



(12) 发明专利

(10) 授权公告号 CN 110351107 B

(45) 授权公告日 2021.04.09

(21) 申请号 201810292295.2

(22) 申请日 2018.04.03

(65) 同一申请的已公布的文献号
申请公布号 CN 110351107 A

(43) 申请公布日 2019.10.18

(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

(72) 发明人 邢超

(74) 专利代理机构 北京中博世达专利商标代理有限公司 11274

代理人 申健

(51) Int.Cl.
H04L 12/24 (2006.01)

(56) 对比文件

CN 107302443 A, 2017.10.27

EP 2733982 A1, 2014.05.21

US 2015026189 A1, 2015.01.22

US 2014006411 A1, 2014.01.02

审查员 唐文森

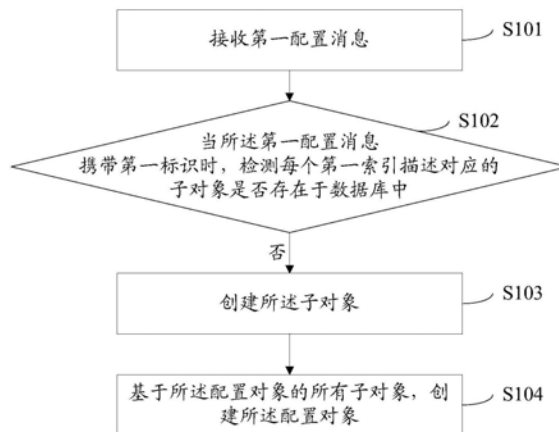
权利要求书3页 说明书13页 附图5页

(54) 发明名称

配置管理方法及装置

(57) 摘要

一种配置管理方法,涉及网络通信技术领域,用于解决NETCONF客户端与NETCONF服务器之间需要多次报文交互才能成功创建配置对象的问题。该方法包括:NETCONF服务器接收第一配置消息,该第一配置消息用于创建配置对象,该第一配置消息包括至少一个第一索引描述,前述至少一个第一索引描述用于指示所述配置对象的所有子对象,每个第一索引描述分别与所述配置对象的一个子对象对应;当第一配置消息携带第一标识时,对于每个第一索引描述,检测所述第一索引描述对应的子对象是否存在于数据库中;若子对象不存在于所述数据库中,创建该子对象;基于配置对象的所有子对象,创建配置对象。本申请适用于创建配置对象的过程中。



1. 一种配置管理方法,其特征在于,所述方法包括:

接收第一配置消息,所述第一配置消息用于创建配置对象,其中,所述配置对象包括多个子对象,所述第一配置消息包括至少一个第一索引描述,所述至少一个第一索引描述用于指示所述配置对象的所有子对象,每个第一索引描述分别与所述配置对象的一个子对象对应;所述配置对象为虚拟系统或者策略路由;

当所述第一配置消息携带第一标识时,对于每个第一索引描述,检测所述第一索引描述对应的子对象是否存在于数据库中,若所述子对象不存在于所述数据库中,创建所述子对象,所述第一标识用于指示自动处理所述配置对象;

基于所述配置对象的所有子对象,创建所述配置对象。

2. 根据权利要求1所述的方法,其特征在于,所述第一索引描述包括索引值;

所述检测所述第一索引描述对应的子对象是否存在于数据库中,包括:

检测所述索引值是否存在于所述数据库中;

所述若所述子对象不存在于数据库中,创建所述子对象,包括:

若所述索引值不存在于所述数据库中,则根据所述索引值,创建所述子对象。

3. 根据权利要求1或2所述的方法,其特征在于,所述子对象包括属性,所述第一索引描述还包括所述属性对应的属性数据;

所述创建所述子对象之前,所述方法还包括:

根据所述属性数据,确定所述子对象的属性的取值。

4. 根据权利要求1或2所述的方法,其特征在于,所述子对象包括属性,所述第一索引描述还包括第二标识,所述第二标识用于确定策略,所述策略用于确定所述子对象的属性的取值;

所述创建所述子对象之前,所述方法还包括:

根据所述第二标识对应的策略,确定所述子对象的属性的取值;

其中,所述策略包括以下至少一种:

以预设值作为所述子对象的属性的取值;

以随机值作为所述子对象的属性的取值,所述随机值处于预设的取值范围中;

根据所述配置对象的资源使用情况,确定所述子对象的属性的取值。

5. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

接收第二配置消息,所述第二配置消息用于删除所述配置对象;

当所述第二配置消息携带所述第一标识时,逐一检测所述配置对象的子对象是否仅以所述配置对象为父对象;

若所述配置对象的一个子对象仅以所述配置对象为父对象,则删除仅以所述配置对象为父对象的子对象;

删除所述配置对象。

6. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

接收第三配置消息,所述第三配置消息用于更新所述配置对象,所述第三配置消息包括至少一个第二索引描述,所述至少一个第二索引描述用于指示所述配置对象更新后的所有子对象,每个第二索引描述分别与所述配置对象更新后的一个子对象对应;

当所述第三配置消息携带所述第一标识时,根据所述至少一个第二索引描述以及数据

库中记录的所述配置对象更新前的子对象的信息,确定所述配置对象的第一子对象和第二子对象;其中,所述第一子对象属于所述配置对象更新后的子对象且不属于所述配置对象更新前的子对象;所述第二子对象属于所述配置对象更新前的子对象且不属于所述配置对象更新后的子对象;

检测所述第一子对象是否存在于数据库中;

若所述第一子对象不存在于数据库中,则创建所述第一子对象;

检测所述第二子对象是否仅以所述配置对象为父对象;

若所述第二子对象仅以所述配置对象为父对象,则删除所述第二子对象;

基于所述第一子对象,更新所述配置对象。

7. 一种配置管理装置,其特征在于,包括:

接收模块,用于接收第一配置消息,所述第一配置消息用于创建配置对象,其中,所述配置对象包括多个子对象,所述第一配置消息包括至少一个第一索引描述,所述至少一个第一索引描述用于指示所述配置对象的所有子对象,每个第一索引描述分别与所述配置对象的一个子对象对应;所述配置对象为虚拟系统或者策略路由;

处理模块,用于当所述第一配置消息携带第一标识时,对于每个第一索引描述,检测所述第一索引描述对应的子对象是否存在于数据库中,当所述子对象不存在于所述数据库时,创建所述子对象,所述第一标识用于指示自动处理所述配置对象;

所述处理模块,还用于基于所述配置对象的所有子对象,创建所述配置对象。

8. 根据权利要求7所述的装置,其特征在于,所述第一索引描述包括索引值;

所述处理模块,用于检测所述索引值是否存在于所述数据库中;当所述索引值不存在于数据库时,根据所述索引值,创建所述子对象。

9. 根据权利要求7或8所述的装置,其特征在于,所述子对象包括属性,所述第一索引描述还包括所述属性对应的属性数据;

所述处理模块,还用于根据所述属性数据,确定所述子对象的属性的取值。

10. 根据权利要求7或8所述的装置,其特征在于,所述第一索引描述还包括第二标识,所述第二标识用于确定策略,所述策略用于确定所述子对象的属性的取值;

所述处理模块,还用于根据所述第二标识对应的策略,确定所述子对象的属性的取值;

其中,所述策略包括以下至少一种:

以预设值作为所述子对象的属性的取值;

以随机值作为所述子对象的属性的取值,所述随机值处于预设的取值范围中;

根据所述配置对象的资源使用情况,确定所述子对象的属性的取值。

11. 根据权利要求7所述的装置,其特征在于,

所述接收模块,还用于接收第二配置消息,所述第二配置消息用于删除所述配置对象;

所述处理模块,还用于当所述第二配置消息携带所述第一标识时,逐一检测所述配置对象的子对象是否存在除所述配置对象之外的其他父对象;当所述配置对象的一个子对象不存在除所述配置对象之外的其他父对象时,删除不存在除所述配置对象之外的其他父对象的子对象;

所述处理模块,还用于删除所述配置对象。

12. 根据权利要求7所述的装置,其特征在于,

所述接收模块,还用于接收第三配置消息,所述第三配置消息用于更新所述配置对象,所述第三配置消息包括至少一个第二索引描述,所述至少一个第二索引描述用于指示所述配置对象更新后的所有子对象,每个第二索引描述分别与所述配置对象更新后的一个子对象对应;

