的逻辑连接关系一般都是基于先验信息通过系统建模完成的。系统建模中会对通信交互过程涉及的应用组件的端口名称、端口ID、虚拟IP、应用PORT、提供者/使用者类型等不会随着组件部署位置发生变化的静态信息进行规划。网络管理组件根据网络规划信息,完成配置信息生成,并将生成的配置信息下发到各个节点的硬件抽象层组件中,硬件抽象层组件需要把与应用组件相关的配置信息发送给应用组件。鉴于系统建模不是本专利的主要发明内容,故在此处不做过多阐述。

[0026] 参阅图3和图4,在以下描述的一个最佳实施例中,一个工作节点上的硬件抽象层组件包括守护进程容器数据接收模块、守护进程容器数据发送模块、硬件抽象层数据发送处理模块、注册/注销模块、内部存储管理模块、路由模块、消息分发处理模块、总线数据接收处理模块、总线数据接收接口模块、总线数据发送驱动挂载接口模块。

[0027] 守护进程容器数据接收模块通过共享Unix域套接字/var/run/hal.sock完成所有从应用容器发出的数据的接收,并将接收到的数据发送给硬件抽象层数据发送处理模块。

[0028] 守护进程容器数据发送模块接收来自消息分发处理模块发送到本节点的数据的接收,并将接收到的数据通过共享Unix域套接字/var/run/hal.sock发送到本节点的应用容器中。

[0029] 注册/注销模块完成端口信息到路由表项的注册/注销,用于支撑路由模块根据路由表配置信息完成消息分发处理。

[0030] 硬件抽象层数据发送处理模块根据接收到的来自容器数据接收模块的数据,解析数据目的端口和虚拟IP,并查询路由模块,获取消息分发的路由表项。

[0031] 路由模块包含两个路由表:虚拟IP路由表和PORT路由表,其中IP路由表用于完成节点间数据的发送,PORT路由表用于完成节点内容器间数据的发送。在进行数据分发时,该模块根据输入数据的目的端口和IP,查询PORT路由表和虚拟IP路由表,并将结果发送给消息分发处理模块。需要注意的是,硬件抽象层中的虚拟IP和PORT,是在异构总线上构造的一层虚拟网络层,该虚拟网络层是不经过操作系统内核网络协议栈处理的,所有通过虚拟IP和PORT的路由只经过硬件抽象层组件进行处理。

[0032] 消息分发处理模块根据路由模块处理结果,如果是发送到本节点的数据,调用容器数据发送模块,完成数据在本节点容器间的发送;如果是发送到其它节点的数据,调用总线数据发送驱动挂载接口模块,完成数据在节点间的转发。

[0033] 总线数据接收处理模块根据接收到的来自异构总线的数据,解析数据目的端口和IP,并查询路由模块,获取消息分发的路由表项。

[0034] 内部存储管理模块完成所有通过硬件抽象层组件发送和接收的数据的缓存和记录,实现对数据收发全流程监控、跟踪、调显和故障处理。

[0035] 总线数据接收接口模块完成异构总线接收回调函数的注册,异构总线驱动收到数

据时根据注册的回调函数完成接收数据的推送。

[0036] 总线数据发送驱动挂接口模块完成异构总线发送驱动接口的挂载,消息分发处理模块在通过异构总线进行数据发送时,根据挂载的驱动接口完成数据在节点间的发送。

[0037] 硬件抽象层接口组件为上层应用组件提供通过硬件抽象层组件进行通信的句柄注册/注销、数据发送、数据接收接口,该组件包含句柄注册/注销模块、应用数据发送处理模块、应用容器数据发送模块、应用容器数据接收模块。

[0038] 句柄注册/注销模块为应用组件提供获取句柄、注册句柄回调、注销句柄等操作,句柄中包含了应用组件发送和接收数据所需的所有配置信息,包括端口名称、端口ID、虚拟IP、应用PORT、提供者/使用者类型。

[0039] 应用数据发送处理模块接收来自应用组件发送的数据,并通过发送句柄将其发送到容器数据发送模块,发送句柄中携带了目的地址的虚拟IP和PORT。

[0040] 应用容器数据发送模块接收来自应用数据发送模块的数据,并将其通过共享Unix域套接字/var/run/hal.sock发送到到硬件抽象层守护进程容器中。

[0041] 应用容器数据接收模块通过共享Unix域套接字/var/run/hal.sock完成所有从硬件抽象层守护进程容器发出到该应用容器的数据的接收,并将接收到的数据根据注册的回调句柄推送给应用组件。

[0042] 参照图5,所述面向嵌入式异构总线的容器网络系统运行流程如下:

P1:在管理节点上部署运行网络管理组件的网络管理容器;

P2:在每个节点上以守护进程方式部署运行硬件抽象层组件的守护进程容器;

P3:根据系统配置,在工作节点上部署运行应用组件的应用容器;

P4:硬件抽象层守护进程容器和应用容器分别将共享数据卷/var/run挂载进各自运行的容器中;

P5:硬件抽象层守护进程容器将所有需要接入容器网络的异构总线设备挂载进自身运行的容器中;

P6:硬件抽象层组件在共享数据卷/var/run目录中创建Unix域套接字hal.sock;

P7:网络管理组件根据嵌入式系统建模信息将网络配置信息下发到各个节点硬件抽象层组件;

P8:应用组件向网络管理组件获取与自己相关的网络配置信息;

P9:应用组件的使用者端口和提供者端口分别向硬件抽象层接口组件进行句柄和句柄回调注册;

P10:硬件抽象层接口组件向硬件抽象层组件进行路由表项添加;

P11:应用组件的使用者端口通过硬件抽象层接口组件进行数据发送;

P12:硬件抽象层接口组件通过/var/run/hal.sock向硬件抽象层组件进行数据发送;

P13:硬件抽象层组件接收到来自应用容器中的硬件抽象层接口组件发送的数据;

