



(12)发明专利

(10)授权公告号 CN 104380661 B

(45)授权公告日 2019.03.26

(21)申请号 201480001392.1

(72)发明人 张科 高月莉

(22)申请日 2014.06.23

(74)专利代理机构 广州三环专利商标代理有限公司 44202

(65)同一申请的已公布的文献号
申请公布号 CN 104380661 A

代理人 郝传鑫 熊永强

(43)申请公布日 2015.02.25

(51)Int.Cl.

(85)PCT国际申请进入国家阶段日
2014.12.02

H04L 12/24(2006.01)

H04L 12/701(2013.01)

G06F 9/44(2006.01)

(86)PCT国际申请的申请数据
PCT/CN2014/080507 2014.06.23

(56)对比文件

CN 102685210 A,2012.09.19,

CN 103177028 A,2013.06.26,

CN 102364435 A,2012.02.29,

US 2008126520 A1,2008.05.29,

(87)PCT国际申请的公布数据
W02015/196330 ZH 2015.12.30

(73)专利权人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

审查员 丛文

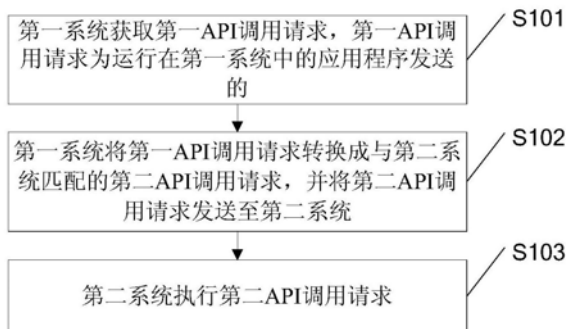
权利要求书3页 说明书21页 附图8页

(54)发明名称

一种路由设备的操作方法、路由设备和终端设备

(57)摘要

本发明实施例公开了一种路由设备的操作方法,所述路由设备包括第一系统和第二系统,所述方法包括:所述第一系统获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在所述第一系统中的应用程序发送的;所述第一系统将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述第二系统;所述第二系统执行所述第二API调用请求。本发明实施例还公开了一种路由设备和终端设备。采用本发明,可降低路由设备上的应用程序的开发难度。



1. 一种路由设备的操作方法,其特征在于,所述路由设备包括第一系统和第二系统,所述第一系统包括安卓操作系统,所述第二系统包括Linux操作系统,所述方法包括:

所述第一系统检测到所述第二系统后,进行双向身份验证,在双向身份验证合法后,所述第一系统与所述第二系统建立通信连接;

所述第一系统获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在所述第一系统中的应用程序发送的;

所述第一系统根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求;或所述第一系统提取所述第一API调用请求中的请求参数,根据所述请求参数生成与所述第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述第二系统;

所述第二系统执行所述第二API调用请求。

2. 根据权利要求1所述的方法,其特征在于,所述第二系统执行所述第二API调用请求之后,所述方法还包括:

所述第二系统向所述第一系统返回第二API调用响应;

所述第一系统将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应,并将所述第一API调用响应返回至所述应用程序。

3. 根据权利要求2所述的方法,其特征在于,所述第一系统将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应,包括:

所述第一系统根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应;或

所述第一系统提取所述第二API调用响应中的响应参数,根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

4. 根据权利要求1-3任意一项所述的方法,其特征在于,所述第一API调用请求包括应用鉴权请求、IP地址配置请求、IP地址获取请求、IP路由协议配置请求、端口配置请求、隧道状态获取请求、隧道配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种或几种。

5. 一种路由设备,其特征在于,所述路由设备包括第一系统和第二系统,所述第一系统包括安卓操作系统,所述第二系统包括Linux操作系统,所述第一系统检测到所述第二系统后,进行双向身份验证,在双向身份验证合法后,所述第一系统与所述第二系统建立通信连接;所述第一系统包括:第一请求获取模块和第一请求转换模块,所述第二系统包括请求执行模块;

所述第一请求获取模块,用于获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在所述第一系统中的应用程序发送的;

所述第一请求转换模块,用于根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求;或提取所述第一API调用请求中的请求参数,根据所述请求参数生成与所述第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述第二系统;

所述请求执行模块,用于执行所述第二API调用请求。

6. 根据权利要求5所述的路由设备,其特征在于,所述第二系统还包括:响应返回模块,所

述第一系统还包括第一响应转换模块；

所述响应返回模块，用于在所述请求执行模块执行所述第二API调用请求之后，向所述第一系统返回第二API调用响应；

所述第一响应转换模块，用于将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应，并将所述第一API调用响应返回至所述应用程序。

7. 如权利要求6所述的路由设备，其特征在于，所述第一响应转换模块在将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应时，具体用于：

根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应；或

提取所述第二API调用响应中的响应参数，根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

8. 根据权利要求5-7任意一项所述的路由设备，其特征在于，所述第一API调用请求包括应用鉴权请求、IP地址配置请求、IP地址获取请求、IP路由协议配置请求、端口配置请求、隧道状态获取请求、隧道状态配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种或几种。

9. 一种路由设备，其特征在于，包括：处理器和存储器，所述存储器中存储一组程序代码，且所述处理器用于调用所述存储器中存储的程序代码，执行以下操作：

检测到第二系统后，进行双向身份验证，在双向身份验证合法后，第一系统与所述第二系统建立通信连接；

获取第一应用程序编程接口API调用请求，所述第一API调用请求为运行在第一系统中的应用程序发送的；

根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求；或提取所述第一API调用请求中的请求参数，根据所述请求参数生成与所述第二系统匹配的第二API调用请求，并将所述第二API调用请求发送至所述第二系统；

执行所述第二API调用请求；

所述第一系统和所述第二系统运行在所述路由设备中，所述第一系统包括安卓操作系统，所述第二系统包括Linux操作系统。

10. 如权利要求9所述的路由设备，其特征在于，在执行所述第二API调用请求之后，所述处理器还用于执行：

向所述第一系统返回第二API调用响应；

将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应，并将所述第一API调用响应返回至所述应用程序。

11. 如权利要求10所述的路由设备，其特征在于，所述处理器执行所述将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应，包括：

根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应；或

提取所述第二API调用响应中的响应参数，根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

12. 如权利要求9-11任意一项所述的路由设备，其特征在于，所述第一API调用请求包

括应用鉴权请求、IP地址配置请求、IP地址获取请求、IP路由协议配置请求、端口配置请求、隧道状态获取请求、隧道配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种或几种。

一种路由设备的操作方法、路由设备和终端设备

技术领域

[0001] 本发明实施例涉及通信技术,尤其涉及一种路由设备的操作方法、路由设备和终端设备。

背景技术

[0002] 智能路由器即智能化管理的路由器,相比于普通路由器,其像个人电脑一样,具有独立的操作系统,可以由用户自行安装各种应用程序。通过安装的应用程序能实现自行控制带宽、自行控制在线人数、自行控制浏览网页和自行控制在线时间等控制功能,做到路由器的智能化管理。

[0003] 在目前的智能路由器系统架构中,智能路由器的操作系统采用Open wrt (Open wireless router,开放无线路由)系统,Open wrt是一个文件系统可写的Linux操作系统,可以基于该操作系统开发和安装应用程序。然而,Open wrt系统不是由路由器厂商专门发布的路由器系统,开发人员在该系统上开发应用程序的难度较高。

发明内容

[0004] 本发明实施例所要解决的技术问题在于,提供一种路由设备的操作方法、路由设备和终端设备,可解决在路由器的系统中开发应用程序难度高的不足。

[0005] 为了解决上述技术问题,本发明实施例第一方面提供了一种路由设备的操作方法,包括:

[0006] 所述第一系统获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在所述第一系统中的应用程序发送的;

[0007] 所述第一系统将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述第二系统;

[0008] 所述第二系统执行所述第二API调用请求。

[0009] 结合第一方面,在第一种可能的实现方式中,所述第二系统执行所述第二API调用请求之后,所述方法还包括:

[0010] 所述第二系统向所述第一系统返回第二API调用响应;

[0011] 所述第一系统将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应,并将所述第一API调用响应返回至所述应用程序。

[0012] 结合第一方面或第一种可能的实现方式,在第二种可能的实现方式中,所述第一系统将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求,包括:

[0013] 所述第一系统根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求;或

[0014] 所述第一系统提取所述第一API调用请求中的请求参数,根据所述请求参数生成与所述第二系统匹配的第二API调用请求。

[0015] 结合第一种可能的实现方式,在第三种可能的实现方式中,所述第一系统将所述

第二API调用响应转换成与所述第一系统匹配的第一API调用响应,包括:

[0016] 所述第一系统根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应;或

[0017] 所述第一系统提取所述第二API调用响应中的响应参数,根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

[0018] 结合第一方面至第三种可能的实现方式中的任意一种,在第四种可能的实现方式中,所述第一API调用请求包括应用鉴权请求、IP地址配置请求、IP地址获取请求、IP路由协议配置请求、端口配置请求、隧道状态获取请求、隧道配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种或几种。

[0019] 结合第一方面至第四种可能的实现方式中的任意一种,在第五种可能的实现方式中,所述第一系统中运行的操作系统包括Android、IOS、Windows Phone和BlackBerry中的任意一种,所述第二系统中运行的操作系统包括Linux或eCOS。

[0020] 本发明第二实施例提供了一种路由设备的操作方法,包括:

[0021] 所述终端设备获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在所述第一系统中的应用程序发送的;

[0022] 所述终端设备将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述路由设备,以使所述路由设备执行所述第二API调用请求。

[0023] 结合第二方面,在第一种可能的实现方式中,所述终端设备将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述路由设备,以使所述路由设备执行所述第二API调用请求之后,所述方法还包括:

[0024] 所述终端设备接收所述第二系统返回的第二API调用响应;

[0025] 所述终端设备将所述第二API调用响应转换成所述第一系统匹配的第一API调用响应,并将所述第一API调用响应返回至所述应用程序。

[0026] 结合第二方面或第一种可能的实现方式,在第二种可能的实现方式中,所述终端设备将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求,包括:

[0027] 所述终端设备根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求;或

[0028] 所述终端设备提取所述第一API调用请求中的请求参数,根据所述请求参数生成与所述第二系统匹配的第二API调用请求。

[0029] 结合第一种可能的实现方式,在第三种可能的实现方式中,所述终端设备将所述第二API调用响应转换成所述第一系统匹配的第一API调用响应,包括:

[0030] 所述终端设备根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应;或

[0031] 所述终端设备提取所述第二API调用响应中的响应参数,根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