所述处理模块,还用于当所述第三配置消息携带第一标识时,根据所述至少一个第二索引描述以及数据库中记录的所述配置对象更新前的子对象的信息,确定所述配置对象的第一子对象和第二子对象;其中,所述第一子对象属于所述配置对象更新后的子对象且不属于所述配置对象更新前的子对象;所述第二子对象属于所述配置对象更新前的子对象且不属于所述配置对象更新后的子对象;

所述处理模块,还用于检测所述第一子对象是否存在于数据库中;当所述第一子对象不存在于数据库时,创建所述第一子对象;

所述处理模块,还用于检测所述第二子对象是否仅以所述配置对象为父对象;当所述第二子对象仅以所述配置对象为父对象,删除所述第二子对象;

所述处理模块,还用于基于所述第一子对象,更新所述配置对象。

13. 一种配置管理系统,其特征在于,包括服务器和客户端;

所述客户端,用于向所述服务器发送配置消息,所述配置消息携带第一标识,所述配置消息包括第一配置消息、第二配置消息或第三配置消息;

所述服务器,用于根据所述客户端发送的配置消息执行上述权利要求1至6任一项所述的配置管理方法。

配置管理方法及装置

技术领域

[0001] 本申请涉及网络通信技术领域,尤其涉及配置管理方法及装置。

背景技术

[0002] 为了满足复杂网络中对于网络设备配置管理的需求,国际互联网工程任务组(The Internet Engineering Task Force, IETF)制定了网络配置协议(Network Configuration Protocol, NETCONF)。NETCONF提供一套管理网络设备的机制,网管设备可以使用这套机制查询、增加、修改、删除网络设备的配置,获取网络设备的配置和状态信息。

[0003] 目前,NETCONF客户端在请求NETCONF服务器创建配置对象时,如果该配置对象包括多个子对象,NETCONF客户端需要先向NETCONF服务器发送多个用于创建该配置对象的子对象的报文。在该配置对象所有的子对象均存在于NETCONF服务器的数据库中后,NETCONF客户端才能向NETCONF服务器发送用于创建配置对象的报文,以使得NETCONF服务器创建该配置对象。可见,在创建配置对象的过程中,NETCONF客户端与NETCONF服务器之间需要进行多次报文的交互。这样,一方面,造成通信资源的浪费;另一方面,也增大了NETCONF客户端与NETCONF服务器的负担。

发明内容

[0004] 本申请提供一种配置管理方法及装置,用于解决现有技术中NETCONF客户端与NETCONF服务器之间需要多次报文交互才能成功创建配置对象的问题。

[0005] 为达到上述目的,本申请采用如下技术方案:

[0006] 第一方面,提供了一种配置管理方法,包括:NETCONF服务器接收第一配置消息,该第一配置消息用于创建配置对象,该第一配置消息包括至少一个第一索引描述,前述至少一个第一索引描述用于指示所述配置对象的所有子对象,每个第一索引描述分别与配置对象的一个子对象对应;当第一配置消息携带第一标识时,对于每个第一索引描述,检测所述第一索引描述对应的子对象是否存在于数据库中;若子对象不存在于数据库中,创建该子对象;基于配置对象的所有子对象,创建配置对象。基于该方案,NETCONF服务器能够根据一个报文(即第一配置消息),成功创建配置对象,从而避免了通信资源的浪费,也降低了NETCONF服务器和NETCONF客户端的负担。

[0007] 一种可能的设计中,第一索引描述包括索引值。因此,检测所述第一索引描述对应的子对象是否存在于数据库中,包括:检测索引值是否存在于数据库中。若子对象不存在于数据库中,创建该子对象,包括:若索引值不存在于数据库中,则根据该索引值,创建子对象。这样,NETCONF服务器可以根据用于创建配置对象的报文(即第一配置消息),先创建不存在于数据库中的子对象,从而保证后续能够成功创建配置对象。

[0008] 一种可能的设计中,子对象包括属性,第一索引描述还包括子对象的属性对应的属性数据。从而,在创建子对象之前,该方法还包括:根据前述属性数据,确定子对象的属性的取值。这样,NETCONF服务器能够确定子对象的属性的取值,从而保证子对象的成功创建。

[0009] 一种可能的设计中,子对象包括属性,第一索引描述还包括第二标识,该第二标识用于确定策略,该策略用于确定子对象的属性的取值。从而,在创建子对象之前,该方法还包括:根据第二标识对应的策略,确定子对象的属性的取值。这样,NETCONF服务器能够确定子对象的属性的取值,从而保证子对象的成功创建。

[0010] 可选地,上述策略包括以下至少一种:

[0011] 以预设值作为子对象的属性的取值;

[0012] 以随机值作为子对象的属性的取值,该随机值处于预设的取值范围中;

[0013] 根据配置对象的资源使用情况,确定子对象的属性的取值。

[0014] 一种可能的设计中,该方法还包括:接收第二配置消息,该第二配置消息用于删除配置对象;当第二配置消息携带第一标识时,逐一检测配置对象的子对象是否仅以该配置对象为父对象;若配置对象的一个子对象仅以该配置对象为父对象,则删除仅以该配置对象为父对象的子对象;删除该配置对象。基于该方案,NETCONF服务器能够根据一个报文(即第二配置消息),删除所述配置对象以及冗余的子对象,从而避免了通信资源的浪费,也降低了NETCONF服务器和NETCONF客户端的负担。

[0015] 一种可能的设计中,该方法还包括:接收第三配置消息,该第三配置消息用于更新配置对象,该第三配置消息包括至少一个第二索引描述,前述至少一个第二索引描述用于指示配置对象更新后的所有子对象,每个第二索引描述分别与配置对象的一个子对象对应;当第三配置消息携带第一标识时,根据前述至少一个第二索引描述以及数据库中记录的配置对象更新前的子对象的信息,确定配置对象的第一子对象和第二子对象;其中,第一子对象属于配置对象更新后的子对象且不属于配置对象更新前的子对象;第二子对象属于配置对象更新前的子对象且不属于配置对象更新后的子对象;检测第一子对象是否存在于数据库中;若第一子对象不存在于数据库中,则创建第一子对象;检测第二子对象是否仅以配置对象为父对象;若第二子对象仅以配置对象为父对象,则删除第二子对象;基于第一子对象,更新配置对象。基于该方案,NETCONF服务器能够根据一个报文(即第三配置消息),成功更新所述配置对象,并且删除冗余的子对象,从而避免了通信资源的浪费,也降低了NETCONF服务器和NETCONF客户端的负担。

[0016] 第二方面,提供了一种配置管理装置,包括:接收模块,用于接收第一配置消息,该第一配置消息用于创建配置对象,该第一配置消息包括至少一个第一索引描述,前述至少一个第一索引描述中用于指示所述配置对象的所有子对象,每个第一索引描述分别与所述配置对象的一个子对象对应;处理模块,用于当第一配置消息携带第一标识时,对于每个第一索引描述,检测所述第一索引描述对应的子对象是否存在于数据库中;当子对象不存在于数据库时,创建该子对象;处理模块,还用于基于配置对象的所有子对象,创建配置对象。

[0017] 一种可能的设计中,第一索引描述包括索引值。处理模块,用于检测索引值是否存在于数据库中;当索引值不存在于数据库时,根据索引值,创建该子对象。

[0018] 一种可能的设计中,子对象包括属性,第一索引描述还包括属性对应的属性数据。处理模块,还用于根据属性数据,确定子对象的属性的取值。

[0019] 一种可能的设计中,第一索引描述还包括第二标识,该第二标识用于确定策略,该策略用于确定子对象的属性的取值。处理模块,还用于根据第二标识对应的策略,确定子对象的属性的取值。

[0020] 可选地,所述策略包括以下至少一种:

[0021] 以预设值作为子对象的属性的取值;

[0022] 以随机值作为子对象的属性的取值,该随机值处于预设的取值范围中;

[0023] 根据配置对象的资源使用情况,确定子对象的属性的取值。

[0024] 一种可能的设计中,接收模块,还用于接收第二配置消息,该第二配置消息用于删除配置对象。处理模块,还用于当第二配置消息携带第一标识时,逐一检测配置对象的子对象是否存在除配置对象之外的其他父对象;当配置对象的一个子对象不存在除配置对象之外的其他父对象时,删除不存在除配置对象之外的其他父对象的子对象。处理模块,还用于删除配置对象。