P14:硬件抽象层组件根据PORT路由表和IP路由表进行路由判断,如果是发送到本节点的数据,就跳转到P15继续处理;如果是发送到其它节点的数据,就跳转到P17继续处理;

P15:硬件抽象层组件将发送到本节点的数据,根据PORT路由表注册的套接字地

址,通过守护进程容器数据发送模块发送到本节点对应的容器中;

P16:本节点的应用容器数据接收模块接收到数据,根据应用组件在硬件抽象层接口组件的句柄注册表中注册的回调函数,将数据推送给应用组件进行应用逻辑处理,并结束本流程;

P17:硬件抽象层组件将数据通过挂接的总线驱动发送接口发送到总线上;

P18:对端节点的总线驱动接收到数据后,根据硬件抽象层组件挂接的驱动接收回调函数将数据推送给总线数据接收处理模块;

P19:总线数据接收处理模块接收到来自总线驱动推送的数据,继续通过P14的路由模块进行数据路由判断。

[0043] 参照图6,在数据发送端,所述应用组件向硬件抽象层接口组件进行句柄注册的处理流程如下:

Q1:应用组件的使用者端口调用硬件抽象层接口组件的句柄注册接口进行句柄注册;

Q2:硬件抽象层接口组件首先根据输入的端口名称进行参数校验,如果校验不成功直接跳转至结束,如果校验成功则继续;

Q3:硬件抽象层接口组件根据端口名称查找网管配置下发信息中对应的端口句柄配置信息,如果查找失败直接跳转至结束,如果查找成功则继续;

Q4:硬件抽象层接口组件将端口注册信息封包,并将封包数据通过Unix域套接字/var/run/hal.sock发往硬件抽象层组件中;

Q5:硬件抽象层组件将端口注册信息添加到PORT路由配置表和IP路由配置表中,并记录该端口对应的共享套接字地址。

[0044] 参照图7,在数据接收端,所述应用组件向硬件抽象层接口组件进行句柄回调注册的处理流程如下:

R1:应用组件的提供者端口调用硬件抽象层接口组件的句柄回调注册接口进行回调注册;

R2:硬件抽象层接口组件首先根据输入信息进行参数校验,如果校验不成功直接跳转至结束,如果校验成功则继续;

R3:硬件抽象层接口组件进行句柄类型判断,如果句柄类型不是提供者类型直接跳转至结束,如果句柄类型是提供者类型则继续执行注册流程;

R4:硬件抽象层接口组件判断句柄是否已经进行回调注册,如果已完成注册则直接跳转至结束,如果未进行注册则继续执行注册流程;

R5:硬件抽象层接口组件将注册回调信息添加到句柄注册表中;

R6:硬件抽象层接口组件将端口注册信息封包,并将封包数据通过Unix域套接字/var/run/hal.sock发往硬件抽象层组件;

R7:硬件抽象层组件将端口注册信息添加到PORT路由配置表中,并记录该端口对应的共享sock地址。

[0045] 参照图8,所述应用组件通过硬件抽象层组件进行数据发送的处理流程如下:

S1:应用组件调用硬件抽象层接口组件的数据发送接口进行数据发送;

S2:硬件抽象层接口组件进行数据参数合法性校验,如果校验失败则直接跳转至

结束,如果校验成功则继续;

S3:硬件抽象层接口组件根据句柄值获取对应的IP和PORT,完成数据包封装并存储至缓存;

S4:硬件抽象层接口组件将数据通过共享Unix域套接字发送至硬件抽象层组件;

S5:硬件抽象层组件通过共享Unix域套接字完成数据接收;

S6:硬件抽象层组件的路由机构根据PORT值判断目的地址是否为本地应用组件,如果是本地组件跳转至S12进行处理,如果不是本地组件就继续;

S7:硬件抽象层组件的消息分发处理模块进行发送方式判断,如果是阻塞式发送,就转到S8,如果是非阻塞式发送,就转到S9;

S8:硬件抽象层组件的消息分发处理模块根据IP在IP/物理地址映射表中查找对应的物理地址信息,如果查找失败,则直接跳转至结束,如果查找成功,就跳转至S11;

S9:硬件抽象层组件的消息分发处理模块将数据包发送至缓存队列等待发送;

S10:硬件抽象层组件的消息分发处理模块的发送任务从缓存队列中取出待发送的数据,根据IP在IP/物理地址映射表中查找对应的物理地址信息,如果查找失败,就直接跳转至结束,如果查找成功,就跳转至S11;

S11:根据物理地址信息,消息分发处理模块调用总线数据发送驱动挂接表中挂接的总线驱动发送接口进行数据发送;

S12:硬件抽象层组件的守护进程容器数据发送模块,根据发送目的地的PORT值查找对应目的容器注册的共享Unix域套接字地址,并通过该套接字地址将数据路由给本地应用容器。

[0046] 参照图9,所述应用组件通过硬件抽象层组件进行数据接收的处理流程如下:

T1:硬件抽象层组件的总线数据接收处理模块,根据在总线驱动上挂接的接收回调注册函数,从异构总线接收到数据,并将数据发送至缓存队列;

T2:硬件抽象层组件的总线数据接收处理模块的接收任务从缓存队列中取出数据,并根据IP判断数据目的地是否为当前节点,如果是当前节点,就跳转至T3,如果不是当前节点,就跳转至T4;

T3:硬件抽象层组件的消息分发处理模块,根据PORT值将数据包通过守护进程容器数据发送模块路由给本地应用组件;

T4:硬件抽象层组件的消息分发处理模块,根据IP查找IP/物理地址映射表,如果查找失败就直接跳转至结束,如果查找成功,就跳转至T5;