[0032] 结合第二方面至第三种可能的实现方式中的任意一种,在第四种可能的实现方式中,所述第一API调用请求包括应用鉴权请求、IP地址配置请求、IP地址获取请求、IP路由协议配置请求、端口配置请求、隧道状态获取请求、隧道配置请求、域名系统DNS信息获取请求

和DNS信息配置请求中的任意一种或几种。

[0033] 结合第二方面至第四种可能的实现方式中的任意一种,在第五种可能的实现方式中,所述第一系统中运行的操作系统包括Android、IOS、Windows Phone和BlackBerry中的任意一种,所述第二系统中运行的操作系统包括Linux或eCOS。

[0034] 本发明实施例第三方面提供了一种路由设备,所述路由设备包括第一系统和第二系统,所述第一系统包括:第一请求获取模块和第一请求转换模块,所述第二系统包括请求执行模块;

[0035] 所述第一请求获取模块,用于获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在所述第一系统中的应用程序发送的;

[0036] 所述第一请求转换模块,用于将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述第二系统;

[0037] 所述请求执行模块,用于执行所述第二API调用请求。

[0038] 结合第三方面,在第一种可能的实现方式中,所述第二系统还包括:响应返回模块,所述第一系统还包括第一响应转换模块;

[0039] 所述响应返回模块,用于在所述请求执行模块执行所述第二API调用请求之后,向所述第一系统返回第二API调用响应;

[0040] 所述第一响应转换模块,用于将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应,并将所述第一API调用响应返回至所述应用程序。

[0041] 结合第三方面或第一种可能的实现方式,在第二种可能的实现方式中,所述第一请求转换模块在将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求时,具体用于:

[0042] 根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求;或

[0043] 提取所述第一API调用请求中的请求参数,根据所述请求参数生成与所述第二系统匹配的第二API调用请求。

[0044] 结合第一种可能的实现方式,在第三种可能的实现方式中,所述第一响应转换模块在将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应时,具体用于:

[0045] 根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应;或

[0046] 提取所述第二API调用响应中的响应参数,根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

[0047] 结合第三方面至第三种可能的实现方式中的任意一种,所述第一API调用请求包括应用鉴权请求、IP地址配置请求、IP地址获取请求、IP路由协议配置请求、端口配置请求、隧道状态获取请求、隧道状态配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种或多种。

[0048] 结合第三方面至第四种可能的实现方式中的任意一种,在第五种可能的实现方式中,所述第一系统中运行的操作系统包括Android、IOS、Windows Phone和BlackBerry中的任意一种,所述第二系统中运行的操作系统包括Linux或eCOS。

[0049] 本发明实施例第四方面提供了一种终端设备,所述终端设备包括第一系统,所述第一系统包括:

[0050] 第二请求获取模块,用于获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在所述第一系统中的应用程序发送的;

[0051] 第二请求转换模块,用于将所述第一API调用请求转换成与包含在路由设备中的第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述路由设备,以使所述路由设备执行所述第二API调用请求。

[0052] 结合第四方面,在第一种可能的实现方式中,所述第一系统还包括:

[0053] 响应接收模块,用于在所述第二请求转换模块将所述第二API调用请求发送至所述路由设备之后,接收所述第二系统返回的第二API调用响应;

[0054] 第二响应转换模块,用于将所述第二API调用响应转换成所述第一系统匹配的第一API调用响应,并将所述第一API调用响应返回至所述应用程序。

[0055] 结合第四方面或第一种可能的实现方式,在第二种可能的实现方式中,所述第二请求转换模块在将所述第一API调用请求转换成与包含在路由设备中的第二系统匹配的第二API调用请求时,具体用于:

[0056] 根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求;或

[0057] 提取所述第一API调用请求中的请求参数,根据所述请求参数生成与所述第二系统匹配的第二API调用请求。

[0058] 结合第一种可能的实现方式,在第三种可能的实现方式中,所述第二响应转换模块在将所述第二API调用响应转换成所述第一系统匹配的第一API调用响应时,具体用于:

[0059] 根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应;或

[0060] 提取所述第二API调用响应中的响应参数,根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

[0061] 结合第四方面至第三种可能的实现方式中的任意一种,在第四种可能的实现方式中,所述第一API调用请求包括应用鉴权请求、IP地址配置请求、IP地址获取请求、IP路由协议配置请求、端口配置请求、隧道状态获取请求、隧道状态配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种或几种。

[0062] 结合第四方面至第四种可能的实现方式中的任意一种,在第五种可能的实现方式中,所述第一系统中运行的操作系统包括Android、IOS、Windows Phone和BlackBerry中的任意一种,所述第二系统中运行的操作系统包括Linux或eCOS。

[0063] 本发明实施例第五方面提供了一种路由设备,包括:处理器和存储器,所述存储器中存储一组程序代码,且所述处理器用于调用所述存储器中存储的程序代码,执行以下操作:

[0064] 获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在第一系统中的应用程序发送的;

[0065] 将所述第一API调用请求转换成与第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至第二系统;

- [0066] 执行所述第二API调用请求；
- [0067] 所述第一系统和所述第二系统运行在所述路由设备中。
- [0068] 结合第五方面，在第一种可能的实现方式中，在执行所述第二API调用请求之后，所述处理器还用于执行：
- [0069] 向所述第一系统返回第二API调用响应；
- [0070] 将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应，并将所述第一API调用响应返回至所述应用程序。
- [0071] 结合第五方面或第一种可能的实现方式，在第二种可能的实现方式中，所述处理器执行所述将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求，包括：
- [0072] 根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求；或
- [0073] 提取所述第一API调用请求中的请求参数，根据所述请求参数生成与所述第二系统匹配的第二API调用请求。
- [0074] 结合第一种可能的实现方式，在第三种可能的实现方式中，所述处理器执行所述将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应，包括：
- [0075] 根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应；或
- [0076] 提取所述第二API调用响应中的响应参数，根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。
- [0077] 结合第五方面至第三种可能的实现方式中的任意一种，在第四种可能的实现方式中，所述第一API调用请求包括应用鉴权请求、IP地址配置请求、IP地址获取请求、IP路由协议配置请求、端口配置请求、隧道状态获取请求、隧道配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种或几种。
- [0078] 结合第五方面至第四种可能的实现方式中的任意一种，在第五种可能的实现方式中，所述第一系统中运行的操作系统包括Android、IOS、Windows Phone和BlackBerry中的任意一种，所述第二系统中运行的操作系统包括Linux或eCOS。
- [0079] 本发明实施例第六方面提供了一种终端设备，包括：处理器和存储器，所述存储器中存储一组程序代码，且所述处理器用于调用所述存储器中存储的程序代码，执行以下操作：
- [0080] 获取第一应用程序编程接口API调用请求，所述第一API调用请求为运行在第一系统中的应用程序发送的；所述第一系统运行在所述终端设备中；
- [0081] 将所述第一API调用请求转换成与包含在路由设备中的第二系统匹配的第二API调用请求，并将所述第二API调用请求发送至所述路由设备，以使路由设备执行所述第二API调用请求。
- [0082] 结合第六方面，在第一种可能的实现方式中，在将所述第二API调用请求发送至所述路由设备之后，所述处理器还用于执行：
- [0083] 接收所述第二系统返回的第二API调用响应；
- [0084] 将所述第二API调用响应转换成所述第一系统匹配的第一API调用响应，并将所述

第一API调用响应返回至所述应用程序。

[0085] 结合第六方面或第一种可能的实现方式,在第二种可能的实现方式中,所述处理器执行所述将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求,包括:

[0086] 根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求;或

[0087] 提取所述第一API调用请求中的请求参数,根据所述请求参数生成与所述第二系统匹配的第二API调用请求。

[0088] 结合第一种可能的实现方式,在第三种可能的实现方式中,所述处理器执行所述将所述第二API调用响应转换成所述第一系统匹配的第一API调用响应,包括:

[0089] 根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应;或

[0090] 提取所述第二API调用响应中的响应参数,根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

[0091] 结合第六方面至第三种可能的实现方式中的任意一种,在第四种可能的实现方式中,所述第一API调用请求包括应用鉴权请求、IP地址配置请求、IP地址获取请求、IP路由协议配置请求、端口配置请求、隧道状态获取请求、隧道配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种或几种。

[0092] 结合第六方面至第四种可能的实现方式中的任意一种,在第五种可能的实现方式中,所述第一系统中运行的操作系统包括Android、IOS、Windows Phone和BlackBerry中的任意一种,所述第二系统中运行的操作系统包括Linux或eCOS。

[0093] 本发明实施例,具有如下有益效果:

[0094] 通过将运行在第一系统中的应用程序发出的API调用请求转换成与第二系统匹配的调用请求,实现了第一系统中的应用程序的跨平台API调用,由于第一系统为通用的开发平台,开发应用程序难度小,开发者因此不需要直接在路由设备的专用的第二系统中开发应用程序,降低了应用程序的开发难度。

附图说明

[0095] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0096] 图1为本发明第一实施例的一种路由设备的操作方法的流程示意图;

[0097] 图2为本发明第二实施例的一种路由设备的操作方法的流程示意图;

[0098] 图3为本发明第三实施例的一种路由设备的操作方法的流程示意图;

[0099] 图4为本发明第四实施例的一种路由设备的操作方法的流程示意图;

[0100] 图5为本发明第一实施例的一种应用程序的API调用方法的流程示意图;

[0101] 图6为本发明第二实施例的一种应用程序的API调用方法的流程示意图;

[0102] 图7为本发明第一实施例的一种路由设备的结构示意图;

- [0103] 图8是本发明第二实施例的一种路由设备的结构示意图；
[0104] 图9是本发明第一实施例的一种终端设备的结构示意图；
[0105] 图10是本发明第二实施例的一种终端设备的结构示意图；
[0106] 图11是本发明第三实施例的一种路由设备的结构示意图；
[0107] 图12是本发明第三实施例的一种终端设备的结构示意图。

具体实施方式

[0108] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0109] 当本发明实施例提及“第一”、“第二”等序数词时,应当理解为仅仅起区分的作用,除非根据上下文其确实表达顺序之意。

[0110] 参见图1,为本发明第一实施例的一种路由设备的操作方法的流程示意图,在本实施例中,所述方法包括:

[0111] S101、第一系统获取第一API调用请求,第一API调用请求为运行在第一系统中的应用程序发送的。

[0112] 具体的,路由设备包括第一系统和第二系统,第一系统可以包括操作系统,还可以包括用于运行操作系统的一些硬件资源,第二系统可以包括操作系统,还可以包括用于运行操作系统的一些硬件资源。第一系统和第二系统可以共用路由设备的硬件资源,第一系统运行的操作系统和第二系统运行的操作系统的类型不相同,第一系统中运行的操作系统可以为通用的操作系统,使用广泛,第二系统中运行的操作系统为路由设备专用的操作系统,使用范围小。应用程序运行在第一系统中,当应用程序需要调用第二系统提供的API(Application Programming Interface,应用程序编程接口)时,应用程序向第一系统发送第一API调用请求,第一API调用请求用于调用第二系统中提供的路由能力API。第一API调用请求中可以包括多个请求参数,例如,请求参数可以包括:操作命令、API的接口版本号、API的类型的和API调用请求的格式等。