[0025] 一种可能的设计中,接收模块,还用于接收第三配置消息,该第三配置消息用于更新配置对象,第三配置消息包括至少一个第二索引描述,前述至少一个第二索引描述用于指示配置对象更新后的所有子对象,每个第二索引描述分别与配置对象的一个子对象对应。处理模块,还用于当第三配置消息携带第一标识时,根据至少一个第二索引描述以及数据库中记录的配置对象更新前的子对象的信息,确定配置对象的第一子对象和第二子对象;其中,第一子对象属于配置对象更新后的子对象且不属于配置对象更新前的子对象;第二子对象属于配置对象更新前的子对象且不属于配置对象更新后的子对象。处理模块,还用于检测第一子对象是否存在于数据库中;当第一子对象不存在于数据库时,创建第一子对象。处理模块,还用于检测第二子对象是否仅以配置对象为父对象;当第二子对象仅以配置对象为父对象,删除第二子对象。处理模块,还用于基于第一子对象,更新配置对象。

[0026] 第三方面,提供了一种服务器,包括:处理器和存储器;该存储器用于存储指令或程序代码,当该服务器运行时,该处理器执行该存储器存储的该指令或程序代码,以使该服务器执行如上述第一方面中任一项所述的配置管理方法。

[0027] 第四方面,提供了一种服务器,包括:处理器;所述处理器用于与存储器耦合,并读取存储器中的指令或程序代码之后,根据所述指令或程序代码执行如上述第一方面中任一项所述的配置管理方法。

[0028] 第五方面,提供了一种计算机可读存储介质,该计算机可读存储介质中存储有指令或程序代码,当其在计算机上运行时,使得计算机可以执行上述第一方面中任一项所述的配置管理方法。

[0029] 第六方面,提供了一种包含指令或程序代码的计算机程序产品,当其在计算机上运行时,使得计算机可以执行上述第一方面中任一项所述的配置管理方法。

[0030] 第七方面,提供了一种芯片系统,该芯片系统包括处理器,用于支持服务器实现上述第一方面所涉及的功能。在一种可能的设计中,该芯片系统还包括存储器,该存储器,用于保存服务器必要的程序指令和数据。该芯片系统,可以由芯片构成,也可以包含芯片和其他分立器件。

[0031] 第八方面,提供了一种配置管理系统,包括:服务器和客户端;其中,客户端用于向服务器发送配置消息,该配置消息携带第一标识,该配置消息包括第一配置消息、第二配置消息或第三配置消息;服务器用于根据接收的配置消息执行第一方面中任一项所述的配置管理方法。

[0032] 其中,第二方面至第八方面中任一种设计方式所带来的技术效果可参见第一方面

中不同设计方式所带来的技术效果,此处不再赘述。

附图说明

- [0033] 图1为本申请实施例提供的一种NETCONF配置系统的架构示意图;
- [0034] 图2为本申请实施例提供的一种设备的结构示意图;
- [0035] 图3为现有的创建配置对象的流程示意图;
- [0036] 图4为本申请实施例提供的一种配置管理方法的流程示意图;
- [0037] 图5为现有的删除配置对象的流程示意图;
- [0038] 图6为本申请实施例提供的另一种配置管理方法的流程示意图;
- [0039] 图7为现有的更新配置对象的流程示意图;
- [0040] 图8为本申请实施例提供的另一种配置管理方法的流程示意图;
- [0041] 图9为本申请实施例提供的一种配置管理装置的结构示意图。

具体实施方式

[0042] 本申请中的术语“第一”、“第二”等仅是为了区分不同的对象,并不对其顺序进行限定。例如,第一基站和第二基站仅仅是为了区分不同的基站,并不对其先后顺序进行限定。

[0043] 本申请中术语“和/或”,仅仅是一种描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。另外,本申请中字符“/”,一般表示前后关联对象是一种“或”的关系。

[0044] 需要说明的是,本申请中,“示例性的”或者“例如”等词用于表示作例子、例证或说明。本申请中被描述为“示例性的”或者“例如”的任何实施例或设计方案不应被解释为比其他实施例或设计方案更优选或更具优势。确切而言,使用“示例性的”或者“例如”等词旨在以具体方式呈现相关概念。

[0045] 为清楚阐述本申请实施例的技术方案,首先对本申请实施例涉及的相关术语进行简要的介绍。

[0046] NETCONF是一种基于可扩展标记语言(Extensible Markup Language,XML)的网络管理协议,NETCONF采用分层结构,分为:内容层、操作层、消息层和传输层。NETCONF协议操作被实现为远程过程调用(remote procedure calls,RPCs)。

[0047] 内容层描述了网络管理所涉及的数据,例如配置数据、状态数据、统计信息等。

[0048] 操作层定义了一组基本操作,例如:<get-config>(查询配置数据)、<edit-config:create>(创建配置对象)、<edit-config:delete>(删除配置对象)、<edit-config:repace>(更新配置对象)<copy-config>(覆盖)等。

[0049] 消息层提供一种简洁的、与传输协议无关的消息编码机制。具体的,NETCONF采用基于RPC(RPC-based)的通信模型。NETCONF通信双方采用<rpc>元素来封装请求报文,采用<rpc-reply>元素来封装与上述请求报文相对应的响应报文。另外,<rpc>包括一个强制属性“message-id”,message-id用于请求报文与响应报文的配对,即请求报文与对应的响应报文具有相同的message-id。值得说明的是,“message-id”的取值是随机产生的,不能由用户指定,以避免出现重复。

[0050] 传输层主要用于在网络设备和NETCONF客户端之间建立通信路径,传输层的协议具体可以为安全壳(Secure Shell,SSH)协议、安全传输层(Transport Layer Security,TLS)协议、简单对象访问协议(Simple Object Access Protocol,SOAP)以及块可扩展交换协议(Blocks Extensible Exchange Protocol,BEEP)等。

[0051] 图1为本申请实施例提供的NETCONF配置系统的架构示意图。该NETCONF配置系统包括:NETCONF客户端和NETCONF服务器。NETCONF客户端和NETCONF服务器之间由面向连接的传输协议(例如,SSH、TLS、SOAP或BEEP等)建立会话。NETCONF客户端和NETCONF服务器之间通信的内容是NETCONF协议定义的双向的XML式的RPC消息,包括从NETCONF客户端发送到NETCONF服务器的请求消息,以及NETCONF服务器向NETCONF客户端返回的响应消息。

[0052] 其中,NETCONF服务器可以为防火墙设备、接入设备、交换机等。NETCONF客户端可以为网管设备、控制器等。

[0053] 一种可能的实现方式中,NETCONF客户端,用于发送配置消息,所述配置消息携带第一标识,所述配置消息包括第一配置消息、第二配置消息或第三配置消息(后续将对每个配置消息进行详细说明)。NETCONF服务器,用于执行本申请实施例提供的配置管理方法。

[0054] 可选地,本申请实施例图1中的NETCONF服务器或者NETCONF客户端可以由一个设备实现,也可以由多个设备共同实现,还可以是一个设备内的一个功能模块,本申请实施例对此不作具体限定。可以理解的是,上述功能模块既可以是硬件设备中的网络元件,也可以是在硬件上运行的软件功能,或者是平台(例如,云平台)上实例化的虚拟化功能。

[0055] 例如,图1所示的NETCONF服务器或NETCONF客户端可以通过图2中的设备来实现。

[0056] 图2为本申请实施例提供的一种设备的硬件结构示意图。该设备200包括至少一个处理器201,通信线路202,存储器203以及至少一个通信接口204。

[0057] 处理器201可以是一个通用中央处理器(central processing unit,CPU),微处理器,特定应用集成电路(application-specific integrated circuit,ASIC),或一个或多个用于控制本申请方案程序执行的集成电路。

[0058] 通信线路202可包括一通路,在上述组件之间传送信息。

[0059] 通信接口204,使用任何收发器一类的装置,用于与其他设备或通信网络通信,如以太网,无线接入网(radio access network,RAN),无线局域网(wireless local area networks,WLAN)等。