T5:硬件抽象层组件的总线数据发送驱动挂接接口模块,根据物理地址信息,调用总线数据发送驱动挂接表中挂接的总线驱动发送接口进行数据发送。

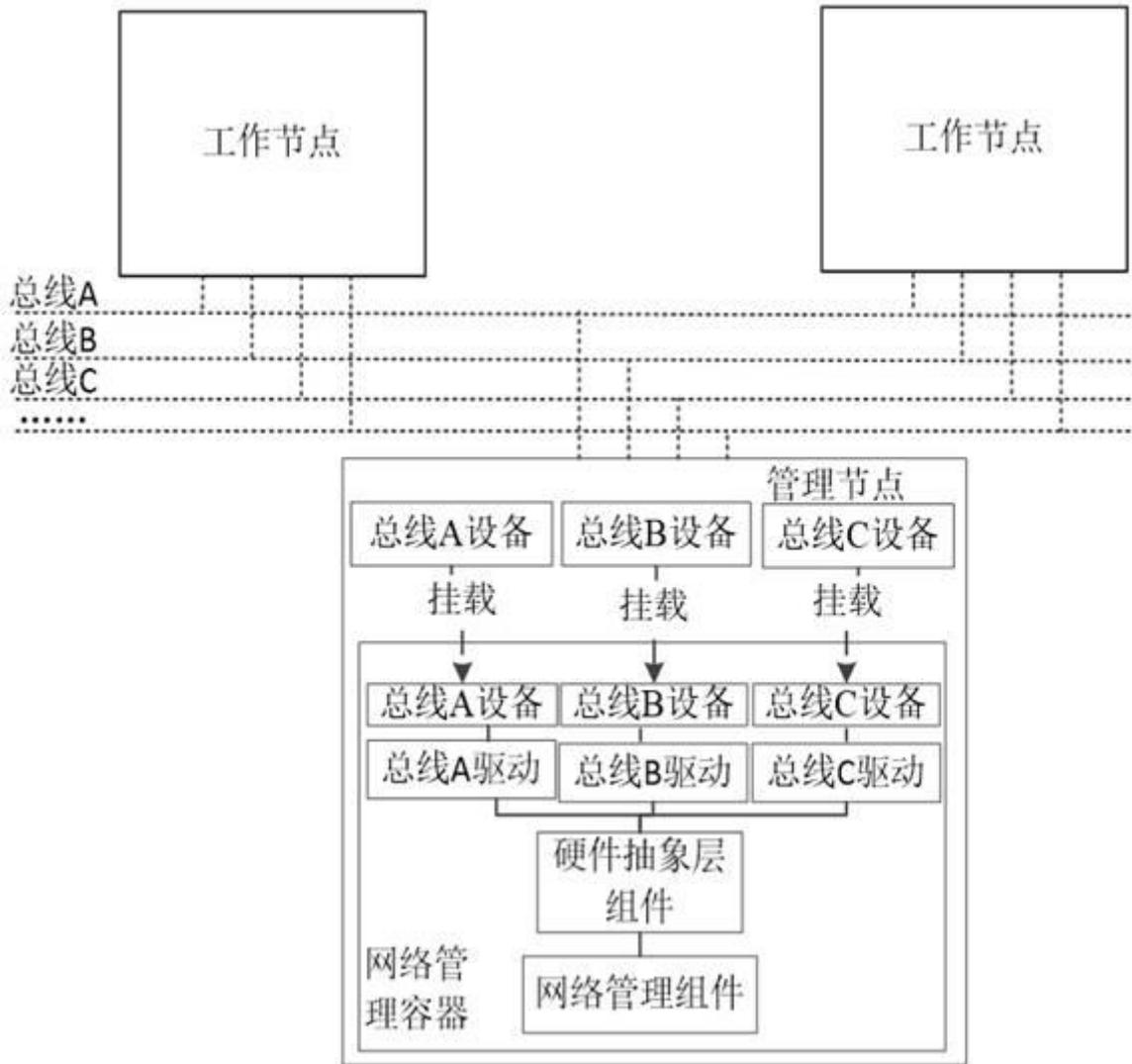


图1

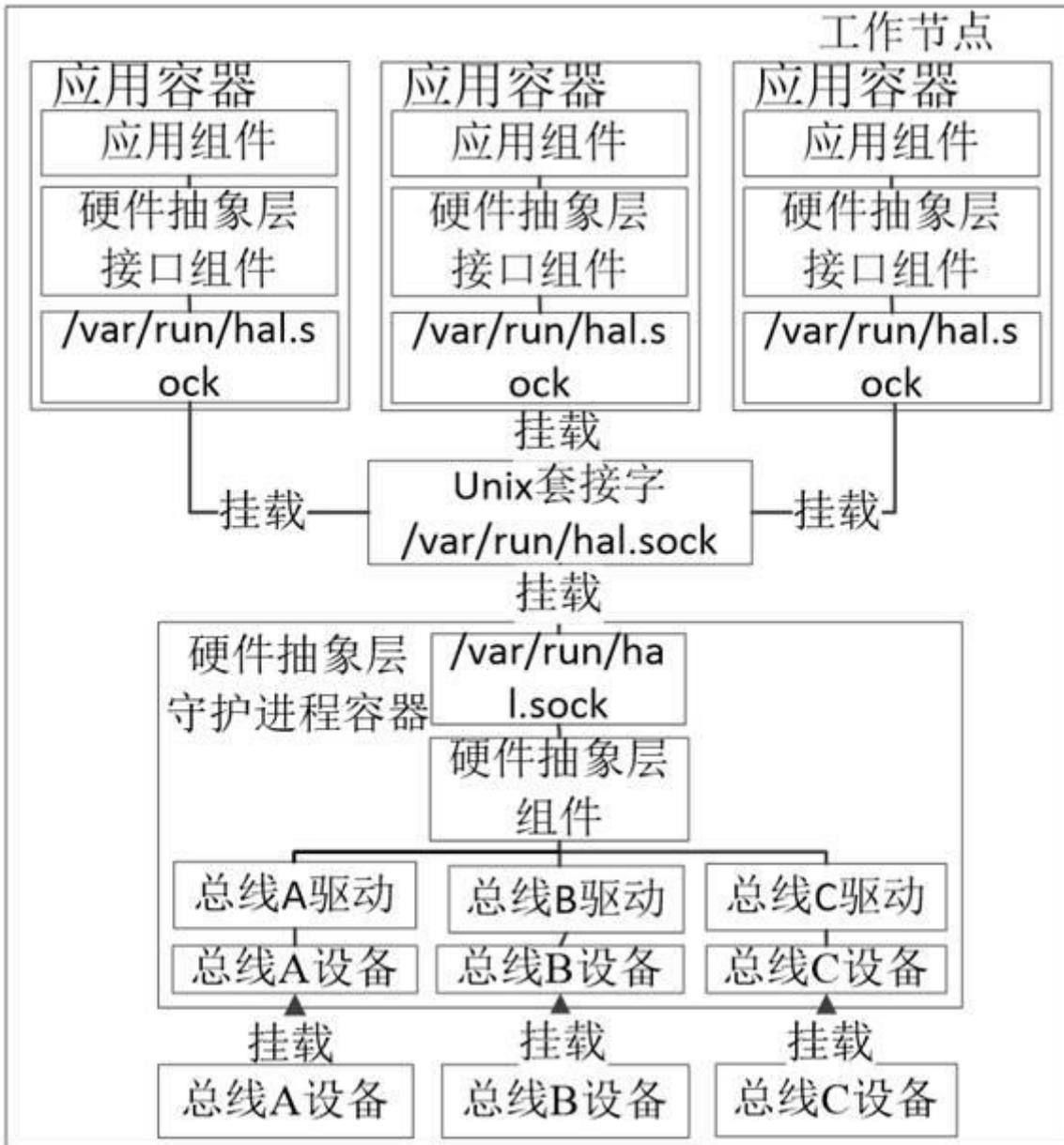


图2