[0113] S102、第一系统将第一API调用请求转换成第二系统匹配的第二API调用请求,并将第二API调用请求发送至第二系统。

[0114] 具体的,第一系统和第二系统中运行的操作系统的类型不一致,不同的操作系统中API调用的语法规则也会不相同,路由设备包括的第一系统将第一API调用请求转换成可被第二系统运行的操作系统识别的第二API调用请求,并将第二API调用请求发送至第二系统运行的操作系统。

[0115] 例如,第一系统中运行Android操作系统,第二系统中运行Linux操作系统,可以得知,Android操作系统中的API调用请求和Linux操作系统中的API调用请求的语法规则是不相同的,第一系统需要将Android操作系统中运行的应用程序发出的第一API调用请求转换成第二系统中运行的Linux操作系统匹配的第二API调用请求。

[0116] 可以理解的是,第一系统识别应用程序发出的第一API调用请求中的参数是否符合要求,若是,在第一系统将第一API调用请求转换成第二系统匹配的第二API调用请求,若

否,则向应用程序返回错误提示信息,该错误提示信息用于表示第一API调用请求参数错误。

[0117] 例如,第一API调用请求中包括的接口版本号为v2,第一系统支持的接口版本号为v3,第一系统识别出第一API调用请求中的接口版本号不符合要求,则第一系统向应用程序返回接口版本号错误的错误提示信息。

[0118] 例如,第一API调用请求的格式为HTTP (HyperText Transfer Protocol,超文本传输协议) 1.0,第一系统支持的格式为HTTP1.1,第一系统识别出第一API调用请求的格式不符合要求,则第一系统向应用程序返回格式错误的错误提示信息。

[0119] S103、第二系统执行第二API调用请求。

[0120] 具体的,第二系统接收到符合与自身匹配的第二API调用请求,执行第二API调用响应,根据第二API调用请求生成相应的第二API调用响应,并将第二API调用响应发送至路由设备。

[0121] 实施本发明的实施例,通过将运行在第一系统中的应用程序发出的API调用请求转换成与第二系统匹配的调用请求,实现了第一系统中的应用程序的跨平台API调用,由于第一系统为通用的开发平台,开发应用程序难度小,开发者因此不需要直接在路由设备的专用的第二系统中开发应用程序,降低了应用程序的开发难度。

[0122] 参见图2,为本发明第二实施例的一种路由设备的操作方法的流程示意图,在本发明实施例中,所述方法包括:

[0123] S201、第一系统获取第一API调用请求,第一API调用请求为运行在第一系统中的应用程序发送的。

[0124] 具体的,路由设备可以包括第一系统和第二系统。第一系统中运行的操作系统为通用的操作系统,使用广泛,第二系统中运行的操作系统为路由设备专用的操作系统,使用范围小。可选的,第一系统中运行的操作系统可以包括:安卓(Android)、苹果移动设备操作系统(IOS)、Windows Phone和BlackBerry中的任意一种,第二系统中运行的操作系统可以包括Linux或eCOS(embedded Configurable operating System,嵌入式可配置操作系统)。

[0125] 需要说明的是,在本发明各实施例中,所述的第一系统和第二系统中运行的操作系统并不限于上述的内容,例如:第二系统中运行的操作系统也可以为Android、IOS等;只要保证第一系统和第二系统中运行的操作系统不同即可。

[0126] 应用程序运行在第一系统中,当应用程序需要调用第二系统提供的API时,应用程序向第一系统发送第一API调用请求,第一API调用请求用于调用第二系统中提供的路由能力API。第一API调用请求中可以包括多个请求参数,可选的,请求参数包括:操作命令、API的接口版本号、API的类型的和API调用请求的格式等。

[0127] 示例性的,第一系统获取应用程序发送的第一API调用请求,第一API调用请求的语法规则符合REST (Representational State Transfer,表征状态转移) 协议的规定。例如,第一API调用请求为:POST/v1/api/category/model HTTP1.1,第一API调用请求中各个请求参数的含义如下:

[0128] POST:表示操作命令;例如,POST为更改命令,GET为查询命令;

[0129] v1:表示应用程序支持的API的接口版本号;例如,如果应用程序只支持v1接口,则不能调用v2接口。

[0130] api:表示节点名,所有的API资源都包含在api这个节点内;

[0131] category:表示API的类型,例如,路由设备的隧道API或DNS (Domain Name System,域名系统) API等类型;

[0132] model:表示category下的子类型,例如,隧道API下的L2TP (Layer 2 Tunneling Protocol,第2层隧道协议) 隧道API;

[0133] HTTP 1.1:表示第一API调用请求的格式。

[0134] 可选的,在本发明的实施例中,第一API调用请求包括应用鉴权请求、隧道状态获取请求、隧道状态配置请求、DNS信息获取请求和DNS配置请求中的任意一种或几种,示例性的,应用程序发送的第一API调用请求的形式如下:

[0135] 1、应用鉴权请求:POST/v1/api/oauth2/token HTTP/1.1

[0136]

字段名	类型	长度	说明	M/O
AppId	string	64	App的唯一ID,需申请。	M
AppSecret	string	64	App对应的密码,需申请。	M

[0137] 2、PPTP (Point to Point Tunneling Protocol,点对点隧道协议) 隧道状态获取请求:GET/v1/api/tunnel/PPTP HTTP/1.1

[0138] 3、PPTP隧道配置请求:POST/v1/api/tunnel/PPTP HTTP/1.1

[0139] 4、L2TP隧道状态获取请求:GET/v1/api/tunnel/L2TP HTTP/1.1

[0140] 5、L2TP隧道配置请求:POST/v1/api/tunnel/L2TP HTTP/1.1

[0141] 6、DNS信息获取请求:GET/v1/api/DNS/server HTTP/1.1

[0142] 7、DNS信息配置请求:POST/v1/api/DNS/server HTTP/1.1

[0143] 8、DNS过滤信息获取请求:GET/v1/api/DNS/filter HTTP/1.1

[0144] 9、DNS过滤信息配置请求:POST/v1/api/DNS/filter HTTP/1.1

[0145]

字段名	类型	长度	说明	M/O
DomainName	string	32	待过滤的 DNS 域名	M
Operation	string	32	操作: add: 添加 delete: 删除	M

[0146] 以上为第一API调用请求的几种示例,具体实施时,第一API调用请求还包括IP地址配置请求、IP地址获取请求、IP路由协议配置请求和端口配置请求中的任意一种,可按照需要自定义符合REST协议的第一API调用请求。

[0147] 可以理解的是,在第一系统获取第一API调用请求之前,第一系统与第二系统之间已建立通信连接。例如,第一系统可通过UPnP (Universal Plug and Play,通用即插即用) 协议检测第二系统,在检测到第二系统后,可进行双向身份验证,在双向身份验证合法后,第一系统与第二系统建立通信连接,二者之间建立的通信连接可以是TCP (Transmission Control Protocol,传输控制协议) 连接,也可以是其他类型的通信连接,本发明不作限制。

[0148] S202、第一系统将第一API调用请求转换成第二系统匹配的第二API调用请求,并将第二API调用请求发送至第二系统。

[0149] 具体的,第一系统和第二系统中运行的操作系统的类型不一致,不同的操作系统中API调用的语法规则也会不相同,路由设备包括的第一系统将第一API调用请求进行转换成可被第二系统运行的操作系统识别的第二API调用请求,并将第二API调用请求发送至第二系统中的操作系统。

[0150] 例如,第一系统中运行的操作系统为Android操作系统,第二系统中运行的操作系统为Linux操作系统,可以得知,Android操作系统中的API调用请求和Linux操作系统中的API调用请求的语法规则是不相同的,第一系统需要将Android操作系统中运行的应用程序发出的第一API调用请求转换成第二系统中运行的Linux操作系统匹配的第二API调用请求。示例性的,应用程序发送的第一API调用请求,第一API调用请求为应用鉴权请求,应用鉴权请求的形式为:

[0151] POST/v1/api/oauth2/token HTTP/1.1

[0152] {

[0153] "AppId":"tunnel",

[0154] "AppSecret":"ZjBiZDI3MGU5Mzk3MWNkMWMzNWM5N2QyMzE2ZGY1YmM1MzhkYjQyZWI5ZD11YTgzMTA5NjNmOTQ5YzE1MTBhMw=="

[0155] }

[0156] 上述应用鉴权请求(即第一API调用请求)的语法规则是符合第一系统中运行的Android操作系统的,第一系统将该应用鉴权请求转换为第二系统中运行的Linux操作系统匹配的应用鉴权请求,即第二API调用请求,表示如下:

[0157] CmdType:oauth2

[0158] CmdCode:1

[0159] Parameter:id=tunnel,sec=ZjBiZDI3MGU5Mzk3MWNkMWMzNWM5N2QyMzE2ZGY1YmM1MzhkYjQyZWI5ZD11YTgzMTA5NjNmOTQ5YzE1MTBhMw==

[0160] 以上是以应用程序发送的第一API调用请求为应用鉴权请求为例进行说明,应用程序发送其他形式的API请求的方法是可参照应用鉴权请求的例子,本发明不再赘述。

[0161] 在本发明的实施例中,第一系统将第一API调用请求转换成第二系统匹配的第二API调用请求的方法可以是:路由设备包括的第一系统根据预设的第一映射关系查询第一API调用请求关联的第二API调用请求,第一映射关系中设置有与第一系统匹配的第一API调用请求和与第二系统匹配的第二API调用请求之间的对应关系,路由设备包括的第一系统获取到第一API调用请求后,通过查询第一映射关系就能得到的第二API调用请求。

[0162] 可选的,第一系统将第一API调用请求转换成第二系统匹配的第一API调用请求的方法还可以是:路由设备包括的第一系统提取第一API调用请求中的请求参数,根据请求参数生成与第二系统匹配的第二API调用请求。

[0163] 可以理解的是,第一系统识别应用程序发出的第一API调用请求中的参数是否符合要求,若是,第一系统将第一API调用请求转换成第二系统匹配的第二API调用请求,若否,向应用程序返回错误提示信息,该错误提示信息用于表示第一API调用请求参数错误。