[0060] 存储器203可以是只读存储器(read-only memory,ROM)或可存储静态信息和指令的其他类型的静态存储设备,随机存取存储器(random access memory,RAM)或者可存储信息和指令的其他类型的动态存储设备,也可以是电可擦可编程只读存储器(electrically erasable programmable read-only memory,EEPROM)、只读光盘(compact disc read-only memory,CD-ROM)或其他光盘存储、光碟存储(包括压缩光碟、激光碟、光碟、数字通用光碟、蓝光光碟等)、磁盘存储介质或者其他磁存储设备、或者能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质,但不限于此。存储器203可以独立存在,通过通信线路202与处理器相连接。存储器203也可以和处理器201集成在一起。

[0061] 其中,存储器203用于存储执行本申请方案的计算机可执行指令。处理器201用于执行存储器203中存储的计算机可执行指令,从而实现本申请下述实施例提供的确定位置

信息的方法。

[0062] 可选地,本申请实施例中的计算机可执行指令也可以称之为应用程序代码,本申请实施例对此不作具体限定。

[0063] 在具体实现中,作为一种实施例,处理器201可以包括一个或多个CPU,例如图2中的CPU0和CPU1。

[0064] 在具体实现中,作为一种实施例,设备200可以包括多个处理器,例如图2中的处理器201和处理器207。这些处理器中的每一个可以是一个单核(single-CPU)处理器,也可以是一个多核(multi-CPU)处理器。这里的处理器可以指一个或多个设备、电路、和/或用于处理数据(例如计算机程序指令)的处理核。

[0065] 在具体实现中,作为一种实施例,设备200还可以包括输出设备205和输入设备206。输出设备205和处理器201通信,可以以多种方式来显示信息。例如,输出设备205可以是液晶显示器(liquid crystal display,LCD),发光二极管(light emitting diode,LED),阴极射线管(cathode ray tube,CRT),或投影仪(projector)等。输入设备206和处理器201通信,可以以多种方式接收用户的输入。例如,输入设备206可以是鼠标、键盘、触摸屏设备或传感设备等。

[0066] 上述的设备200可以是一个通用设备或者是一个专用设备。在具体实现中,设备200可以是台式机、便携式电脑、网络服务器、掌上电脑(personal digital assistant,PDA)、移动手机、平板电脑、无线终端设备、嵌入式设备或有图2中类似结构的设备。本申请实施例不限定设备200的类型。

[0067] 目前,在NETCONF服务器创建配置对象的过程中,若配置对象包括多个子对象,且配置对象的子对象不存在于NETCONF服务器的数据库中,则会导致NETCONF服务器创建配置对象失败。为了避免这种情况,一般情况下,如图3所示,NETCONF客户端先发送创建子对象的报文(请求报文1、请求报文2)给NETCONF服务器,以便于NETCONF服务器创建子对象。NETCONF服务器也需要向NETCONF客户端通报创建子对象的情况(响应报文1、响应报文2)。在配置对象所有的子对象均存在于NETCONF服务器的数据库后,NETCONF客户端再发送创建配置对象的报文(请求报文n)给NETCONF服务器,以便于NETCONF服务器成功创建该配置对象。但是,上述多次报文交互的过程,即浪费了通信资源,又增加了NETCONF服务器与NETCONF客户端的负担。

[0068] 为了解决上述问题,如图4所示,本申请实施例提供了一种配置管理方法,该方法应用于NETCONF服务器,该方法包括如下步骤:

[0069] S101、接收第一配置消息。

[0070] 其中,所述第一配置消息用于创建配置对象。示例性的,所述配置对象包括:虚拟系统或者策略路由,本申请实施例对此不作任何限制。

[0071] 可选地,所述第一配置消息包括所述配置对象的属性。所述配置对象的属性包括名称、描述、子对象等。示例性的,所述子对象包括但不限于:虚拟局域网(virtual local area network,VLAN),虚拟局域扩展网(virtual extensible local area network,VXLAN),或者资源类。其中,资源类用于指示配置对象所需要的网络资源。可选地,资源类的属性包括:最大在线用户数目和/或最大接口带宽,本申请实施例对此不作任何限制。

[0072] 可选地,所述第一配置消息还包括至少一个第一索引描述(index profile),前述

至少一个第一索引描述用于指示所述配置对象的所有子对象。

[0073] 每个第一索引描述分别与所述配置对象的一个子对象对应,即每个第一索引描述用于指示所述配置对象的一个子对象。第一索引描述包括索引值,该索引值用于在NETCONF服务器的数据库中查询对应的子对象。可选地,上述索引值可以为子对象的名称或标识,本申请实施例对此不作限定。

[0074] 例如,第一索引描述为<assign-vxlan>5000</assign-vxlan>,其中<assign-vxlan>说明该子对象的类型为vxlan,5000为索引值,第一索引描述用于说明配置对象包括子对象vxlan 5000。

[0075] 又例如,第一索引描述为<assign-vlan>10</assign-vlan>,其中<assign-vlan>说明该子对象的类型为vlan,10为索引值,该第一索引描述用于说明配置对象包括子对象vlan 10。

[0076] 一种可选地实现方式中,在NETCONF服务器接收到第一配置消息后,NETCONF服务器根据所述第一配置消息中携带的所述配置对象的名称,检测所述配置对象是否存在于数据库中。若所述配置对象存在于数据库中,NETCONF服务器向NETCONF客户端返回第一报错消息,所述第一报错消息用于表示该配置对象已存在。示例性,该第一报错消息可为如下形式:“data-exists”。若所述配置对象不存在于数据库中,NETCONF服务器检测所述第一配置消息是否携带第一标识。若所述第一配置消息未携带第一标识,NETCONF服务器执行现有技术的方案,直接创建所述配置对象。若所述第一配置消息携带第一标识,则NETCONF执行下述步骤S102-S104。

[0077] 该第一标识用于指示NETCONF服务器自动处理所述配置对象。自动处理所述配置对象包括自动创建/删除/更新所述配置对象和所述配置对象的子对象。示例性的,所述第一标识可以为:config autoconfig="true"。当然,所述第一标识可以采用其他实现方式,本申请实施例对此不作任何限制。

[0078] S102、当所述第一配置消息携带第一标识时,对于每个第一索引描述,检测所述第一索引描述对应的子对象是否存在于数据库中。

[0079] 一种可选地实现方式中,由于数据库会记录存在于数据库的子对象的索引值,因此NETCONF服务器检测所述第一索引描述的索引值是否存在于数据库中,以判断所述第一索引描述对应的子对象是否存在于数据库中。

[0080] S103、若所述子对象不存在于数据库中,创建所述子对象。

[0081] 结合步骤S102进行说明,当第一索引描述的索引值不存在于数据库时,说明该第一索引描述对应的子对象不存在于数据库中。这种情况下,NETCONF服务器根据该第一索引描述的索引值,创建该第一索引描述对应的子对象。

[0082] 例如,第一索引描述为<assign-vlan>10</assign-vlan>,根据该第一索引描述,如果NETCONF服务器的数据库中不存在vlan 10,NETCONF服务器可直接创建子对象vlan 10。

[0083] 另外,某些类型的子对象除了名称还有其他属性,例如类型为vxlan的子对象除了名称还有目标地址这一属性。对于这一类子对象,NETCONF服务器还需先确定子对象的属性的取值,才能成功创建该子对象。对此,本申请实施例提供如下确定子对象的属性的取值的方法。

[0084] 方法一、所述第一索引描述携带属性数据,因此,NETCONF服务器可以根据所述第一索引描述中的属性数据,确定子对象的属性的取值。

[0085] 例如,对应子对象vxlan 5000的第一索引描述携带的属性数据为192.168.0.2,则NETCONF服务器可以根据该属性数据,确定子对象vxlan 5000的目的地址为192.168.0.2。

[0086] 方法二、所述第一索引描述还包括第二标识,所述第二标识用于确定策略,所述策略用于确定所述子对象的属性的取值。因此,所述NETCONF服务器根据第二标识对应的策略,确定所述子对象的属性的取值。

[0087] 可选地,所述策略包括以下至少一种:

[0088] (1) 以预设值作为子对象的属性的取值。

[0089] 例如,目标地址的预设值为192.168.0.2,则NETCONF服务器可确定类型为vxlan的子对象的目标地址为192.168.0.2。

[0090] (2) 以随机值作为所述子对象的属性的取值,所述随机值处于预设的取值范围内。

[0091] 例如,目标地址的预设的取值范围为192.168.0.2到192.168.0.254,因此,可选地,NETCONF服务器可确定类型为vxlan的子对象的目标地址为192.168.0.3。

[0092] (3) 根据所述配置对象的资源使用情况,确定所述子对象的属性的取值。

[0093] 例如,假设配置对象为虚拟系统,子对象为资源类,若该虚拟系统配置的策略路由的数目越多,则资源类中最大在线用户数目以及最大接口带宽这两个属性的取值也越大。

[0094] 当然,所述策略还可以采用其他实现方式,本申请实施例对此不作任何限制。

[0095] 示例性的,所述第二标识可以为:autoconfig-strategy="xxx"。其中,xxx为不同的内容时,第二标识与不同的策略对应。例如,当xxx为default时,则第二标识与上述策略(1)对应。又例如,当xxx为random时,第二标识与上述策略(2)对应。当xxx为dynamic时,第二标识与上述策略(3)对应。当然,所述第二标识可以采用其他实现方式,本申请实施例对此不作任何限制。

[0096] 可选地,为了简化报文,第一配置消息携带对应于所有第一索引描述的第二标识,而不是每一个第一索引描述分别携带各自的第二标识。从而,在第一配置消息携带唯一的第二标识的情况下,NETCONF服务器根据第一配置消息携带的第二标识对应的策略,确定每个第一索引描述对应的子对象的属性的取值。

[0097] 值得说明的是,所述配置对象可能包括n级子对象,这种情况下,第一级子对象以所述配置对象作为父对象,第二级子对象以某个第一级子对象作为父对象,以此类推,第n级子对象以某个第n-1级子对象作为父对象。其中,n为自然数。

[0098] 在所述配置对象包括n级子对象的场景下,NETCONF服务器优先创建第n级的子对象,然后再创建第n-1级的子对象,以此类推,最后创建第1级的子对象。

[0099] 例如,NETCONF客户端请求NETCONF服务器创建策略路由1,策略路由1包括子对象虚拟系统vsys 1,虚拟系统vsys 1包括子对象vlan 100以及资源类rc 1。可以理解的是,对于策略路由1来说,虚拟系统vsys 1即为第一级子对象,vlan 100以及资源类rc 1即为第二级子对象。若虚拟系统vsys 1、vlan 100以及资源类rc 1均不存在于NETCONF服务器的数据库中,则NETCONF服务器先创建vlan 100和资源类rc 1;然后,基于vlan 100和资源类rc 1,创建虚拟系统vsys 1。

[0100] S104、基于所述配置对象的所有子对象,创建所述配置对象。

[0101] 其中,步骤S104的具体实现方式可参考现有技术,本申请实施例对此不作任何限制。

[0102] 可选地,在成功创建所述配置对象后,NETCONF服务器向NETCONF客户端返回第一成功响应消息,该第一成功响应消息用于表示创建所述配置对象成功。可选地,为了简化第一成功响应消息在网络传输中的带宽开销,第一成功响应消息可以为如下形式:<ok>。

[0103] 可选地,在创建所述配置对象失败后,NETCONF服务器向NETCONF客户端返回第一失败响应消息,该第一失败响应消息用于表示创建所述配置对象失败。可选地,为了简化第一失败响应消息在网络传输中的带宽开销,第一失败响应消息可以为如下形式:<rpc-error>。

[0104] 采用图4所示的方案,NETCONF服务器能够根据一个报文(即第一配置消息),成功创建配置对象,从而避免了通信资源的浪费,也降低了NETCONF服务器和NETCONF客户端的负担。

[0105] 在基于图4所示方法创建所述配置对象后,由于配置对象失效、组网环境变更等因素,NETCONF服务器可能需要删除所述配置对象。如图5所示,目前,NETCONF服务器在接收到用于删除配置对象的报文(请求报文1)后,仅会删除该配置对象,而不会一并删除该配置对象的子对象。但是,在配置对象被删除后,仅以该配置对象为父对象的子对象对NETCONF服务器来说是冗余的。因此,为了节省NETCONF服务器的存储空间,NETCONF客户端还需向NETCONF服务器额外发送用于删除冗余的子对象的请求报文(请求报文2和请求报文3),NETCONF服务器也需要向NETCONF客户端通报删除子对象的情况(响应报文2和响应报文3)。这样,即浪费了通信资源,又增加了NETCONF服务器与NETCONF客户端的负担。

[0106] 为了解决该问题,如图6所示,本申请实施例提供另一种配置管理方法,应用于NETCONF服务器,该方法包括以下步骤:

[0107] S201、接收第二配置消息。

[0108] 其中,所述第二配置消息用于删除所述配置对象。所述第二配置消息包括所述配置对象的名称。

[0109] 一种可选地实现方式中,在接收到所述第二配置消息后,NETCONF服务器根据第二配置消息携带的所述配置对象的名称,检测所述配置对象是否存在于数据库中。若所述配置对象不存在于数据库中,NETCONF服务器向NETCONF客户端返回第二报错消息,该第二报错消息用于表示该配置对象不存在。示例性的,该第二报错消息可以为如下形式:“data-missing”。或者,NETCONF服务器忽略掉第二配置消息,不作任何操作。若所述配置对象存在于数据库中,NETCONF服务器检测所述第二配置消息是否携带第一标识。若所述第二配置消息未携带第一标识,则NETCONF服务器执行现有技术的方案,直接删除所述配置对象。若所述第二配置消息携带第一标识,则NETCONF服务器执行下述步骤S202-S204。

[0110] S202、当所述第二配置消息携带第一标识时,逐一检测所述配置对象的子对象是否仅以所述配置对象为父对象。

[0111] 可以理解的是,对于配置对象的子对象来说,配置对象为该子对象的父对象。

[0112] 一种可选地实现方式中,NETCONF服务器根据所述第二配置消息携带的配置对象的名称,从数据库中确定所述配置对象的所有子对象。然后,NETCONF服务器逐一检测所述配置对象的子对象是否仅以所述配置对象为父对象。

[0113] S203、删除仅以所述配置对象为父对象的子对象。

[0114] 可以理解的是,配置对象的子对象仅以所述配置对象为父对象,相当于该子对象不存在除所述配置对象之外的其他父对象。

[0115] 值得说明的是,对于NETCONF服务器来说,若一个子对象仅存在一个父对象,且该父对象待删除或者已删除,则该子对象是冗余子对象。在这种情况下,为了节省存储空间,NETCONF服务器从数据库中删除冗余子对象的相关数据。

[0116] 值得说明的是,在配置对象包括n级子对象的场景下,NETCONF服务器先删除第1级子对象中的冗余子对象,再删除第2级子对象中的冗余子对象,依次类推,最后删除第n级子对象中的冗余子对象。

[0117] S204、删除所述配置对象。

[0118] 一种可选地实现方式中,NETCONF服务器从数据库中删除所述配置对象的相关数据。

[0119] 值得说明的是,本申请实施例不限制步骤S202-S203与步骤S204在执行顺序上的先后区别。即NETCONF服务器可以先执行步骤S202-S203,再执行步骤S204。或者,NETCONF服务器可以先执行步骤S204,再执行步骤S202-S203。

[0120] 可选地,在成功删除所述配置对象后,NETCONF服务器向NETCONF客户端返回第二成功响应消息,该第二成功响应消息用于表示删除所述配置对象成功。可选地,为了简化第二成功响应消息在网络传输中的带宽开销,第二成功响应消息可以为如下形式:<ok>。

[0121] 可选地,在删除所述配置对象失败后,NETCONF服务器向NETCONF客户端返回第二失败响应消息,该第二失败响应消息用于表示删除所述配置对象失败。可选地,为了简化第二失败响应消息在网络传输中的带宽开销,第二失败响应消息可以为如下形式:<rpc-error>。

[0122] 采用图6所示的方案,NETCONF服务器能够根据一个报文(即第二配置消息),删除所述配置对象以及冗余子对象,从而避免了通信资源的浪费,也降低了NETCONF服务器和NETCONF客户端的负担。