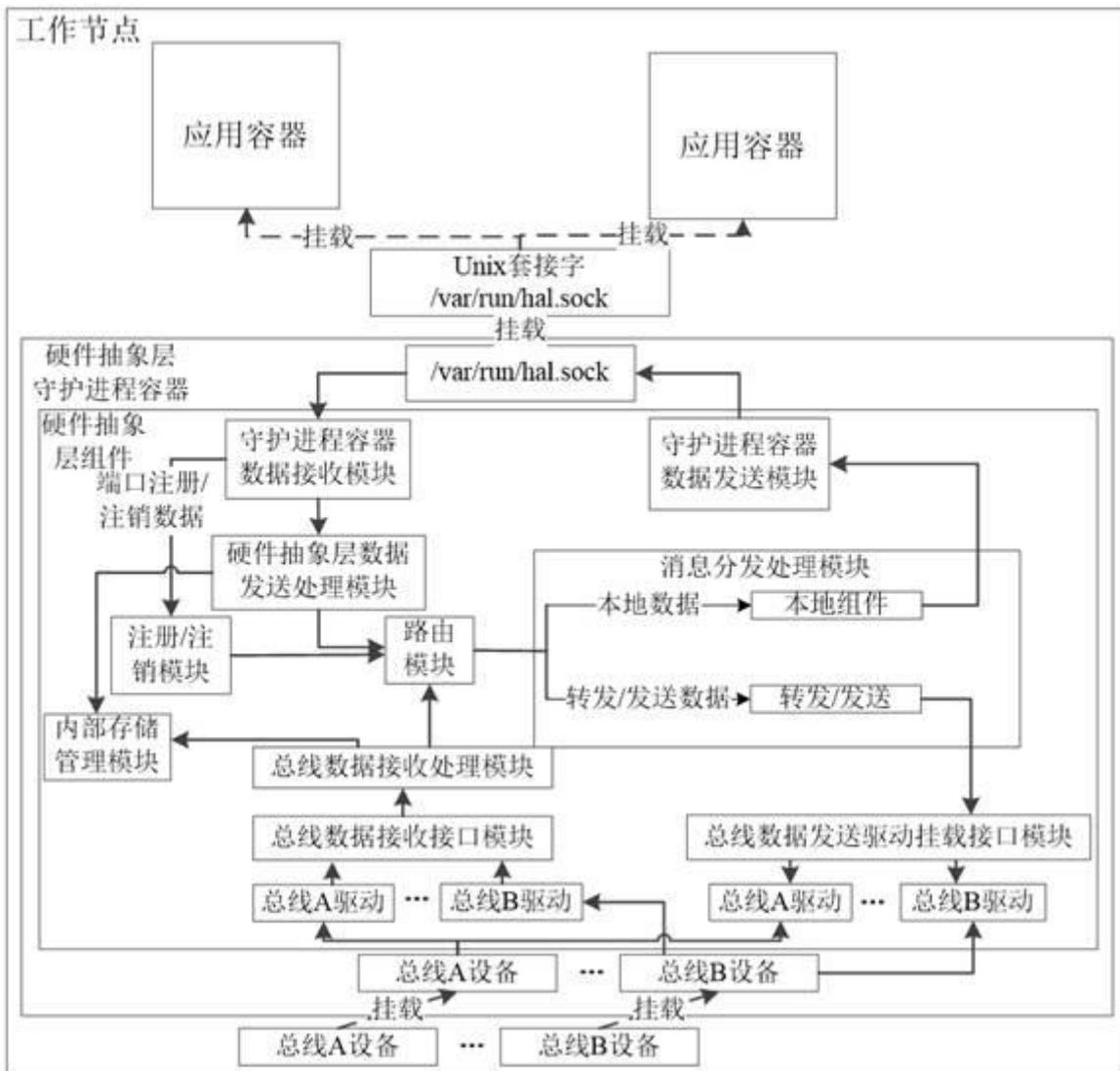


图3

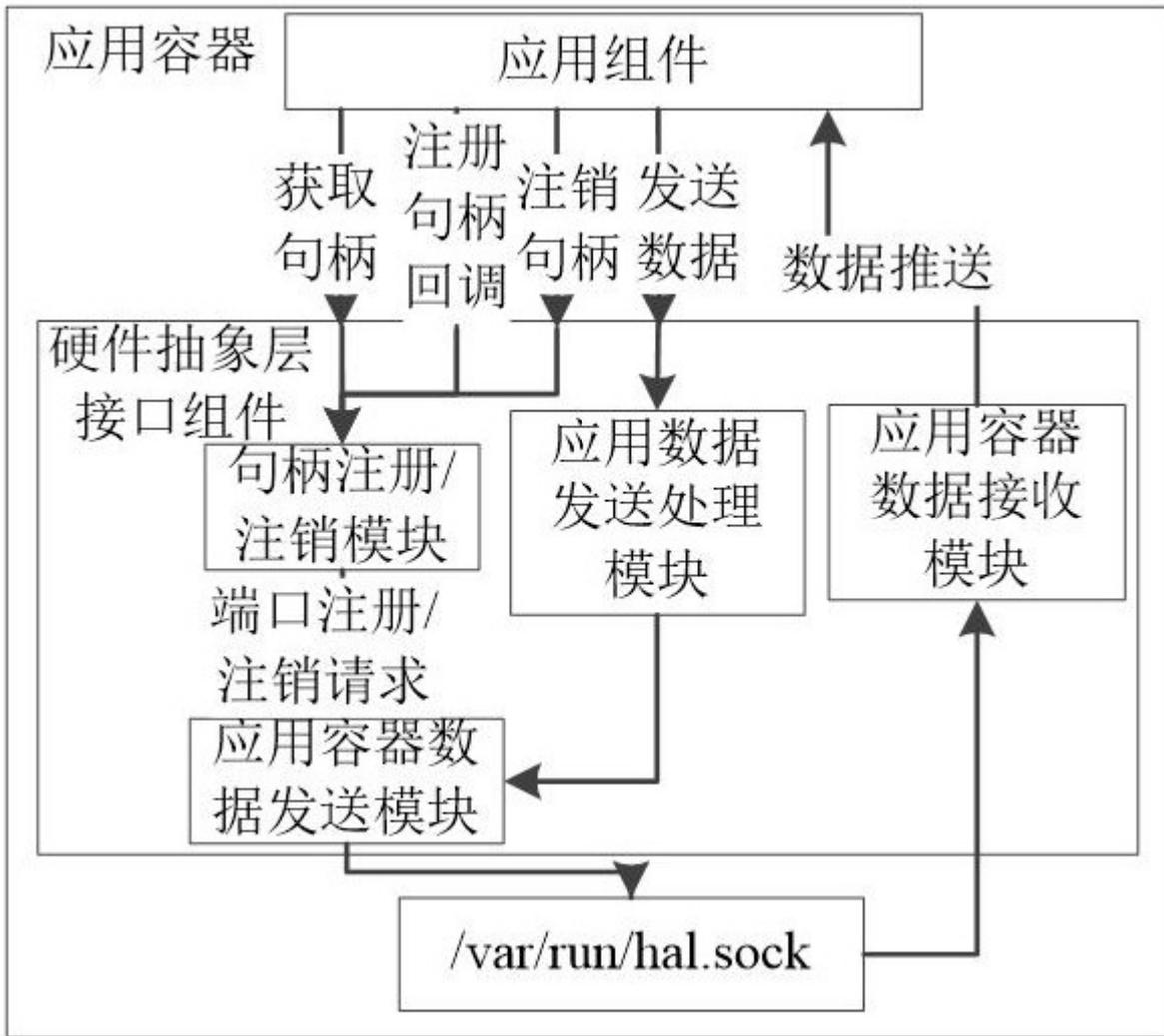