[0164] 例如,第一API调用请求中包括的接口版本号为v2,第一系统支持的接口版本号为

v3,第一系统识别出第一API调用请求中的接口版本号不符合要求,则第一系统向应用程序返回接口版本号错误的错误提示信息。

[0165] 例如,第一API调用请求中包括的格式为HTTP (HyperText Transfer Protocol,超文本传输协议) 1.0,第一系统支持的格式为HTTP1.1,第一系统识别出第一API调用请求中的格式不符合要求,则第一系统向应用程序返回格式错误的错误提示信息。

[0166] S203、第二系统执行第二API调用请求。

[0167] 具体的,第二系统接收到符合与自身匹配的第二API调用请求,执行第二API调用请求所指示的操作,第二系统根据第二API调用请求生成相应的第二API调用响应,并将第二API调用响应发送至路由设备包括的第一系统。

[0168] S204、第二系统向第一系统返回第二API调用响应。

[0169] S205、第一系统将第二API调用响应转换成与第一系统匹配的第一API调用响应,并将第一API调用响应返回至应用程序。

[0170] 具体的,第一系统和第二系统中运行的操作系统的类型不一致,不同的操作系统中API调用的语法规则也会不相同,路由设备包括的第一系统将第二系统返回的第二API调用响应进行转换成与自身匹配的识别的第一API调用响应,并将第一API调用响应发送至第一系统中的应用程序。

[0171] 例如,第一系统中运行的操作系统为Android操作系统,第二系统中运行的操作系统为Linux操作系统,第二系统向第一系统返回的第二API调用响应为应用鉴权响应,表示如下:

[0172] ErrorCode:0

[0173] Parameter:token=M8ERGDUs1ulpJiojh0XufZd82a8Jh3n19ZTPCNfDgf1Xxpg0qbiD
DbtLbDRUSTfS2cLQ0w5qKkrIROWDe8VPhoys0qkSjxKjJt9vYdVIxkon9l0lrUbGgJ6g9q7qVPIEh

[0174] 第一系统将从第二系统接收到的应用鉴权响应转换成与自身匹配的应用鉴权响应,表示如下:

[0175] HTTP/1.1200 OK

[0176] {"AccessToken":"M8ERGDUs1ulpJiojh0XufZd82a8Jh3n19ZTPCNfDgf1Xxpg0qbiD
btLbDRUSTfS2cLQ0w5qKkrIROWDe8VPhoys0qkSjxKjJt9vYdVIxkon9l0lrUbGgJ6g9q7qVPI
Eh","RefreshToken":"","TokenType":"oauth"}

[0177] 路由设备将第一API调用响应发送至应用程序,应用程序呈现该第二API调用响应。

[0178] 在本发明的实施例中,第一系统将第二API调用响应转换成与自身匹配的第一API调用响应的方法可以是:路由设备包括的第一系统根据预设的第二映射关系查询第二API调用响应关联的第一API调用响应,第二映射关系中设置有与第一系统匹配的第一API调用响应和与第二系统匹配的第二API调用响应之间的对应关系,路由设备包括的第一系统获取到第二系统返回的第二API调用响应后,通过查询第一映射关系就能得到的第一API调用响应。

[0179] 可选的,第一系统将第二API调用响应转换成与自身匹配的第一API调用响应的方法还可以是:路由设备包括的第一系统提取第二API调用响应中的响应参数,根据响应参数生成与第一系统匹配的第一API调用请求。

[0180] 可以理解的是,第一系统在向第二系统发送第二API调用请求是,判断在预设的时长内是否接收到第二系统根据第二API调用请求返回的第二调用响应,若为否,向应用程序发送超时提示信息。

[0181] 例如,预设的时长设置为50ms,第一系统向第二系统发送生成的第二API调用请求时,启动定时器,若在50ms内未接收到第二API调用响应,向应用程序发送超时提示信息,超时提示信息的形式可以为错误信息代码,例如,超时提示信息为“Error 404”,具体的形式可以根据需要进行设置,本发明不作限制。应用程序的使用者可根据超时提示信息查询到出错原因。

[0182] 实施本发明的实施例,通过将运行在第一系统中的应用程序发出的API调用请求转换成与第二系统匹配的调用请求,实现了第一系统中的应用程序的跨平台API调用,由于第一系统为通用的开发平台,开发应用程序难度小,开发者因此不需要直接在路由设备的专用的第二系统中开发应用程序,降低了应用程序的开发难度。

[0183] 参见图3,为本发明第三实施例的一种路由设备的操作方法的流程示意图,在本发明实施例中,所述方法包括:

[0184] S301、终端设备获取第一应用程序编程接口API调用请求,第一API调用请求为运行在第一系统中的应用程序发送的。

[0185] 具体的,终端设备包括第一系统,路由设备包括第二系统。第一系统可以包括操作系统,还可以包括用于运行操作系统的一些硬件资源,第二系统可以包括操作系统,还可以包括用于运行操作系统的一些硬件资源。终端设备和路由设备的硬件资源是独立的,终端设备运行的操作系统和路由设备运行的操作系统的类型是不相同,终端设备中运行的操作系统可以为通用的操作系统,使用范围广泛,路由设备中运行的操作系统为专用的操作系统,使用范围小。应用程序运行在终端设备的第一系统中,当应用程序需要调用路由设备的第二系统提供的API时,应用程序向终端设备的第一系统发送第一API调用请求,第一API调用请求用于调用路由设备的第二系统中提供的路由能力API。第一API调用请求中可以包括多个请求参数,例如,请求参数包括:操作命令、API的接口版本号、API的类型的和API调用请求的格式等。

[0186] S302、终端设备将第一API调用请求转换成第二系统匹配的第二API调用请求,并将第二API调用请求发送至路由设备,以使路由设备执行第二API调用请求。

[0187] 具体的,终端设备的第一系统和路由设备的第二系统中运行的操作系统的类型不一致,不同的操作系统中API调用的语法规则也会不相同,终端设备包括的第一系统将第一API调用请求转换成可被路由设备包括的第二系统运行的操作系统识别的第二API调用请求,并将第二API调用请求发送至路由设备包括的第二系统运行的操作系统,路由设备接收到第二API调用请求后,执行第二API调用请求指示的操作,并生成相应的第二API调用响应。

[0188] 例如,终端设备包括的第一系统中运行Android操作系统,路由设备中包括的第二系统中运行Linux操作系统,可以得知,Android操作系统中的API调用请求和Linux操作系统中的API调用请求的语法规则是不相同的,终端设备包括的第一系统需要将Android操作系统中运行的应用程序发出的第一API调用请求转换成路由设备包括的第二系统中运行的Linux操作系统匹配的第二API调用请求。

[0189] 可以理解的是,终端设备识别应用程序发出的第一API调用请求中的参数是否符合要求,若是,终端设备将第一API调用请求转换成路由设备包括的第二系统匹配的第二API调用请求,若否,向应用程序返回错误提示信息,该错误提示信息用于表示第一API调用请求参数错误。

[0190] 例如,第一API调用请求中包括的接口版本号为v2,终端设备包括的第一系统支持的接口版本号为v3,终端设备识别出第一API调用请求中的接口版本号不符合要求,则终端设备向应用程序返回接口版本号错误的错误提示信息。

[0191] 例如,第一API调用请求中包括的格式为HTTP (HyperText Transfer Protocol,超文本传输协议) 1.0,第一系统支持的格式为HTTP1.1,终端设备识别出第一API调用请求中的格式不符合要求,则终端设备向第一系统中运行的应用程序返回格式错误的错误提示信息。

[0192] 实施本发明的实施例,通过将运行在第一系统中的应用程序发出的API调用请求转换成与第二系统匹配的调用请求,实现了第一系统中的应用程序的跨平台API调用,由于第一系统为通用的开发平台,开发应用程序难度小,开发者因此不需要直接在路由设备的专用的第二系统中开发应用程序,降低了应用程序的开发难度。

[0193] 参见图4,为本发明第四实施例的一种路由设备的操作方法的流程示意图,在本实施例中,所述方法包括:

[0194] S401、终端设备获取第一应用程序编程接口API调用请求,第一API调用请求为运行在第一系统中的应用程序发送的。

[0195] 具体的,终端设备包括第一系统,路由设备包括第二系统。终端设备和路由设备的硬件资源是独立的,终端设备运行的操作系统和路由设备运行的操作系统的类型是不相同,终端设备中运行的操作系统为通用的操作系统,使用范围广泛,路由设备中运行的操作系统为专用的操作系统,使用范围小,可选的,终端设备包括的第一系统中运行的操作系统包括:Android、IOS、Windows Phone和BlackBerry中的任意一种,路由设备包括的第二系统中运行的操作系统包括Linux或eCOS。在本发明实施例中,终端设备可以是台式电脑、笔记本电脑、平板电脑、智能手机、智能游戏机、智能影音娱乐设备或个人数字助理等终端,本发明不作限制。

[0196] 应用程序运行在终端设备包括的第一系统中,当应用程序需要调用路由设备包括的第二系统提供的API时,应用程序向终端设备包括的第一系统发送第一API调用请求,第一API调用请求用于调用路由设备包括的第二系统中提供的路由能力API。第一API调用请求中可以包括多个请求参数,可选的,请求参数包括:操作命令、API的接口版本号、API的类型的和API调用请求的格式等。

[0197] 可以理解的是,在终端设备获取第一API调用请求之前,终端设备与路由设备之间已建立通信连接。例如。终端设备可通过路由设备广播的信标Beacon信号检测到路由设备的存在,终端设备在检测到路由设备后,可进行双向身份验证,在双向身份验证合法后,终端设备与路由设备建立通信连接,二者之间建立的通信连接可以是TCP (Transmission Control Protocol,传输控制协议) 连接,也可以是其他类型的通信连接,本发明不作限制。

[0198] S402、终端设备将第一API调用请求转换成第二系统匹配的第二API调用请求,并将第二API调用请求发送至路由设备,以使路由设备执行第二API调用请求。

[0199] 具体的,终端设备包括的第一系统和路由设备包括的第二系统中运行的操作系统的类型不一致,不同的操作系统中API调用的语法规则也会不相同,终端设备将第一API调用请求进行转换成可被路由设备包括的第二系统运行的操作系统识别的第二API调用请求,并将第二API调用请求发送至第二系统中的操作系统。路由设备接收到符合与自身匹配的第二API调用请求,执行第二API调用请求所指示的操作,路由设备根据第二API调用请求生成相应的第二API调用响应,并将第二API调用响应发送至终端设备设备包括的第一系统。