[0123] 在基于图4所示方法创建所述配置对象后,由于组网环境变更等因素,NETCONF服务器可能需要更新所述配置对象。目前,如图7所示,在更新配置对象的过程中,NETCONF客户端需要先向NETCONF服务器发送用于创建更新后的子对象的报文(请求报文1),以便于NETCONF服务器创建更新后的子对象,从而保证所有更新后的子对象均存在于数据库中。然后,NETCONF客户端向NETCONF服务器发送用于更新配置对象的报文(请求报文2);NETCONF根据该报文,更新所述配置对象。最后,所述配置对象的某些更新前的子对象在更新后不再是所述配置对象的子对象,并且,这些更新前的子对象仅以所述配置对象为父对象。这种情况下,这一类子对象对于NETCONF服务器而言是冗余的。因此,为了节省NETCONF服务器的存储空间,NETCONF客户端还需要发送用于删除冗余子对象的报文(请求报文3)给NETCONF服务器。可见,在现有的更新配置对象的过程中,NETCONF客户端与NETCONF服务器需要进行多次报文交互,即浪费了通信资源,又增大了NETCONF客户端与NETCONF服务器的负担。

[0124] 为了解决该问题,如图8所示,本申请实施例提供另一种配置管理方法,该方法应用于NETCONF服务器,该方法包括以下步骤:

[0125] S301、接收第三配置消息。

[0126] 其中,所述第三配置消息用于更新配置对象。所述第三配置消息包括所述配置对象的属性。所述配置对象的属性包括名称、子对象等。

[0127] 可选地,所述第三配置消息还包括至少一个第二索引描述,所述至少一个第二索引描述用于指示所述配置对象更新后的所有子对象。每个第二索引描述分别与所述配置对象更新后的一个子对象对应,即每个第二索引描述用于指示所述配置对象更新后的一个子对象。

[0128] 第二索引描述包括索引值,该索引值用于在NETCONF服务器的数据库中查询对应的子对象。

[0129] 例如,第二索引描述为<assign-vxlan>4000</assign-vxlan>, <assign-vxlan>说明该子对象的类型为vxlan,4000为索引值,第二索引描述用于说明配置对象在更新后存在子对象vxlan4000。

[0130] 一种可选地实现方式中,在接收到所述第三配置消息后,NETCONF服务器检测所述第三配置消息是否携带第一标识。若所述第三配置消息未携带第一标识,则NETCONF服务器执行现有技术的方案,直接更新所述配置对象。若所述第三配置消息携带第一标识,则NETCONF服务器执行下述步骤S302-S307。

[0131] 示例性的,所述第一标识可以为:config autoconfig="true"。

[0132] S302、当所述第三配置消息携带第一标识时,根据所述至少一个第二索引描述以及数据库中记录的所述配置对象更新前的子对象的信息,确定所述配置对象的第一子对象和第二子对象。

[0133] 其中,所述更新前的子对象的信息包括更新前的子对象的索引值。

[0134] 其中,所述第一子对象是所述配置对象更新后的子对象且不是所述配置对象更新前的子对象。例如,配置对象更新前的子对象有A1、B1和C1,该配置对象更新后的子对象有A1、B1和C2,则子对象C2即为第一子对象。

[0135] 所述第二子对象是所述配置对象更新前的子对象且不是所述配置对象更新后的子对象。例如,配置对象更新前的子对象有A1、B1和C1,该配置对象更新后的子对象有A1、B1和C2,则子对象C1即为第二子对象。

[0136] 可选地,第一子对象可采用如下方法来确定:以每个第二索引描述的索引值,查询数据库中记录的所述配置对象更新前的子对象的信息中是否存在相同的索引值。若不存在,则所述第二索引描述对应的子对象为第一子对象。

[0137] 可选地,第二子对象可采用如下方法来确定:以更新前的子对象的索引值,查询所有的第二索引描述中是否存在相同的索引值。若所有的第二索引描述中均不存在与更新前的子对象的索引值相同的索引值,则该更新前的子对象为第二子对象。

[0138] S303、检测所述第一子对象是否存在于数据库中。

[0139] S304、若所述第一子对象不存在于数据库中,则创建所述第一子对象。

[0140] 其中,步骤S303-S304的具体实现方式可参考图4所示方法中步骤S102-S103的描述,本申请实施例对此不再赘述。

[0141] S305、检测所述第二子对象是否仅以所述配置对象为父对象。

[0142] S306、若所述第二子对象仅以所述配置对象为父对象,则删除所述第二子对象。

[0143] 其中,步骤S305-S306的具体实现方式可参考图6所示方法中步骤S202-S203的描

述,本申请实施例对此不再赘述。

[0144] S307、基于所述第一子对象,更新所述配置对象。

[0145] 其中,步骤S307可参考现有技术的实现方式,本申请实施例对此不再赘述。

[0146] 可选地,在成功更新所述配置对象后,NETCONF服务器向NETCONF客户端返回第三成功响应消息,该第三成功响应消息用于表示更新所述配置对象成功。可选地,为了简化第三成功响应消息在网络传输中的带宽开销,第三成功响应消息可以为如下形式:<ok>。

[0147] 可选地,在更新所述配置对象失败后,NETCONF服务器向NETCONF客户端返回第三失败响应消息,该第三失败响应消息用于表示更新所述配置对象失败。可选地,为了简化第三失败响应消息在网络传输中的带宽开销,第三失败响应消息可以为如下形式:<rpc-error>。

[0148] 采用图8所示的方案,NETCONF服务器能够根据一个报文(即第三配置消息),成功更新所述配置对象,并且删除冗余的子对象,从而避免了通信资源的浪费,也降低了NETCONF服务器和NETCONF客户端的负担。

[0149] 上述主要从NETCONF服务器的角度对本申请实施例提供的方案进行了介绍。可以理解的是,上述NETCONF服务器为了实现上述功能,其包括了执行各个功能相应的硬件结构或软件模块。本领域技术人员应该很容易意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,本申请能够以硬件或硬件和计算机软件的结合形式来实现。某个功能究竟以硬件还是计算机软件驱动硬件的方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。

[0150] 本申请实施例可以根据上述方法示例对NETCONF服务器进行功能模块的划分,例如,可以对应各个功能划分各个功能模块,也可以将两个或两个以上的功能集成在一个处理模块中。上述集成的模块既可以采用硬件的形式实现,也可以采用软件功能模块的形式实现。需要说明的是,本申请实施例中对模块的划分是示意性的,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式。

[0151] 比如,在采用对应各个功能划分各个功能模块的情况下,图9示出了能够实现上述实施例中的方法的配置管理装置900的一种可能的结构示意图。该配置管理装置900可以为NETCONF服务器,还可以为NETCONF服务器中的芯片等形态。如图9所示,该配置管理装置900包括:处理模块901和接收模块902。其中,所述处理模块901用于支持该配置管理装置900执行图4中的步骤S102-S104,图6中的步骤S202-S204,图8中的步骤S302-S307,和/或用于本文所描述的技术的其它过程。所述接收模块902,用于支持该配置管理装置900执行图4中的步骤S101,图6中的步骤S201,图8中的步骤S301,和/或用于本文所描述的技术的其它过程。其中,上述方法实施例涉及的所有相关内容均可以援引到对应功能模块的功能描述,在此不再赘述。

[0152] 在本申请实施例中,该装置以对应各个功能划分各个功能模块的形式来呈现,或者,该装置以采用集成的方式划分各个功能模块的形式来呈现。这里的“模块”可以包括ASIC,电路,执行一个或多个软件或固件程序的处理器和存储器,集成逻辑电路,或其他可以提供上述功能的器件。

[0153] 在一个简单的实施例中,本领域的技术人员可以想到该配置管理装置900可以采

用图2所示的设备200形式。比如,图9中的处理模块901可以通过图2的设备200中的处理器201来实现。具体的,接收模块902可以通过由处理器201来调用存储器202中存储的应用程序代码来执行,本申请实施例对此不作任何限制。

[0154] 可选地,本申请实施例还提供了一种芯片系统,该芯片系统包括处理器,用于支持NETCONF服务器实现上述配置管理方法。在一种可能的设计中,该芯片系统还包括存储器。该存储器,用于保存必要的程序指令和数据。当然,存储器也可以不在芯片系统中。该芯片系统,可以由芯片构成,也可以包括芯片和其他分立器件,本申请实施例对此不作具体限定。