图4

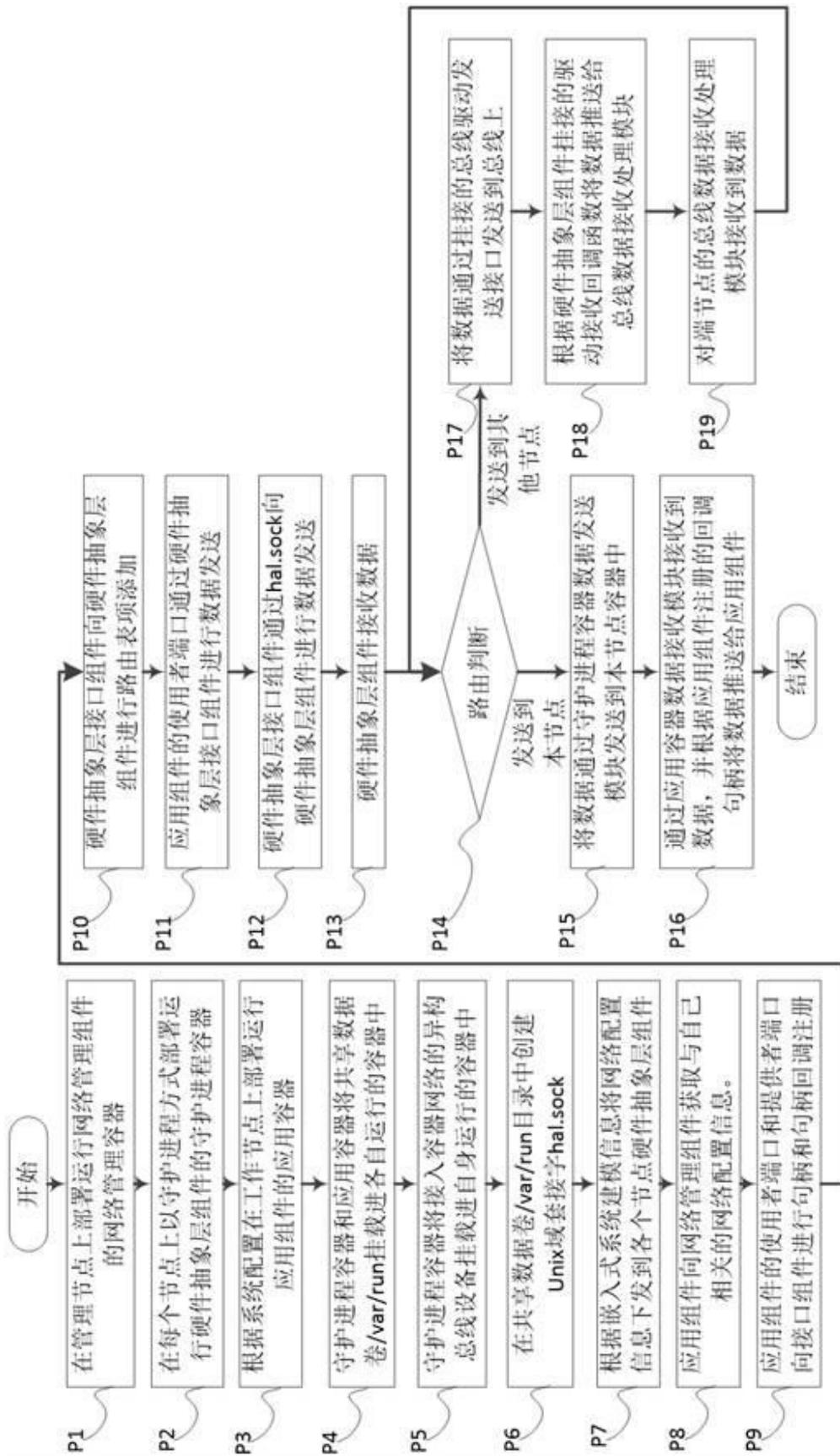


图5

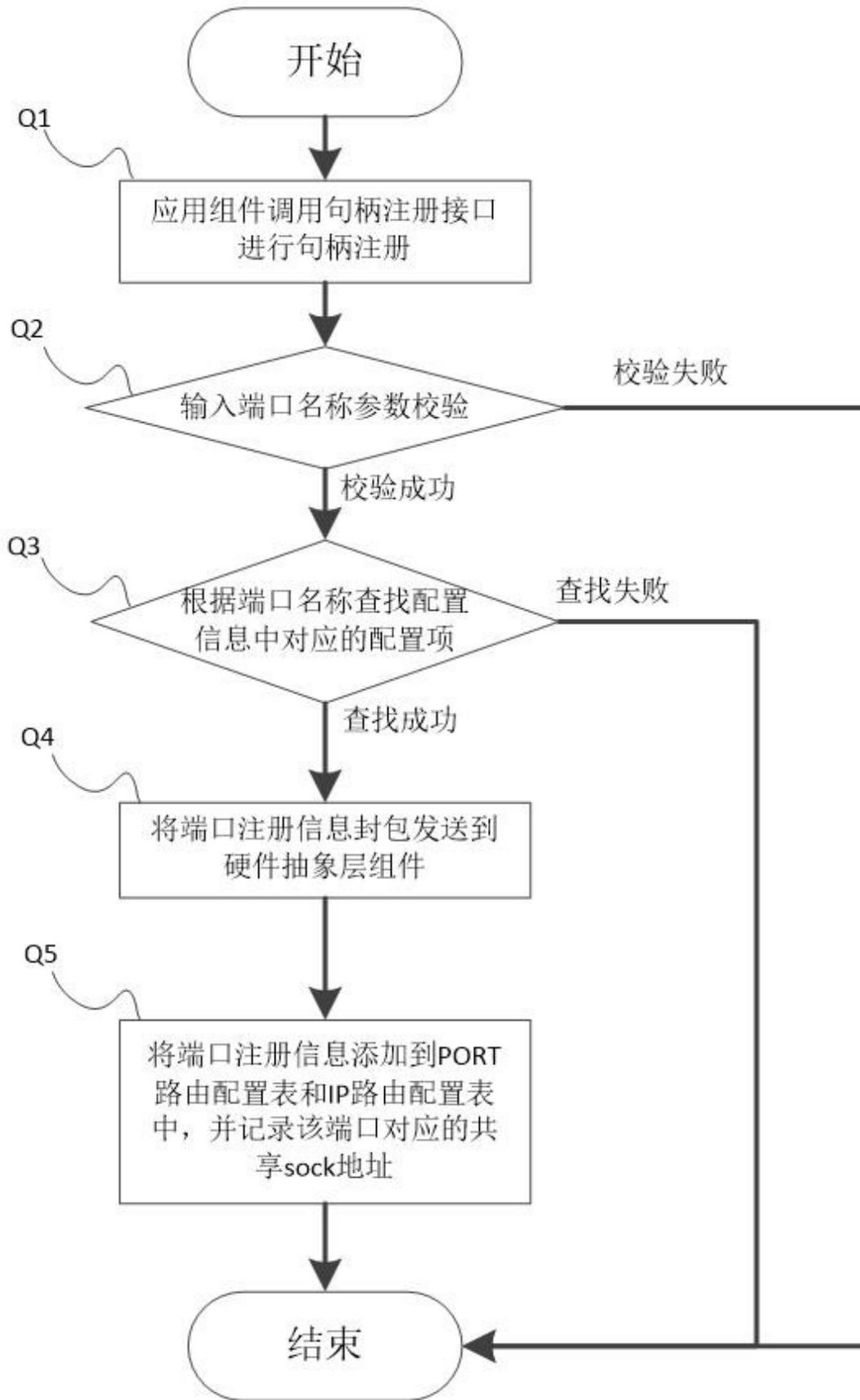