[0200] 例如,终端设备包括的第一系统中运行的操作系统为Android操作系统,路由设备包括的第二系统中运行的操作系统为Linux操作系统,可以得知,Android操作系统中的API调用请求和Linux操作系统中的API调用请求的语法规则是不相同的,终端设备需要将Android操作系统中运行的应用程序发出的第一API调用请求转换成路由设备包括的第二系统中运行的Linux操作系统匹配的第二API调用请求。示例性的,应用程序发送的第一API调用请求,第一API调用请求为应用鉴权请求;第二API调用请求为Linux操作系统匹配的应用鉴权请求;具体的应用鉴权请求的形式可以参见图2所示实施例步骤202中的描述,在此不再赘述。应用程序发送其他形式的API请求的方法是可参照应用鉴权请求的例子,本发明不再赘述。

[0201] 在本发明的实施例中,终端设备将第一API调用请求转换成第二系统匹配的第二API调用请求的方法可以是:终端设备根据预设的第一映射关系查询第一API调用请求关联的第二API调用请求,第一映射关系中设置有与第一系统匹配的第一API调用请求和与第二系统匹配的第二API调用请求之间的对应关系,终端设备获取到第一API调用请求后,通过查询第一映射关系就能得到的第二API调用请求。

[0202] 可选的,终端设备将第一API调用请求转换成第二系统匹配的第一API调用请求的方法还可以是:终端设备提取第一API调用请求中的请求参数,根据请求参数生成与第二系统匹配的第二API调用请求。

[0203] 可以理解的是,终端设备识别应用程序发出的第一API调用请求中的参数是否符合要求,若是,终端设备将第一API调用请求转换成第二系统匹配的第二API调用请求,若否,向应用程序返回错误提示信息,该错误提示信息用于表示第一API调用请求参数错误。

[0204] S403、终端设备接收第二系统返回的第二API调用响应。

[0205] S404、终端设备将第二API调用响应转换成第一系统匹配的第一API调用响应,并将第一API调用响应返回至应用程序。

[0206] 具体的,终端设备包括的第一系统和路由设备包括的第二系统中运行的操作系统的类型不一致,不同的操作系统中API调用的语法规则也会不相同,终端设备将第二系统返回的第二API调用响应进行转换成与自身匹配的识别的第一API调用响应,并将第一API调用响应发送至第一系统中的应用程序。

[0207] 例如,终端设备包括的第一系统中运行的操作系统为Android操作系统,路由设备包括的第二系统中运行的操作系统为Linux操作系统,路由设备向第一系统返回的第二API调用响应为应用鉴权响应,表示如下:

[0208] ErrorCode:0

[0209] Parameter:token=M8ERGDUs1u1pJiojh0XufZd82a8Jh3n19ZTPCNfDgf1Xxpg0qbi

DbtLbDRUSTfS2cLQ0w5qKkrIROWDe8VPhoys0qkSjxKjJt9vYdVIxkon9l0lrUbGgJ6g9q7qVPIEh
[0210] 终端设备将从第二系统接收到的应用鉴权响应转换成与自身匹配的应用鉴权响应,表示如下:

[0211] HTTP/1.1200 OK

[0212] {"AccessToken":"M8ERGDUs1ulpJiojh0XufZd82a8Jh3n19ZTPCNfDgflXxpg0qbiDbtLbDRUSTfS2cLQ0w5qKkrIROWDe8VPhoys0qkSjxKjJt9vYdVIxkon9l0lrUbGgJ6g9q7qVPIEh","RefreshToken":"","TokenType":"oauth"}

[0213] 终端设备将转换后得到第一API调用响应发送至应用程序,应用程序呈现该第二API调用响应。

[0214] 在本发明的实施例中,终端设备将第二API调用响应转换成与自身匹配的第一API调用响应的方法可以是:终端设备根据预设的第二映射关系查询第二API调用响应关联的第一API调用响应,第二映射关系中设置有与第一系统匹配的第一API调用响应和与第二系统匹配的第二API调用响应之间的对应关系,终端设备获取到第二系统返回的第二API调用响应后,通过查询第一映射关系就能得到的第一API调用响应。

[0215] 可选的,终端设备将第二API调用响应转换成与自身匹配的第一API调用响应的方法还可以是:终端设备提取第二API调用响应中的响应参数,根据响应参数生成与第一系统匹配的第一API调用请求。

[0216] 可以理解的是,终端设备在向第二系统发送第二API调用请求时,判断在预设的时长内是否接收到第二系统根据第二API调用请求返回的第二调用响应,若为否,向应用程序发送超时提示信息。

[0217] 实施本发明的实施例,通过将运行在第一系统中的应用程序发出的API调用请求转换成与第二系统匹配的调用请求,实现了第一系统中的应用程序的跨平台API调用,由于第一系统为通用的开发平台,开发应用程序难度小,开发者因此不需要直接在路由设备的专用的第二系统中开发应用程序,降低了应用程序的开发难度。

[0218] 参见图5和图6,为本发明实施例的一种应用程序的API调用方法的流程示意图,本发明实施例适用于两种场景,第一种场景:路由设备包括第一系统和第二系统,第一系统和第二系统共用路由设备的硬件资源,第一系统运行的操作系统和第二系统运行的操作系统的类型不相同;第二种场景:终端设备包括第一系统,路由设备包括第二系统,终端设备和路由设备的硬件资源是独立的,终端设备运行的操作系统和路由设备运行的操作系统的类型是不相同。应用程序为隧道应用程序Tunnel APP,Tunnel APP调用第一系统中的API 1和API 2实现PPTP隧道建立功能,具体流程如下:

[0219] 1、使用UPnP协议发现设备。第一系统使用UPnP协议发现第二系统;

[0220] 2、建立通信连接。第一系统和第二系统之间建立通信连接,例如TCP连接;

[0221] 3、第一系统向第二系统发送应用鉴权请求;

[0222] 具体的,Tunnel APP调用第一系统中的鉴权API时向第一系统发送API调用请求,第一系统将该API调用请求转换为与第二系统匹配的应用鉴权请求,第二系统接收到应用鉴权请求后,执行鉴权处理;

[0223] 4、第二系统向第一系统返回应用鉴权响应;第一系统将应用鉴权响应转换成与自身匹配的API调用响应;

[0224] 5、第一系统向第二系统发送PPTP隧道状态获取请求;Tunnel APP调用第一系统中的隧道API 1时向第一系统发送API调用请求,第一系统将该API调用请求转换成与第二系统匹配的PPTP隧道状态获取请求,第二系统根据PPTP隧道状态获取请求获取路由设备的PPTP隧道状态信息,并生成携带PPTP隧道状态信息的响应结果返回至第一系统,第一系统将该响应转换成与自身的匹配的响应结果,并将该响应结果发送至Tunnel APP进行呈现;

[0225] 6、第一系统向第二系统发送PPTP隧道配置请求;Tunnel APP调用第一系统中的隧道API 2配置路由设备的PPTP隧道,该过程与步骤5相同;

[0226] 7、第一系统向第二系统发送PPTP隧道状态获取请求;Tunnel APP再次调用第一系统中的隧道API 1,以查看路由设备上PPTP隧道配置是否成功。

[0227] 参见图6,在本发明实施例中,应用程序为DNS APP,DNS APP调用第一系统中的API 3和API 4实现DNS过滤功能,具体流程如下:

[0228] 1、使用UPnP协议发现设备;

[0229] 2、第一系统和第二系统间建立通信连接;

[0230] 3、应用鉴权请求;

[0231] 4、应用鉴权响应;

[0232] 5、DNS过滤信息获取请求;DNS APP调用第一系统中的隧道API 3时向第一系统发送API调用请求,第一系统将该API调用请求转换成与第二系统匹配的DNS过滤信息获取请求,第二系统根据DNS过滤信息获取请求获取路由设备的DNS过滤信息,并生成携带PPTP隧道过滤信息的响应结果返回至第一系统,第一系统将该响应转换成与自身的匹配的响应结果,并将该响应结果发送至DNS APP进行呈现;

[0233] 6、DNS过滤信息配置请求;第一系统调用隧道API 4配置路由设备的DNS过滤信息,此过程和步骤5相同。

[0234] 参见图7,为本发明实施例的一种路由设备的结构示意图,在本发明实施例中,所述路由设备包括第一系统和第二系统,第一系统中包括第一请求获取模块10和第一请求转换模块11,第二系统中包括请求执行模块12。

[0235] 第一请求获取模块10,用于获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在所述第一系统中的应用程序发送的。

[0236] 具体的,路由设备包括第一系统和第二系统。应用程序运行在第一系统中,当应用程序需要调用第二系统提供的API时,应用程序向第一请求获取模块10发送第一API调用请求,第一API调用请求用于调用第二系统中提供的路由能力API。第一API调用请求中可以包括多个请求参数,例如,请求参数包括:操作命令、API的接口版本号、API的类型的和API调用请求的格式等。

[0237] 第一请求转换模块11,用于将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述第二系统。

[0238] 具体的,第一系统和第二系统中运行的操作系统的类型不一致,不同的操作系统中API调用的语法规则也会不相同,第一请求转换模块11将第一API调用请求转换成可被第二系统运行的操作系统识别的第二API调用请求,并将第二API调用请求发送至第二系统运行的操作系统。

[0239] 请求执行模块12,用于执行所述第二API调用请求。

[0240] 具体的,请求执行模块12接收到符合与第二系统匹配的第二API调用请求,执行第二API调用响应,根据第二API调用请求生成相应的第二API调用响应,并将第二API调用响应发送至第一系统。

[0241] 实施本发明的实施例,通过将运行在第一系统中的应用程序发出的API调用请求转换成与第二系统匹配的调用请求,实现了第一系统中的应用程序的跨平台API调用,由于第一系统为通用的开发平台,开发应用程序难度小,开发者因此不需要直接在路由设备的专用的第二系统中开发应用程序,降低了应用程序的开发难度。

[0242] 参见图8,为本发明第二实施例的一种路由设备的结构示意图,在本发明实施例中,所述路由设备包括第一系统和第二系统,其中,所述第一系统除包括第一请求获取模块10和第一请求转换模块11之外,还包括第一响应转换模块14,所述第二系统除包括请求执行模块12之外,还包括响应返回模块13。

[0243] 响应返回模块12,用于在所述请求执行模块执行所述第二API调用请求之后,向所述第一系统返回第二API调用响应。

[0244] 第一响应转换模块13,用于将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应,并将所述第一API调用响应返回至所述应用程序。

[0245] 可选的,第一请求转换模块11在将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求时,具体用于:

[0246] 根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求;或

[0247] 提取所述第一API调用请求中的请求参数,根据所述请求参数生成与所述第二系统匹配的第二API调用请求。

[0248] 可选的,第一响应转换模块13在将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应时,具体用于:

[0249] 根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应;或

[0250] 提取所述第二API调用响应中的响应参数,根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

[0251] 可选的,第一API调用请求包括应用鉴权请求、隧道状态获取请求、隧道状态配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种或几种。

