



(12) 发明专利

(10) 授权公告号 CN 111865781 B

(45) 授权公告日 2022.10.11

(21) 申请号 201910337917.3

(22) 申请日 2019.04.25

(65) 同一申请的已公布的文献号  
申请公布号 CN 111865781 A

(43) 申请公布日 2020.10.30

(73) 专利权人 伊姆西IP控股有限责任公司  
地址 美国马萨诸塞州

(72) 发明人 应治 王鲲 吴鹏飞 刘金鹏

(74) 专利代理机构 北京市金杜律师事务所  
11256  
专利代理师 王茂华 林程程

(51) Int. Cl.  
H04L 45/12 (2022.01)  
H04L 45/30 (2022.01)

(56) 对比文件

US 7940756 B1, 2011.05.10  
US 2017286179 A1, 2017.10.05  
US 2015142942 A1, 2015.05.21

审查员 白红昌

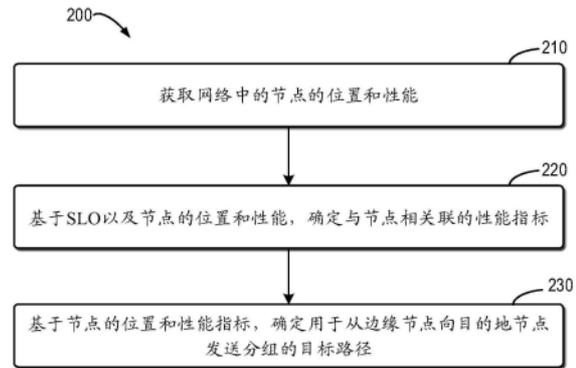
权利要求书3页 说明书10页 附图4页

(54) 发明名称

用于路径优化的方法、设备和计算机程序产品

(57) 摘要

本公开的实施例涉及用于路径优化的方法、设备和计算机程序产品。一种用于路径优化的方法包括：在包含多个节点的网络的边缘节点处，获取网络中的节点的位置和性能；基于服务等级目标 (SLO) 以及节点的位置和性能，确定与节点相关联的性能指标，该性能指标指示节点的性能与SLO之间的差距；以及基于节点的位置和性能指标，确定用于从边缘节点向目的地节点发送分组的目标路径。本公开的实施例能够根据网络环境中的动态变化来实时优化传输分组流量的路径，以此方式可以尽可能地满足端到端服务等级目标。



1. 一种用于路径优化的方法,包括:

在包含多个节点的网络的边缘节点处,基于所述网络中的所述多个节点中的一个或多个节点对来自所述边缘节点的至少一个请求的响应,获取所述一个或多个节点的位置和性能,所述多个节点在所述边缘节点和目的地节点之间;

由所述边缘节点基于服务等级目标SLO以及所述一个或多个节点的位置和性能,确定与所述一个或多个节点相关联的性能指标,所述性能指标指示相应节点的性能与所述SLO之间的差距;以及

由所述边缘节点基于所述一个或多个节点的所述位置和所述性能指标,确定用于从所述边缘节点向所述目的地节点发送分组的目标路径。

2. 根据权利要求1所述的方法,其中获取所述一个或多个节点的位置和性能包括:

向所述网络中的所述一个或多个节点发送探测分组;

接收来自所述一个或多个节点的针对所述探测分组的响应消息;以及

基于所述响应消息,获取所述一个或多个节点的位置和性能。

3. 根据权利要求2所述的方法,还包括以下中的至少一项:

响应于所获取的所述一个或多个节点中的各个节点的位置与该节点的参考位置不同,利用所获取的所述位置来更新所述参考位置;以及

响应于所获取的所述一个或多个节点中的各个节点的性能与该节点的参考性能不同,利用所获取的所述性能来更新所述参考性能。

4. 根据权利要求1-3中任一项所述的方法,其中确定所述目标路径包括:

确定所述边缘节点是否是所述目的地节点;

响应于所述边缘节点不是所述目的地节点,

基于所述网络的所述边缘节点的位置和所述目的地节点,确定所述边缘节点的下一跳节点的候选集合,

基于所述候选集合中的节点的性能指标,确定所述下一跳节点,以及

至少部分地基于所述边缘节点和所述下一跳节点,确定所述目标路径。

5. 根据权利要求4所述的方法,其中至少部分地基于所述边缘节点和所述下一跳节点确定所述目标路径包括:

确定所述下一跳节点是否是所述目的地节点;以及

响应于所述下一跳节点是所述目的地节点,将从所述边缘节点到所述目的地节点之间的路径确定为所述目标路径。

6. 根据权利要求1所述的方法,还包括:

使得所述分组基于所述目标路径而从所述边缘节点向所述目的地节点路由。

7. 根据权利要求6所述的方法,还包括:

在所述分组到达所述目的地节点之后,获取所述目标路径中的节点的性能指标;以及

基于所述目标路径中的节点的性能指标,计算与所述目标路径相关联的性能指标。

8. 根据权利要求1所述的方法,其中所述SLO包括以下中的至少一项:最小带宽、响应时间、往返延时、故障时间、故障频率、可靠性、修复时间、内存占用、队列长度、准确性、完整性以及吞吐量。

9. 根据权利要求1所述的方法,其中所述目的地节点由所述边缘节点设置。

10. 一种电子设备,包括:

至少一个处理单元;

至少一个存储器,所述至少一个存储器被耦合到所述至少一个处理单元并且存储用于由所述至少一个处理单元执行的指令,所述指令当由所述至少一个处理单元执行时,使得所述设备执行动作,所述动作包括:

在包含多个节点的网络的边缘节点处,基于所述网络中的所述多个节点当中的一个或多个节点对来自所述边缘节点的至少一个请求的响应,获取所述一个或多个节点的位置和性能,所述多个节点在所述边缘节点和目的地节点之间;

基于服务等级目标SLO以及所述一个或多个节点的位置和性能,确定与所述一个或多个节点相关联的性能指标,所述性能指标指示相应节点的性能与所述SLO之间的差距;以及

基于所述一个或多个节点的所述位置和所述性能指标,确定用于从所述边缘节点向所述目的地节点发送分组的目标路径。

11. 根据权利要求10所述的设备,其中获取所述一个或多个节点的位置和性能包括:

向所述网络中的所述一个或多个节点发送探测分组;

接收来自所述一个或多个节点的针对所述探测分组的响应消息;以及

基于所述响应消息,获取所述一个或多个节点的位置和性能。

12. 根据权利要求11所述的设备,还包括以下中的至少一项:

响应于所获取的所述一个或多个节点中的各个节点的位置与该节点的参考位置不同,利用所获取的所述位置来更新所述参考位置;以及

响应于所获取的所述一个或多个节点中的各个节点的性能与该节点的参考性能不同,利用所获取的所述性能来更新所述参考性能。

13. 根据权利要求10-12中任一项所述的设备,其中确定所述目标路径包括:

确定所述边缘节点是否是所述目的地节点;

响应于所述边缘节点不是所述目的地节点,

基于所述网络的所述边缘节点的位置和所述目的地节点,确定所述边缘节点的下一跳节点的候选集合,

基于所述候选集合中的节点的性能指标,确定所述下一跳节点,以及

至少部分地基于所述边缘节点和所述下一跳节点,确定所述目标路径。

14. 根据权利要求13所述的设备,其中至少部分地基于所述边缘节点和所述下一跳节点确定所述目标路径包括:

确定所述下一跳节点是否是所述目的地节点;以及

响应于所述下一跳节点是所述目的地节点,将从所述边缘节点到所述目的地节点之间的路径确定为所述目标路径。

15. 根据权利要求10所述的设备,还包括:

使得所述分组基于所述目标路径而从所述边缘节点向所述目的地节点路由。

16. 根据权利要求15所述的设备,还包括:

在所述分组到达所述目的地节点之后,获取所述目标路径中的节点的性能指标;以及

基于所述目标路径中的节点的性能指标,计算与所述目标路径相关联的性能指标。

17. 根据权利要求10所述的设备,其中所述SLO包括以下中的至少一项:最小带宽、响应

时间、往返延时、故障时间、故障频率、可靠性、修复时间、内存占用、队列长度、准确性、完整性以及吞吐量。

18. 根据权利要求10所述的设备,其中所述目的地节点由所述边缘节点设置。

19. 一种计算机存储介质,所述计算机存储介质包括机器可执行指令,所述机器可执行指令在由设备执行时使所述设备执行根据权利要求1-9中的任一项所述的方法。

## 用于路径优化的方法、设备和计算机程序产品

### 技术领域

[0001] 本公开的实施例总体涉及计算机网络领域,具体涉及用于路径优化的方法、设备和计算机程序产品。

### 背景技术

[0002] 在传统网络架构中,当考虑如何将分组从源转发到目的地的问题时,总体上采用诸如路由协议、服务质量(QoS)、资源预留协议(RSVP)、软件定义网络(SDN)等技术。例如,在涉及路由协议的大多数情况下,路由器基于最优尽力策略(Best-effort strategy)来做出关于下一跳的路由决定,这意味着路由器仅仅是将分组转发到所声称的最接近目的地的路由器,而这种最优尽力策略并不一定适合所有类型的网络流量,因而对网络性能没有保障。与QoS有关的技术包括流量分类、排队和整形方法、策略控制方法等等,这些技术基于分组标记或流信息被静态配置在网络设备上,通常不支持应用感知并且缺乏分组流的全貌。

[0003] RSVP是一种由流量的接收者单向配置的传输层协议,同时也是静态配置的路由方法,对终端应用不具有感知能力。因而,RSVP无法在传输过程中根据流量类型进行动态调整网络行为。SDN是一种基于控制器的网络架构。在SDN架构中,尽管控制器能够动态地监控和管理网络,但是控制器也成为整个网络的瓶颈,原因在于:1)每个控制器只能处理有限数量的网络节点和流量,极大地限制了网络规模;2)当控制器发生故障时,将导致整个站点停机;3)网络完全信任控制器,使得网络中的边缘节点无法根据自身需要调整网络行为。

### 发明内容

[0004] 本公开的实施例提供了用于路径优化的方法、设备和计算机程序产品。

[0005] 在本公开的第一方面,提供了一种用于路径优化的方法。该方法包括:在包含多个节点的网络的边缘节点处,获取网络中的节点的位置和性能;基于服务等级目标(SLO)以及节点的位置和性能,确定与节点相关联的性能指标,性能指标指示节点的性能与SLO之间的差距;以及基于节点的位置和性能指标,确定用于从边缘节点向目的地节点发送分组的目标路径。

[0006] 在本公开的第二方面,提供了一种电子设备。该设备包括:至少一个处理单元;至少一个存储器,该至少一个存储器被耦合到至少一个处理单元并且存储用于由至少一个处理单元执行的指令,该指令当由至少一个处理单元执行时,使得设备执行动作,该动作包括:在包含多个节点的网络的边缘节点处,获取网络中的节点的位置和性能;基于服务等级目标(SLO)以及节点的位置和性能,确定与节点相关联的性能指标,该性能指标指示节点的性能与SLO之间的差距;以及基于节点的位置和性能指标,确定用于从边缘节点向目的地节点发送分组的目标路径。

[0007] 在本公开的第三方面,提供了一种计算机程序产品。该计算机程序产品被有形地存储在非瞬态计算机存储介质中并且包括机器可执行指令。该机器可执行指令在由设备执行时使该设备执行根据本公开的第一方面所描述的方法的任意步骤。

[0008] 提供发明内容部分是为了以简化的形式来介绍对概念的选择,它们在下文的具体实施方式中将被进一步描述。发明内容部分无意标识本公开的关键特征或必要特征,也无意限制本公开的范围。

### 附图说明

[0009] 通过结合附图对本公开示例性实施例进行更详细的描述,本公开的上述以及其它目的、特征和优势将变得更加明显,其中,在本公开示例性实施例中,相同的参考标号通常代表相同部件。

[0010] 图1示出了本公开的实施例能够在其中被实施的示例网络的示意图;

[0011] 图2示出了根据本公开的实施例的用于路径优化的方法的流程图;

[0012] 图3示出了根据本公开的实施例的用于路径优化的方法的流程图;

[0013] 图4示出了根据本公开的实施例的用于路径优化的方法的流程图;

[0014] 图5示出了根据本公开的实施例的用于路径优化的学习模型的示意图;

[0015] 图6示出了根据本公开的实施例的用于路径优化的学习模型的验证性测试示意图;

[0016] 图7示出了根据本公开的实施例的电子设备的框图;