图6

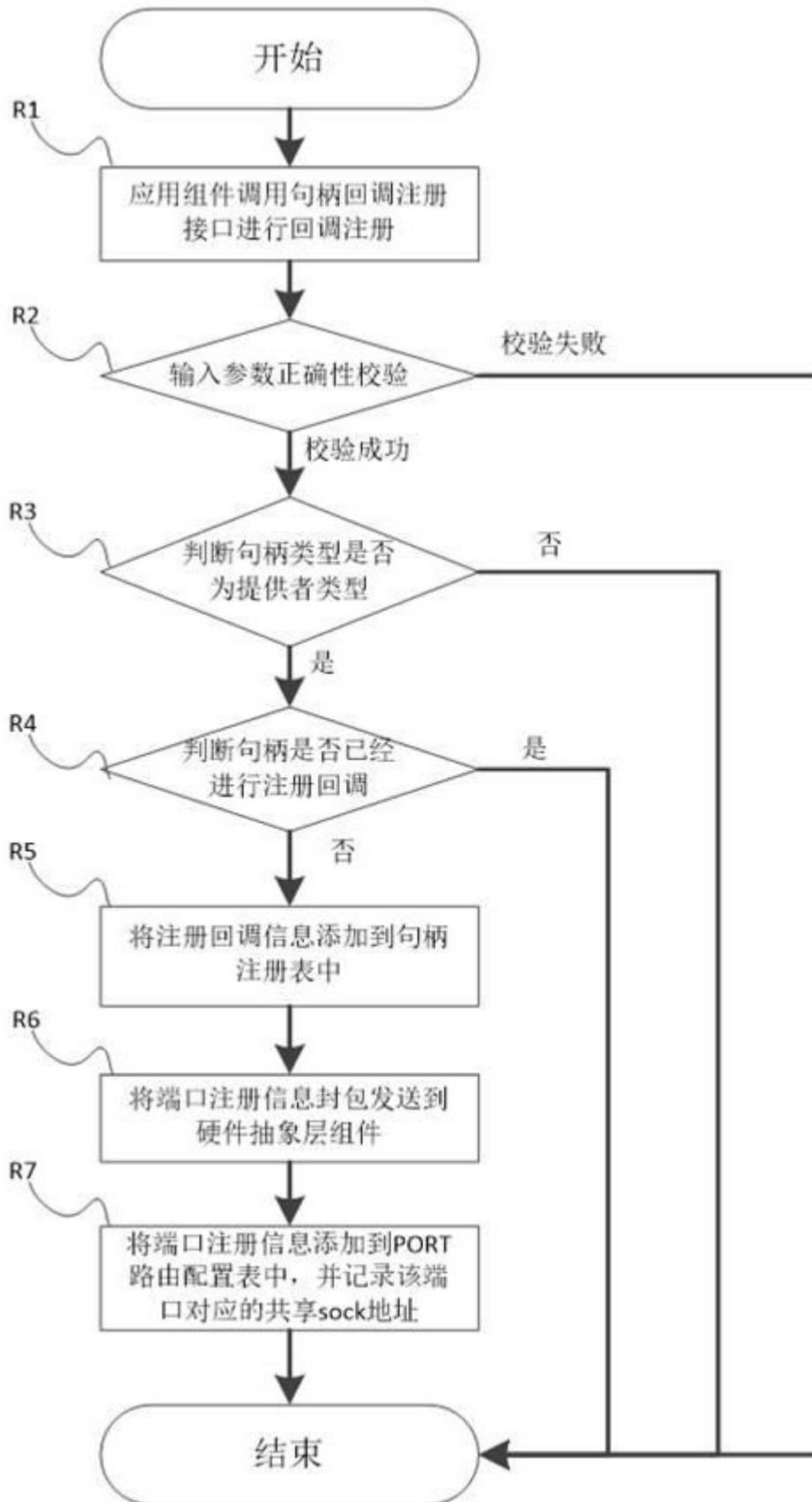


图7