[0252] 可选的,第一系统中运行的操作系统包括Android、IOS、Windows Phone和BlackBerry中的任意一种,所述第二系统中运行的操作系统包括Linux或eCOS。

[0253] 图7和图8所示实施例与图1和图2所示的方法实施例属于同一构思,其带来的技术效果也相同,具体可以参照上述方法实施例的描述,此处不再赘述。

[0254] 参见图9,为本发明第二实施例的一种终端设备的结构示意图,在本发明实施例中,所述终端设备包括第一系统,所述第一系统包括第二请求获取模块20和第二请求转换模块21。

[0255] 第二请求获取模块20,用于获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在所述第一系统中的应用程序发送的。

[0256] 具体的,终端设备包括第一系统,路由设备包括第二系统。应用程序运行在终端设

备的第一系统中,当应用程序需要调用路由设备的第二系统提供的API时,第二请求获取模块20获取应用程序发送第一API调用请求,第一API调用请求用于调用路由设备的第二系统中提供的路由能力API。第一API调用请求中可以包括多个请求参数,例如,请求参数包括:操作命令、API的接口版本号、API的类型的和API调用请求的格式等。

[0257] 第二请求转换模块21,用于将所述第一API调用请求转换成与包含在路由设备中的第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述路由设备,以使所述路由设备执行所述第二API调用请求。

[0258] 具体的,终端设备的第一系统和路由设备的第二系统中运行的操作系统的类型不一致,不同的操作系统中API调用的语法规则也会不相同,第二请求转换模块21将第一API调用请求转换成可被路由设备包括的第二系统运行的操作系统识别的第二API调用请求,并将第二API调用请求发送至路由设备包括的第二系统运行的操作系统,路由设备接收到第二API调用请求后,执行第二API调用请求指示的操作,并生成相应的第二API调用响应。

[0259] 实施本发明的实施例,通过将运行在第一系统中的应用程序发出的API调用请求转换成与第二系统匹配的调用请求,实现了第一系统中的应用程序的跨平台API调用,由于第一系统为通用的开发平台,开发应用程序难度小,开发者因此不需要直接在路由设备的专用的第二系统中开发应用程序,降低了应用程序的开发难度。

[0260] 参见图10,为本发明第二实施例的一种终端设备的结构示意图,在本发明实施例中,所述终端设备除包括第二请求获取模块20和第二请求转换模块21之外,还包括响应接收模块22和第二响应转换模块23。

[0261] 响应接收模块22,用于在所述第二请求转换模块将所述第二API调用请求发送至所述路由设备之后,接收所述第二系统返回的第二API调用响应。

[0262] 第二响应转换模块23,用于将所述第二API调用响应转换成所述第一系统匹配的第一API调用响应,并将所述第一API调用响应返回至所述应用程序。

[0263] 可选的,第二请求转换模块21在将所述第一API调用请求转换成与包含在路由设备中的第二系统匹配的第二API调用请求时,具体用于:

[0264] 根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求;或

[0265] 提取所述第一API调用请求中的请求参数,根据所述请求参数生成与所述第二系统匹配的第二API调用请求。

[0266] 第二响应转换模块23在将所述第二API调用响应转换成所述第一系统匹配的第一API调用响应时,具体用于:

[0267] 根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应;或

[0268] 提取所述第二API调用响应中的响应参数,根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

[0269] 第一API调用请求包括应用鉴权请求、隧道状态获取请求、隧道状态配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种。

[0270] 第一系统中运行的操作系统包括Android、IOS、Windows Phone和BlackBerry中的任意一种,所述第二系统中运行的操作系统包括Linux或eCOS。

[0271] 图9、图10所示实施例与图3、图4所示的方法实施例属于同一构思,其带来的技术效果也相同,具体请参照上述方法实施例的描述,此处不再赘述。

[0272] 参见图11,为本发明第三实施例的一种路由设备的结构示意图,在本发明实施例中,所述路由设备包括处理器61和存储器62,路由设备中的处理器61的数量可以是一个或多个,图11以一个处理器为例。本发明的一些实施例中,处理器61和存储器62可通过总线或其他方式连接,图11中以总线连接为例。

[0273] 其中,存储器62中存储一组程序代码,且处理器61用于调用存储器62中存储的程序代码,用于执行以下操作:

[0274] 获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在第一系统中的应用程序发送的;

[0275] 将所述第一API调用请求转换成与第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述第二系统;

[0276] 执行所述第二API调用请求;所述第一系统和所述第二系统运行在所述路由设备中。

[0277] 在本发明的一些实施例中,在执行所述第二API调用请求之后,处理器61还用于执行:

[0278] 向所述第一系统返回第二API调用响应;

[0279] 将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应,并将所述第一API调用响应返回至所述应用程序。

[0280] 在本发明的一些实施例中,处理器61执行所述将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求,包括:

[0281] 根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求;或

[0282] 提取所述第一API调用请求中的请求参数,根据所述请求参数生成与所述第二系统匹配的第二API调用请求。

[0283] 在本发明的一些实施例中,处理器61执行所述将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应,包括:

[0284] 根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应;或

[0285] 提取所述第二API调用响应中的响应参数,根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

[0286] 在本发明的一些实施例中,所述第一API调用请求包括应用鉴权请求、隧道状态获取请求、隧道配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种或几种。

[0287] 在本发明的一些实施例中,所述第一系统中运行的操作系统包括Android、IOS、Windows Phone和BlackBerry中的任意一种,所述第二系统中运行的操作系统包括Linux或eCOS。

[0288] 实施本发明的实施例,通过将运行在第一操作系统中的应用程序发出的API调用请求转换成与第二操作系统匹配的调用请求,以及将第二操作系统返回的调用响应转换成

与第一操作系统匹配的调用响应,实现了应用程序的跨平台API调用,使第一操作系统中的应用程序可以使用第二操作系统的资源,兼容性好,降低了应用程序的开发难度。

[0289] 图11所示实施例与图1、图2所示的方法实施例属于同一构思,其带来的技术效果也相同,具体请参照上述方法实施例的描述,此处不再赘述。

[0290] 参见图12,为本发明第三实施例的一种终端设备的结构示意图,在本发明实施例中,所述终端设备包括处理器71和存储器72,终端设备中的处理器71的数量可以是一个或多个,图12以一个处理器为例。本发明的一些实施例中,处理器71和存储器72可通过总线或其他方式连接,图12中以总线连接为例。

[0291] 其中,存储器72中存储一组程序代码,且处理器71用于调用存储器72中存储的程序代码,用于执行以下操作:

[0292] 获取第一应用程序编程接口API调用请求,所述第一API调用请求为运行在第一系统中的应用程序发送的;所述第一系统运行在所述终端设备中;

[0293] 将所述第一API调用请求转换成与第二系统匹配的第二API调用请求,并将所述第二API调用请求发送至所述路由设备,以使路由设备执行所述第二API调用请求。

[0294] 在本发明的一些实施例中,在将所述第二API调用请求发送至所述路由设备之后,处理器71还用于执行:

[0295] 接收所述第二系统返回的第二API调用响应;

[0296] 将所述第二API调用响应转换成所述第一系统匹配的第一API调用响应,并将所述第一API调用响应返回至所述应用程序。

[0297] 在本发明的一些实施例中,处理器71执行所述将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求,包括:

[0298] 根据预设的第一映射关系将所述第一API调用请求转换成与所述第二系统匹配的第二API调用请求;或

[0299] 提取所述第一API调用请求中的请求参数,根据所述请求参数生成与所述第二系统匹配的第二API调用请求。

[0300] 在本发明的一些实施例中,处理器71执行所述将所述第二API调用响应转换成所述第一系统匹配的第一API调用响应,包括:

[0301] 根据预设的第二映射关系将所述第二API调用响应转换成与所述第一系统匹配的第一API调用响应;或

[0302] 提取所述第二API调用响应中的响应参数,根据所述响应参数生成与所述第一系统匹配的所述第一API调用响应。

[0303] 在本发明的一些实施例中,所述第一API调用请求包括应用鉴权请求、IP地址配置请求、IP地址获取请求、IP路由协议配置请求、端口配置请求、隧道状态获取请求、隧道配置请求、域名系统DNS信息获取请求和DNS信息配置请求中的任意一种或几种。

[0304] 在本发明的一些实施例中,所述第一系统中运行的操作系统包括Android、IOS、Windows Phone和BlackBerry中的任意一种,所述第二系统中运行的操作系统包括Linux或eCOS。

[0305] 图12所示实施例与图3、图4所示的方法实施例属于同一构思,其带来的技术效果也相同,具体请参照上述方法实施例的描述,此处不再赘述。

[0306] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的程序可存储于一计算机可读取存储介质中,该程序在执行时,可包括如上述各方法的实施例的流程。其中,所述的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory,ROM)或随机存储记忆体(Random Access Memory,RAM)等。

[0307] 以上所揭露的仅为本发明一种较佳实施例而已,当然不能以此来限定本发明之权利范围,本领域普通技术人员可以理解实现上述实施例的全部或部分流程,并依本发明权利要求所作的等同变化,仍属于发明所涵盖的范围。