[0017] 图8示出了可以用来实施本公开内容的实施例的示例设备的示意性框图。

[0018] 在各个附图中,相同或对应的标号表示相同或对应的部分。

### 具体实施方式

[0019] 下面将参照附图更详细地描述本公开的优选实施例。虽然附图中显示了本公开的优选实施例,然而应该理解,可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反,提供这些实施例是为了使本公开更加透彻和完整,并且能够将本公开的范围完整地传达给本领域的技术人员。

[0020] 在本文中使用的术语“包括”及其变形表示开放性包括,即“包括但不限于”。除非特别申明,术语“或”表示“和/或”。术语“基于”表示“至少部分地基于”。术语“一个示例实施例”和“一个实施例”表示“至少一个示例实施例”。术语“另一实施例”表示“至少一个另外的实施例”。术语“第一”、“第二”等等可以指代不同的或相同的对象。下文还可能包括其他明确的和隐含的定义。

[0021] 如上所述,在传统网络架构中普遍使用的路由技术已经不足以满足下一代网络互连架构的要求。本公开的发明人认识到下一代网络至少应当具有以下属性:

[0022] 1) 面向端到端服务等级目标(SLO):从终端用户和应用的角度来看,诸如SLO等端到端网络性能是最重要的考量,而中间节点之间的性能对它们而言则没有那么重要。

[0023] 2) 应用感知能力:由于现在的应用,例如HTTP流量较为复杂,并且许多不同的应用和网页都在相同端口处运行。因而,使用端口号和IP地址已经不足以区分流量类型。与此同时,不同的应用具有不同的网络要求和SLO。例如,一些应用对延时敏感,而另一些应用对带宽敏感。因而,网络应当具有以不同方式对待这些应用的能力,即应用感知能力。

[0024] 3) 动态智能:网络应当利用人工智能技术从随着时间而变化的网络环境中进行学习以及处置不确定的网络环境。

[0025] 4) 去中心化:如上所述,对于诸如SDN等基于控制器的网络架构,其控制器已经成为整个网络的瓶颈,因而期望利用去中心化的架构来解决诸如SDN等中心化架构所面临的控制器性能瓶颈和单点故障等问题。

[0026] 为了实现上述属性并赶上不断演进的技术和要求,本公开的示例实施例提出了一种动态智能的去中心化网络架构,使得能够从终端用户和应用的角度来量化在给定时间内对服务执行情况的期望,从而尽可能满足应用级的端到端SL0。此外,根据本公开的实施例所提出的网络架构还具备应用感知能力。

[0027] 图1示出了本公开的实施例能够在其中被实现的示例网络100的框图。如图1所示,网络100包括多个节点110、120-1至120-5以及130,其中节点110为边缘节点,节点130为目的地节点,并且节点120-1至120-5为边缘节点110与目的地节点130之间的中间节点。

[0028] 在本公开的上下文中,边缘节点是指在网络互连架构中部署在终端用户侧或其附近的网络节点,这样的网络节点对最终接入用户具有较好的响应能力和连接速度。而另一方面,这样的网络节点通常不能控制对网络流量的处理。

[0029] 应当理解,仅出于示例性的目的描述网络100的结构,而不暗示对于本公开的范围的任何限制。例如,网络100可以包括任意数量、任意类型的节点,并且本公开的实施例还可以被应用到与网络100不同的网络中。在一些实施例中,边缘节点110可以是网络互连架构中的任意末端节点。在一些实施例中,边缘节点110与目的地节点130之间的中间节点不限于所示的5个节点120-1至120-5,而是可以包括任何适当数目的中间节点。

[0030] 在本公开的实施例中,网络100中的各个节点110、120-1至120-5以及130之间存在如由图1中的实线所示的网络流量,以及如由图1中的虚线所示的探测流量。在一些实施例中,网络流量由在边缘节点110上运行的应用产生。探测流量例如可以包括由节点110基于隧道协议而向其他节点120-1至120-5以及130发送的探测分组,以用于获取节点120-1至120-5以及130的位置和性能。

[0031] 传统上,当要从边缘节点110向目的地节点130发送分组时,例如,要将应用所产生的网络流量转发至目的地节点130,可以存在许多可能的路径。如果采用最短路径策略或者最优尽力策略,则可以确定最短路径为从边缘节点110经由节点120-2和120-5到目的地节点130的路径102。这意味着在边缘节点110处运行的所有应用所产生的网络流量都将经由路径102被转发至目的地节点130。