[0155] 由于本申请实施例提供的装置可用于执行上述配置管理方法,因此其所能获得的技术效果可参考上述方法实施例,本申请实施例在此不再赘述。

[0156] 以上所述,仅为本申请的具体实施方式,但本申请的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本申请揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应以所述权利要求的保护范围为准。



图1

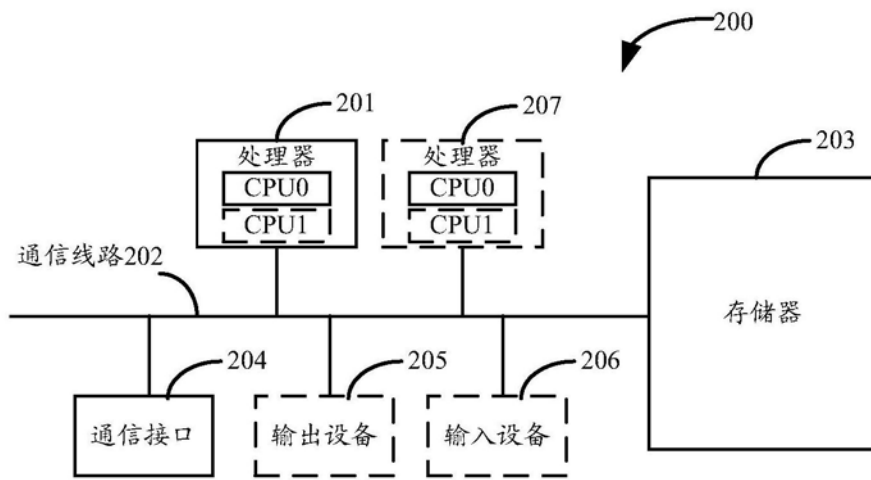


图2

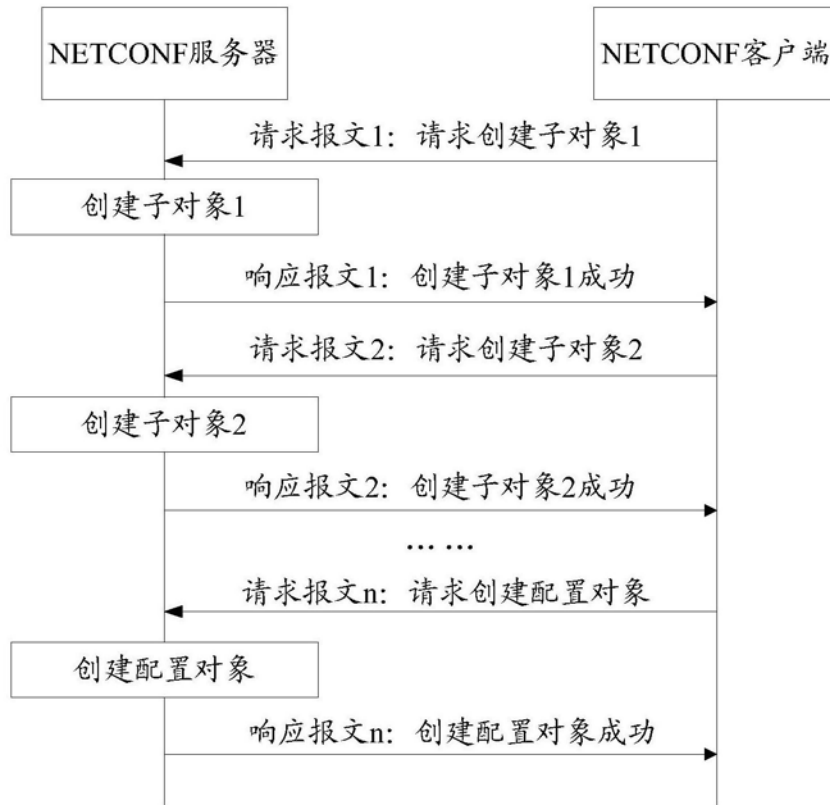