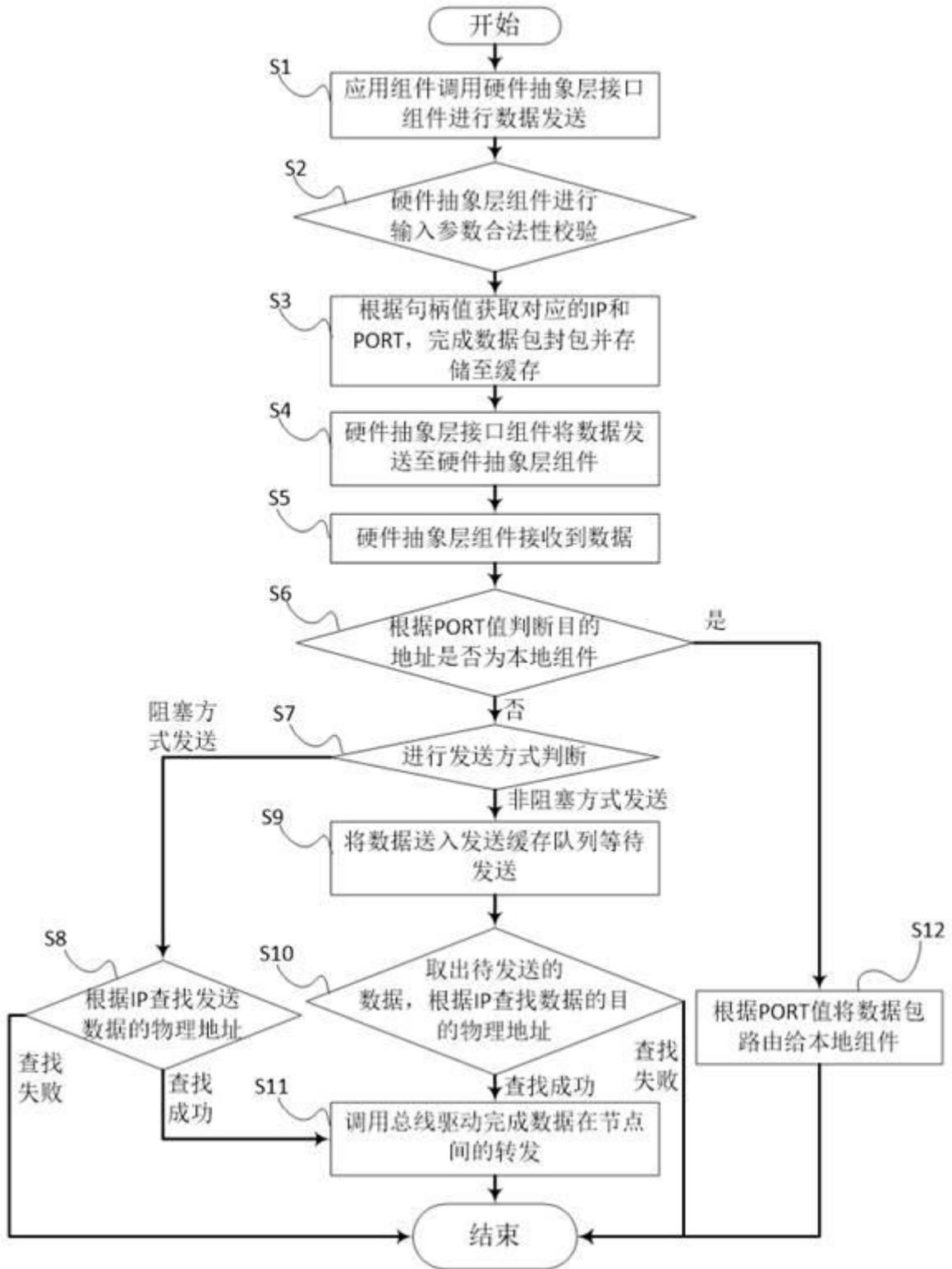


图8

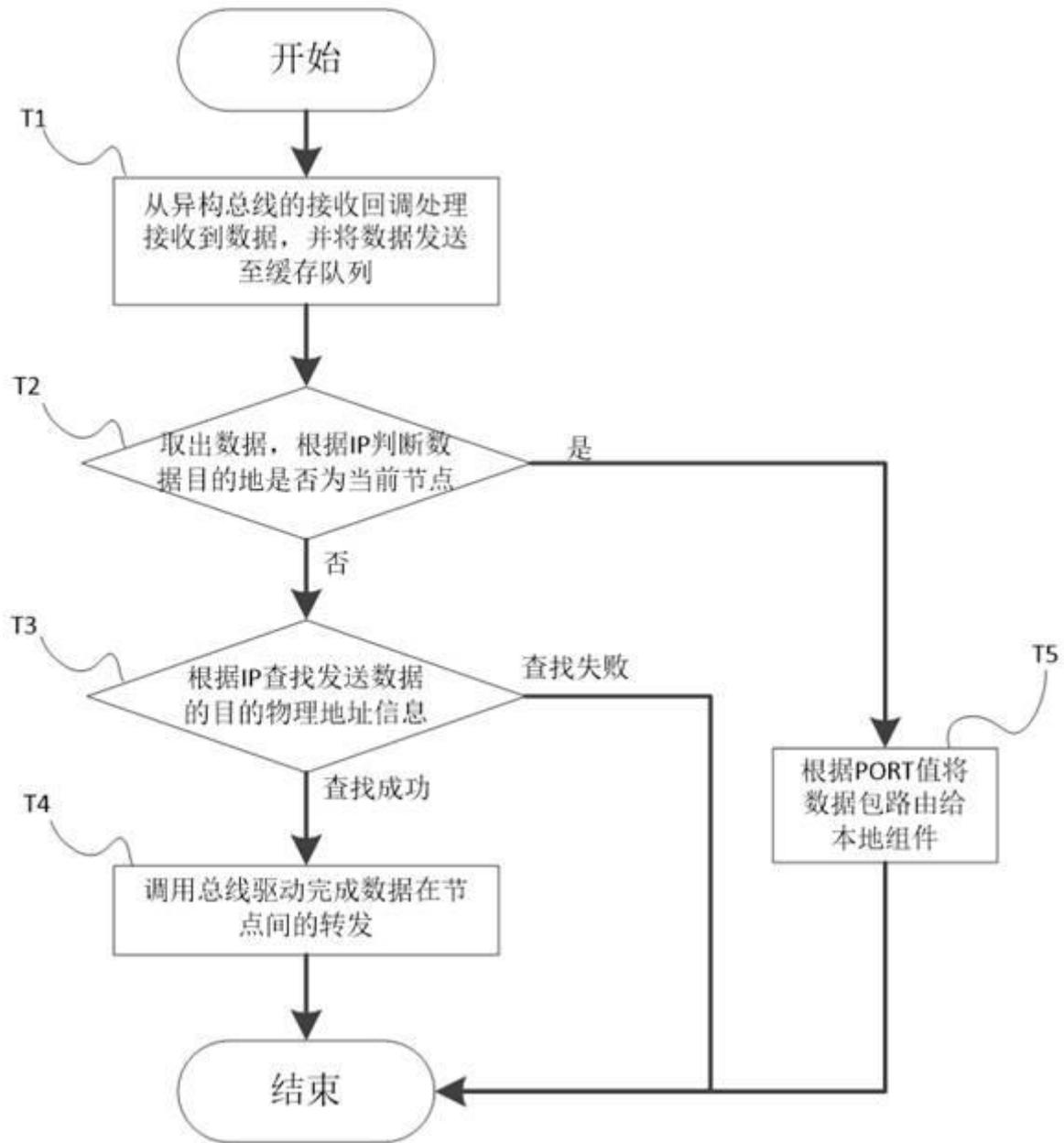


图9