[0032] 然而,如上面所提及的,在边缘节点110处同时运行的不同应用可能分别具有不同的网络要求和SL0。因而,针对这些应用所产生的不同类型的分组采用相同的路由策略是不合适的。考虑以下场景,假设从节点120-2直接到节点120-5的链路具有高延时,而从路径120-2经由节点120-3到节点120-5的链路具有低延时。如果要发送的是对延时敏感的应用所产生的分组,即感兴趣的SL0为低延时,则对于这样的分组,相比于最短路径102,显然选择从节点110经由节点120-2、120-3、120-5到节点130的路径103更为有利。

[0033] 本公开的实施例提供了一种用于路径优化的方案,以解决上述问题和其他潜在问题中的一个或多个。该方案能够将路径优化的中心决策权下放到边缘节点侧,实现动态智能的路由策略,从而满足边缘节点处运行的许多应用及其产生的网络流量对不同SL0及网络性能的需求。

[0034] 图2示出了根据本公开的实施例的用于路径优化的方法200的流程图。例如,方法

200可以在如图1所示的边缘节点110处执行。应当理解,方法200还可以包括未示出的附加动作和/或可以省略所示出的动作,本公开的范围在此方面不受限制。以下结合图1所示的网络来描述方法200。

[0035] 在框210处,边缘节点110获取网络中的节点的位置和性能。

[0036] 在一些实施例中,边缘节点110可以获取网络100中的节点120-1至120-5以及目的地节点130的位置和性能。作为示例,节点的位置包括节点在网络拓扑中的物理位置和逻辑位置。节点的性能可以包括,但不限于节点的当前最小带宽、响应时间、往返延时、故障时间、故障频率、可靠性、修复时间、内存占用、队列长度、准确性、完整性以及吞吐量等等。

[0037] 图3示出了根据本公开的实施例的用于路径优化的示例方法300的流程图。方法300可以作为如图2中的框210所述的步骤的一种实现方式,并且该实现方式是示例性的,而非限制性的。如图3所示,在框310处,边缘节点110向网络中的节点,例如节点120-1至120-5以及目的地节点130发送探测分组。

[0038] 在一些实施例中,边缘节点110可以被部署有探测组件,并由探测组件向网络100中的节点120-1至120-5以及130发送探测分组以完整探测整个网络100,探测分组可以例如是图1中由虚线所示的探测流量。在一些实施例中,可以使用诸如标签-堆栈隧道等源路由技术来将探测分组发送到网络中的特定节点或者跟随特定的路径。

[0039] 在框320处,边缘节点110接收来自网络中的节点的针对探测分组的响应消息。例如,由边缘节点110的探测组件接收来自网络100中的节点120-1至120-5以及130的针对探测分组的响应消息,该响应消息指示相应节点的位置和实时性能。

[0040] 然后,在框330处,边缘节点110基于所接收的响应消息,获取相应的节点的位置和性能。

[0041] 在一些实施例中,边缘节点110还可以响应于所获取的节点的位置与该节点的参考位置不同,利用所获取的位置来更新该节点的参考位置。

[0042] 在这样的实施例中,边缘节点110可以被部署有存储设备,存储设备被配置为存储网络100中的节点的参考位置和参考性能。如果所获取的节点的位置与存储设备中所存储的该节点的参考位置不同,则边缘节点110利用所获取的位置来更新存储设备中存储的该节点的参考位置。换言之,当网络中的节点的位置不发生变化时,则无需更新存储设备中所存储的参考位置。或者,当网络中出现新节点时,在存储设备中可能没有存储该节点的参考位置。在这样的情况中将新节点的参考位置将被添加并存储在存储设备中。

[0043] 在一些实施例中,边缘节点110还可以响应于所获取的节点的性能与该节点的参考性能不同,利用所获取的性能来更新该节点的参考性能。在这样的实施例中,如果所获取的节点的性能与存储设备中所存储的该节点的参考性能不同,则边缘节点110利用所获取的性能来更新存储设备中存储的该节点的参考性能。换言之,当网络中的节点的性能不发生变化时,则无需更新存储设备中所存储的相应的参考性能。或者,当网络中出现新节点时,在存储设备中可能没有存储该节点的参考性能。在这样的情况中,新节点的参考性能将被添加并存储在存储设备中。