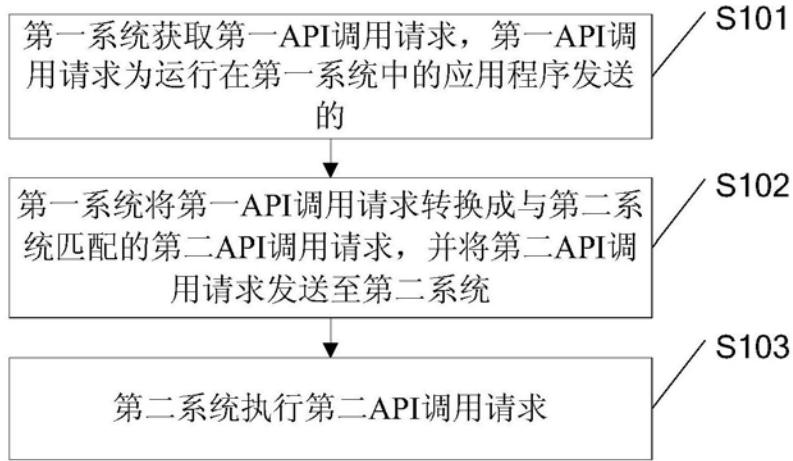


图1

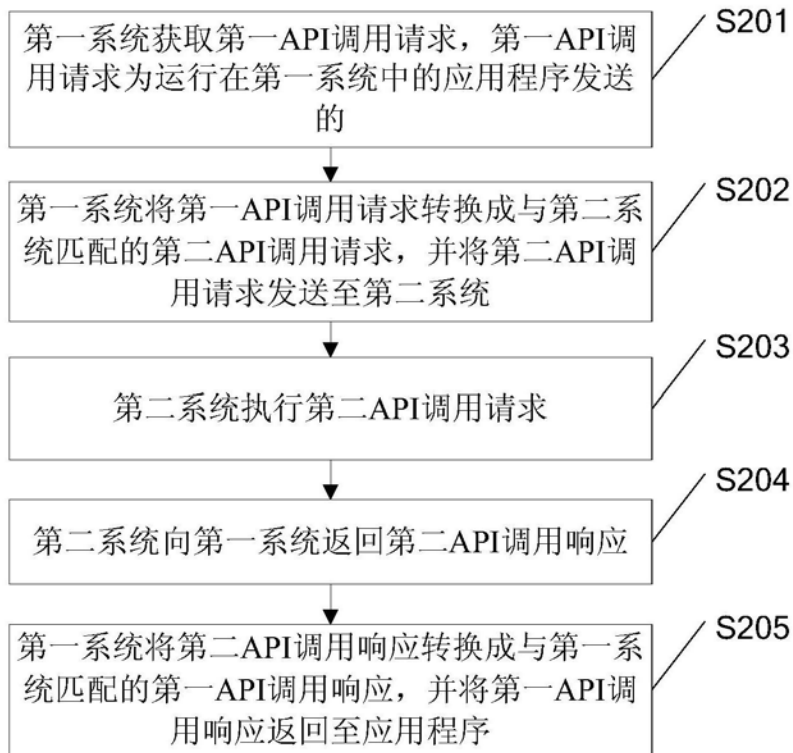


图2

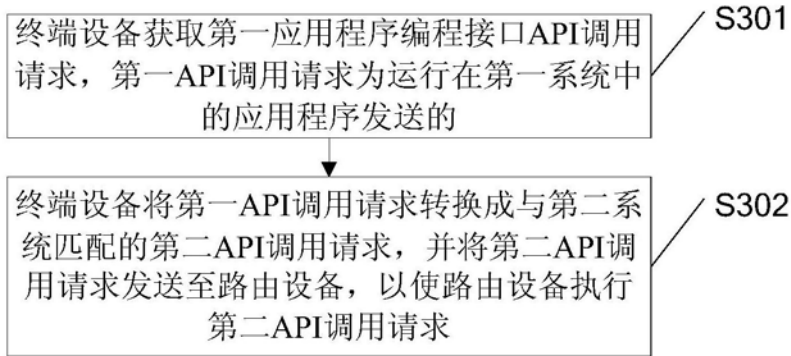


图3

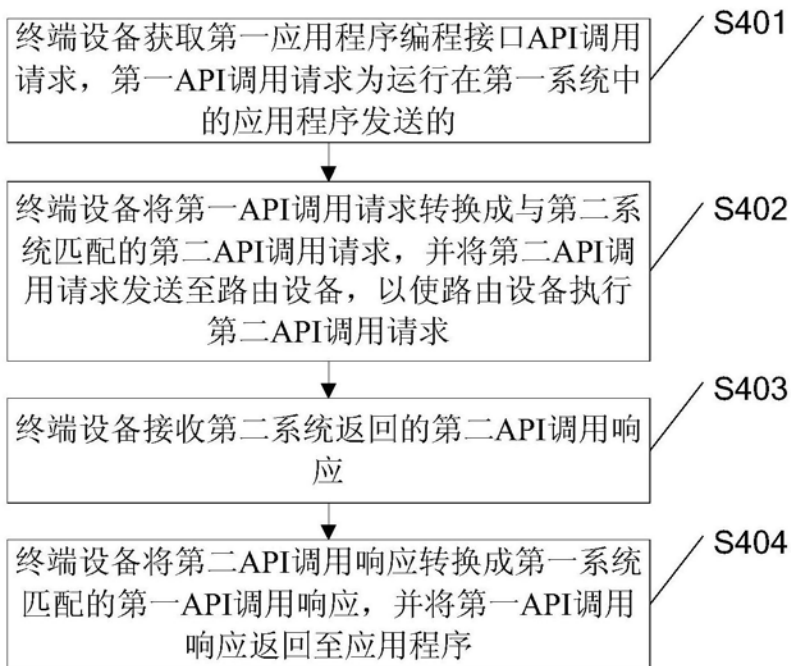


图4

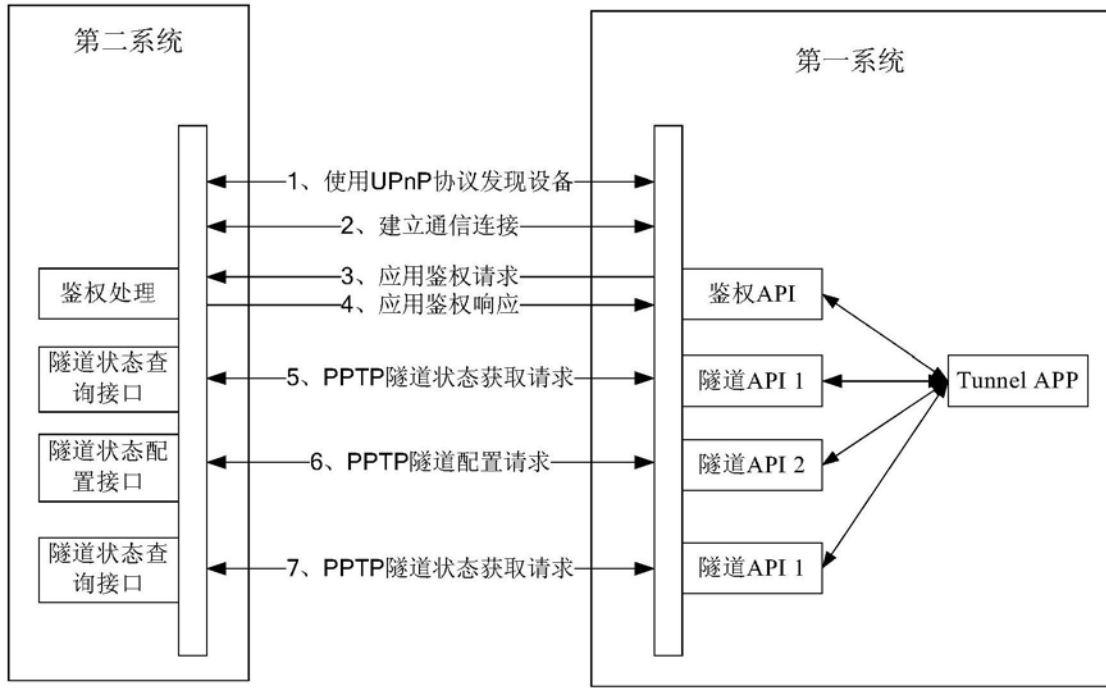


图5

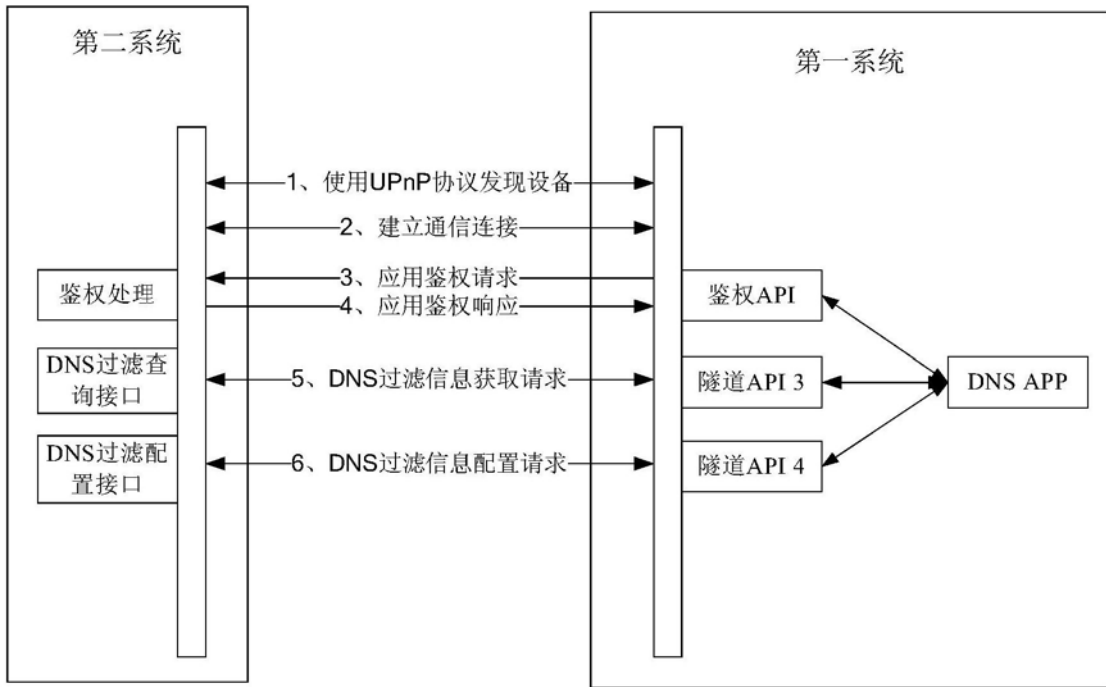


图6