图3

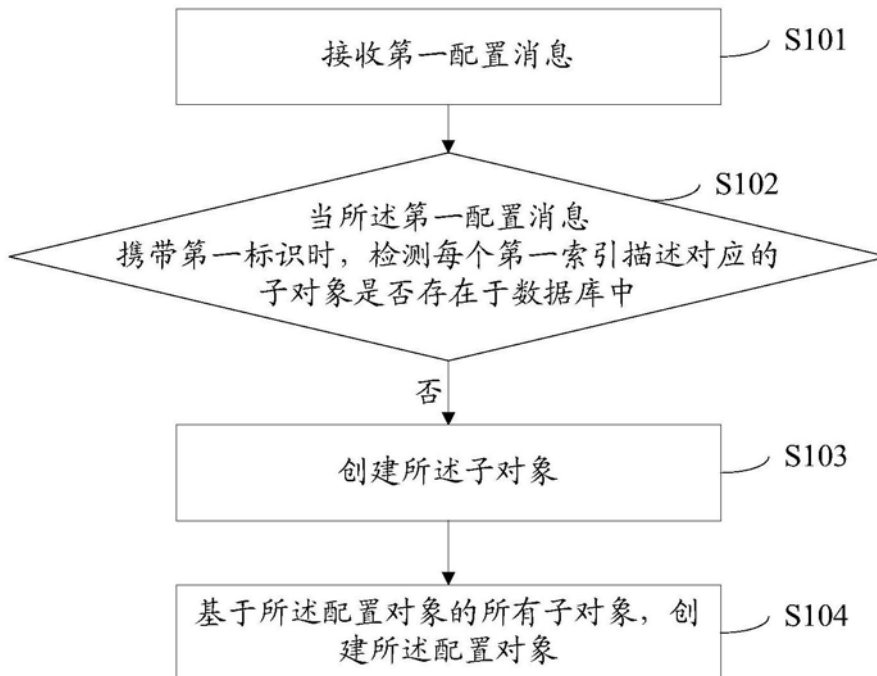


图4

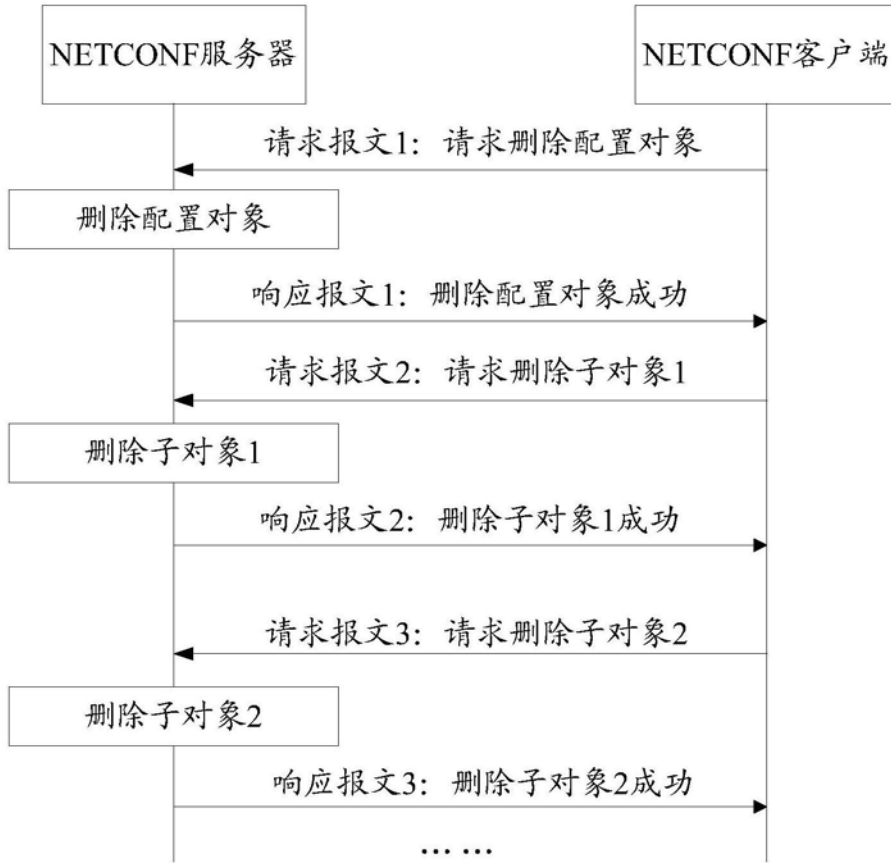


图5

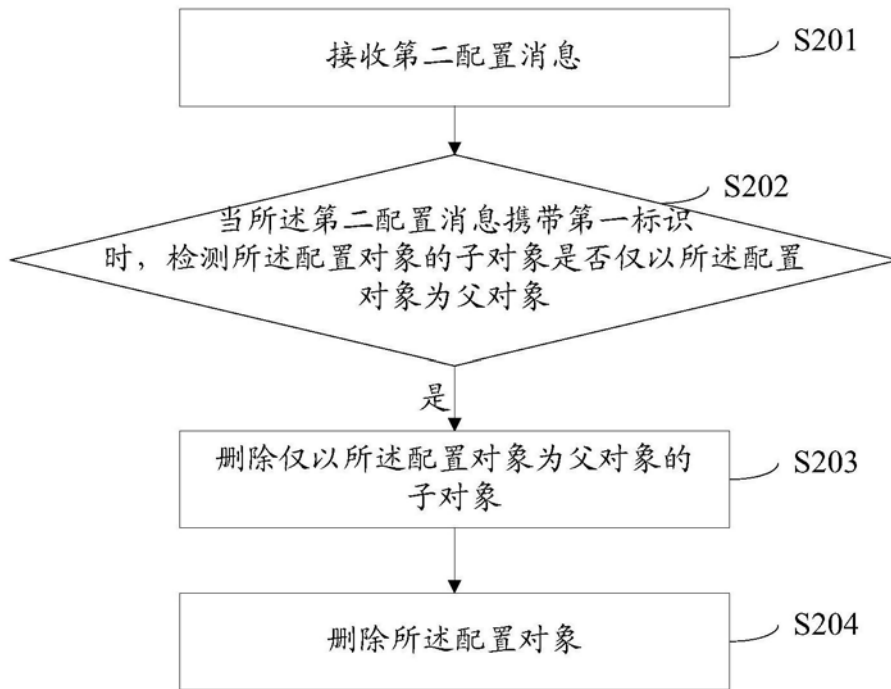


图6

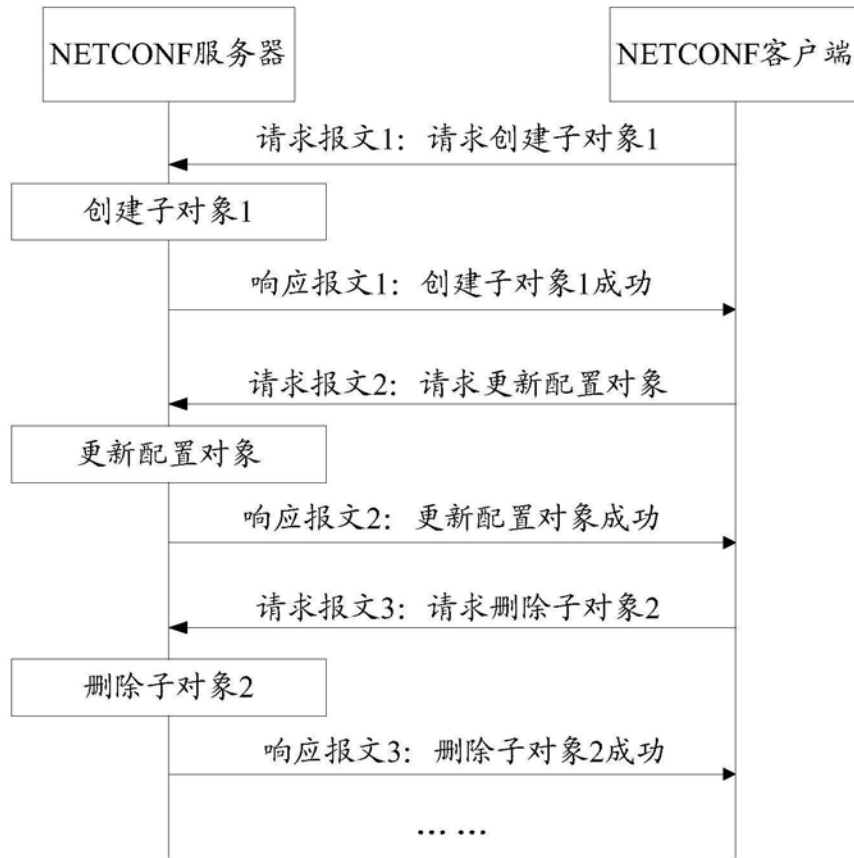


图7

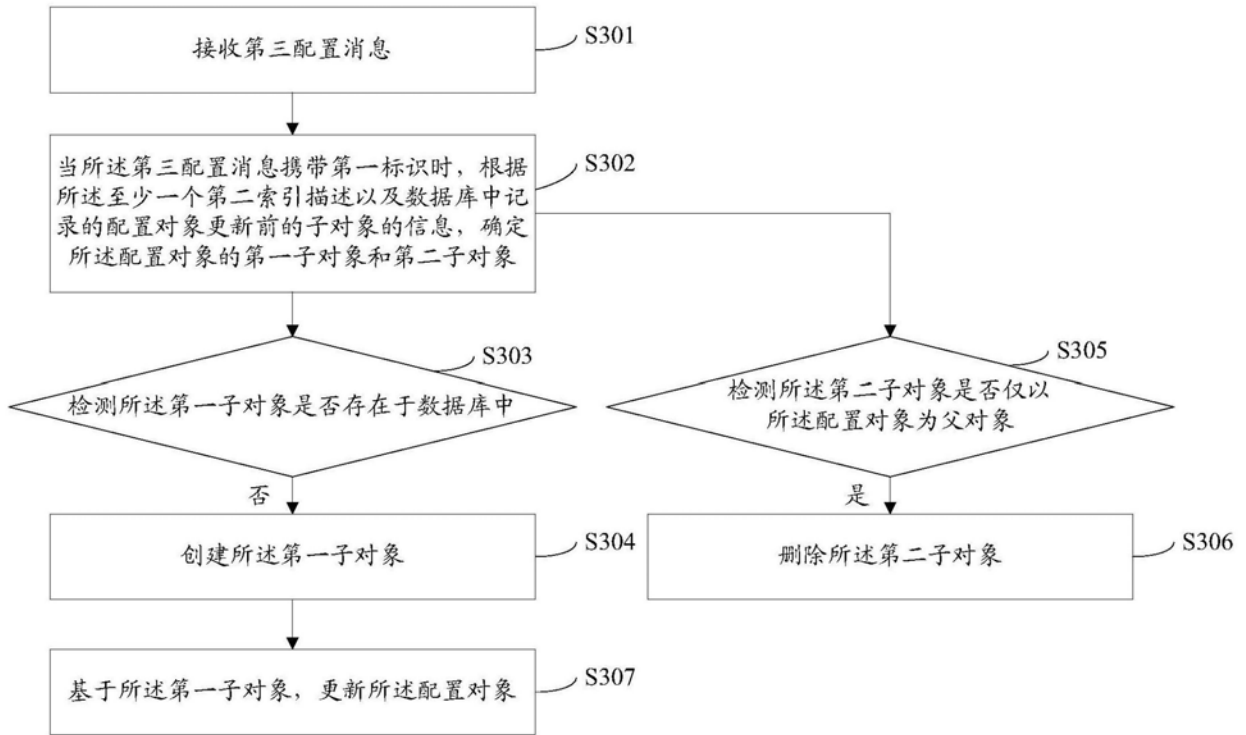


图8

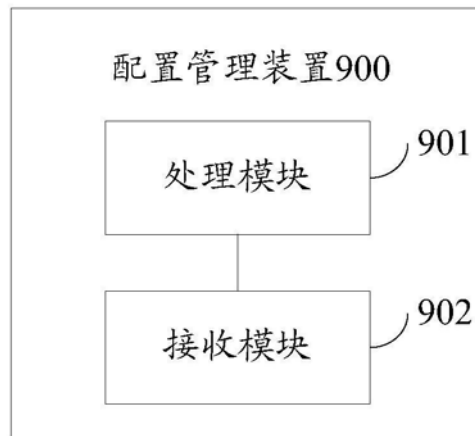


图9