[0044] 在一些实施例中,节点的参考位置和参考性能可以被关联地存储在边缘节点110的存储设备中。

[0045] 返回参考图2。在框220处,边缘节点110基于SLO以及所获取的节点的位置和性能,

确定与节点相关联的性能指标。

[0046] 在一些实施例中,SL0可以由边缘节点110设置,并且可以与在节点上运行的应用相关联。SL0可以包括、但不限于最小带宽、响应时间、往返延时、故障时间、故障频率、可靠性、修复时间、内存占用、队列长度、准确性、完整性以及吞吐量等。例如,在节点上运行的应用产生对延时敏感的网络流量的情形中,可以将SL0设置为往返延时。类似地,在应用对延时不敏感,但是所产生的网络流量需要占用较大带宽的情况下,可以将SL0设置为最小带宽。在一些实施例中,SL0可以是数值或多维向量。在其他实施例中,SL0还可以采用其他任意适当的形式,本公开的范围在此方面不受限制。

[0047] 在一些实施例中,性能指标可以指示节点的性能与SL0之间的差距,从而可以用来评估节点是否符合分组的网络要求。因而,可以根据节点的性能指标来选择哪一或哪些节点可以作为从边缘节点110到目的地节点130的路径中的节点。根据本公开的实施例,可以通过任何适当的方式来确定与节点相关联的性能指标,无论是目前已知的还是将来开发的。而且,性能指标可以具有任何适当的表征形式。

[0048] 然后,在框230处,边缘节点110基于节点的位置和性能指标,确定用于从边缘节点向目的地节点发送分组的目标路径。在一些实施例中,可以利用基于强化学习算法的学习模型来确定目标路径,这将在之后详细讨论。

[0049] 根据本公开的实施例,针对节点处运行的应用的不同网络要求和SL0,可以确定并选择最接近SL0的路径以用于路由分组。以此方式,不仅使得边缘节点具备应用感知能力和对路径优化的决策权,还可以实现对网络的动态配置和去中心化。

[0050] 图4示出了根据本公开的实施例的用于路径优化的示例方法400的流程图。方法400可以作为如图2中的框230所述的步骤的一种实现方式,并且该实现方式是示例性的,而非限制性的。如图4所示,在框410处,边缘节点110确定其是否是目的地节点130。如果不是,则进行至框420,否则,该方法400结束。

[0051] 在框420处,边缘节点110基于边缘节点的位置和目的地节点的位置,确定边缘节点110的下一跳节点的候选集合。例如,在图1所示的网络100中,下一跳节点的候选集合包括节点120-1和节点120-2。

[0052] 在框430处,边缘节点110基于候选集合中的节点的性能指标,确定下一跳节点。作为示例,假设SL0被设置为延时 $\leq 15\text{ms}$ ,由边缘节点110基于探测分组而获取的节点120-1的延时为 $18\text{ms}$ ,并且节点120-2的延时为 $13\text{ms}$ 。进而,边缘节点110将节点120-2确定为下一跳节点。

[0053] 在框440处,边缘节点110至少部分地基于边缘节点和下一跳节点,确定目标路径。例如,在上面的示例中,边缘节点110至少部分地基于边缘节点110和下一跳节点120-2,确定目标路径为如图1所示的路径103。

[0054] 在一些实施例中,框440中所述的步骤的一种示例实现包括:由边缘节点110确定下一跳节点是否是目的地节点;以及响应于下一跳节点是目的地节点,将从边缘节点到目的地节点之间的路径确定为目标路径。

[0055] 在一些实施例中,在确定目标路径之后,边缘节点110使得分组基于目标路径而从边缘节点向目的地节点路由。例如,边缘节点110使得在其上运行的应用所产生的网络流量基于路径103而从边缘节点110向目的地节点130路由。

[0056] 在一些实施例中,在分组到达目的地节点之后,边缘节点110获取目标路径中的节点的性能指标。然后,边缘节点110可以基于目标路径中的节点的性能指标来计算与目标路径相关联的性能指标,以用于评估整条目标路径是否符合SLO及网络要求。根据本发明的实施例,与目标路径相关联的性能指标可以通过任何适当的方式来确定,不论是目前已知的还是将来开发的。而且,与目标路径相关联的性能指标可以具有任意适当的表征形式。例如,可以计算目标路径中的各个节点的性能指标的加权平均值来作为整条路径的性能指标。本公开的范围在此方面不受限制。