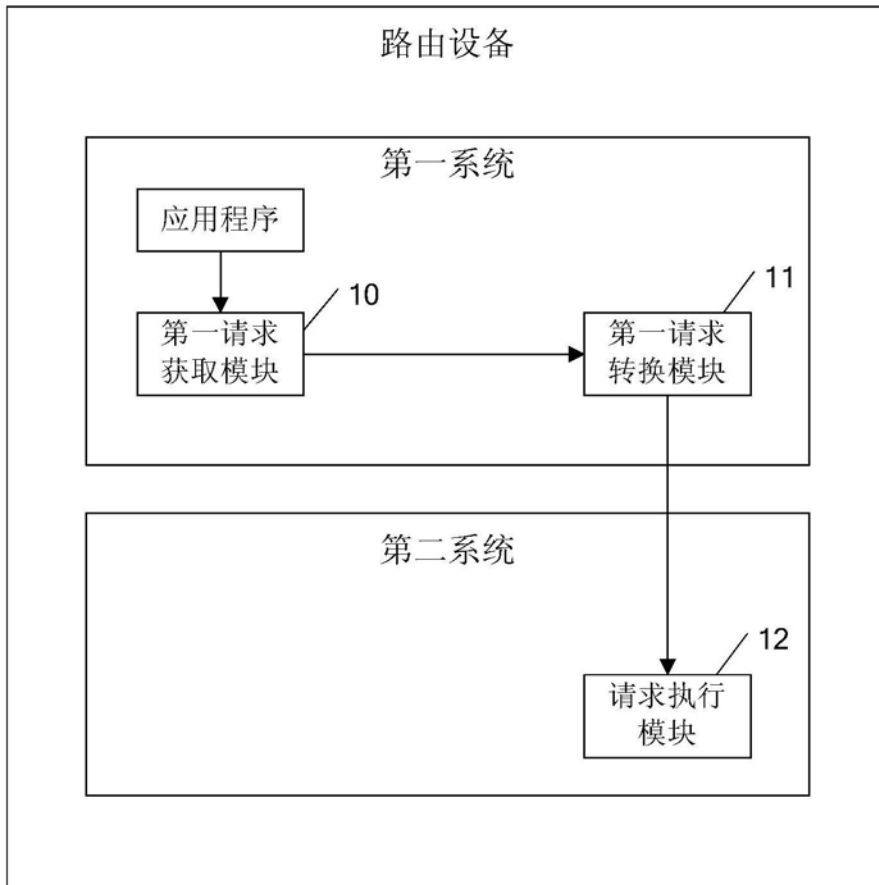


图7

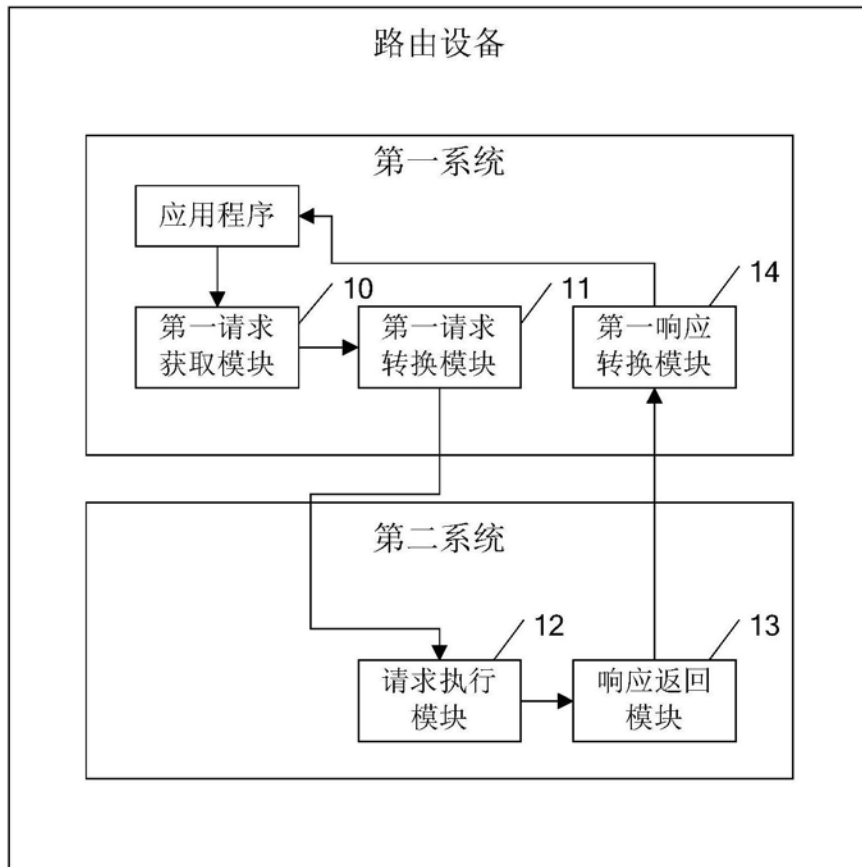


图8

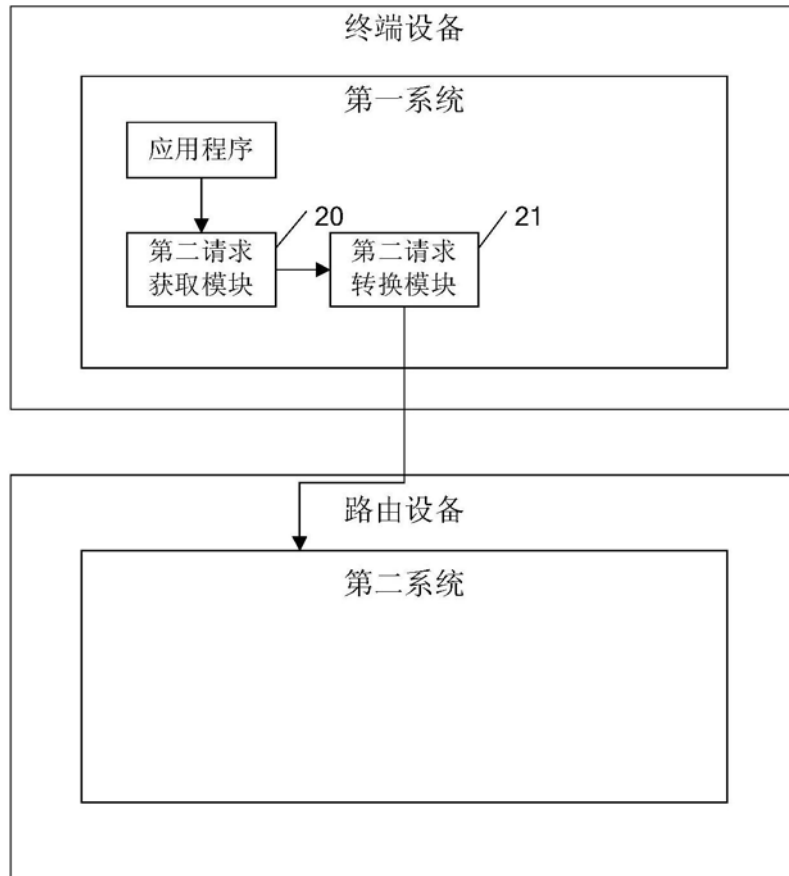


图9

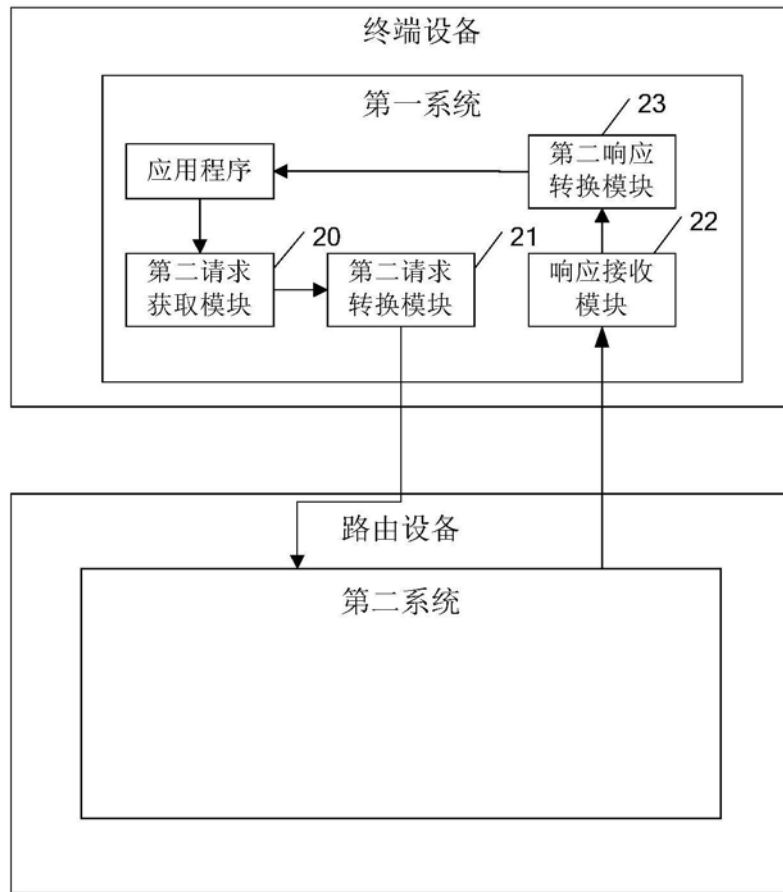


图10

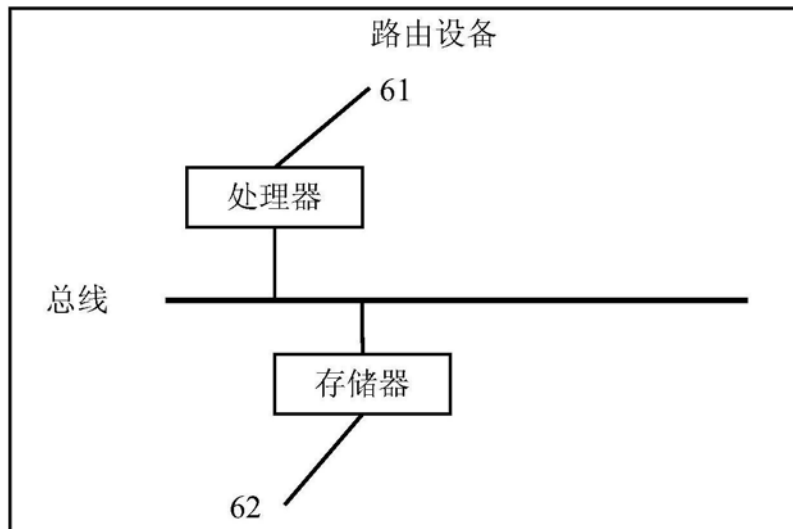


图11

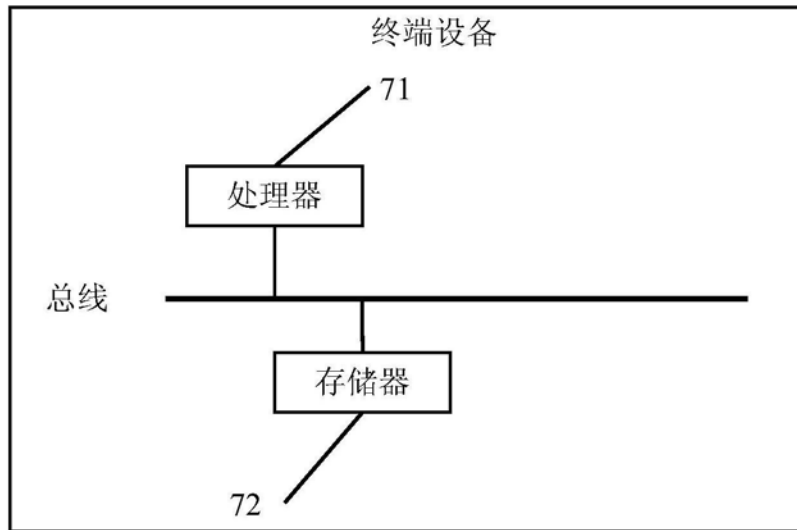


图12