[0057] 以此方式,所确定的目标路径中的每个节点都是以尽量接近SLO和尽可能满足网络要求为目标而选择的。此外,根据本公开的实施例,不仅利用与节点相关联的性能指标来评估节点的好坏,还利用与路径相关联的性能指标来评估整条路径的好坏,实现了面向SLO的、更加灵活和全面的路由策略。

[0058] 图5示出了根据本公开的实施例的用于路径优化的学习模型500的示意图。学习模型500可以被实施在边缘节点110处,以用于实现如图2所示的框220-230、如图3所示的框310-330以及如图4所示的框410-440所述的步骤。借助于如图5所示的学习模型500可以将路径优化问题转换为强化学习问题。这样,所确定的目标路径可以是学习模型500基于强化学习算法而执行强化学习过程,并最终收敛得到的结果。下面将结合表1进行描述。

[0059] 表1

学习模型 500 的强化学习过程	
环境	网络 100, 其中包括节点 120-1 至 120-5 以及 130
服务等级目标 (SLO)	最小带宽、响应时间、往返延时、故障时间、故障频率、可靠性、修复时间、内存占用、队列长度、准确性、完整性、吞吐量...
状态 $S$	节点 120-1 至 120-5 以及 130 的位置和性能
动作 $A$	1) 选择下一跳节点并向下一跳节点发送分组(不包括传入节点); 2) 不再继续发送
奖励函数 $R_s^a$	$= \begin{cases} 0 & \text{if } P_{ss}^a = \text{终止} \\ -1 & \text{否则} \end{cases}$ (最短路径路由策略)  $= -\frac{1}{2}  TargetPerf_s - ActualPerf_s ^2$ (基于 SLO 的策略)

[0061] 如表1所示,环境表示学习模型500执行动作时所处的场景。在本公开的示例实施例中,环境可以是网络100。状态S表示从环境返回学习模型500的当前状况,例如,网络100中的节点120-1至120-5以及130的位置和性能。 $A_t$ 表示学习模型500基于强化学习算法而做出的决策,例如边缘节点110基于学习模型500所确定的目标路径而向路径中的下一跳节点发送分组。奖励R表示环境对学习模型500的即时返回值,以用于评估学习模型500所执行的上一个动作。在本公开的示例实施例中,奖励R可以是基于奖励函数 $R_{s,a}$ 而计算的与节点相

关联的性能指标。应当理解,表1中示出的各种部署、函数、参数以及决策仅仅是示例性的,而非限制性的。

[0062] 作为示例,在初始时刻 $t$ ,学习模型500可以从网络100获取节点的初始位置和初始性能,即状态 $S_t$ ,并且状态 $S_t$ 作为学习模型500的输入。学习模型500基于状态 $S_t$ 并利用奖励函数 $R_s a$ 来计算奖励 $R_t$ ,即与节点相关联的性能指标。然后,学习模型500可以至少部分地基于状态 $S_t$ 和奖励 $R_t$ 来确定从边缘节点到目的地节点的目标路径,并且执行动作 $A_t$ 。动作 $A_t$ 可以是例如,向下一跳节点发送分组。应当理解,利用不同的奖励函数 $R_s a$ 可以计算出不同的奖励 $R_t$ 。仅作为示例,表1示出了分别针对最短路径路由策略和基于SL0的路由策略的奖励函数,但是在本公开的其他示例实施例中还可以使用任何适当的奖励函数。本公开的范围在此方面不受限制。

[0063] 由于动作 $A_t$ 的执行,环境的状态 $S$ 会发生改变。例如,在下一时刻 $t+1$ ,状态从 $S_t$ 变为 $S_{t+1}$ 。相应地,奖励也从 $R_t$ 变为 $R_{t+1}$ 。进而,状态 $S_{t+1}$ 和奖励 $R_{t+1}$ 作为学习模型500的输入,并且学习模型500可以基于状态 $S_t$ 和 $S_{t+1}$ 以及动作 $A_t$ 来确定转移函数 $P(S_t, A_t, S_{t+1})$ ,并基于转移函数 $P$ 来确定要执行的下一动作 $A_{t+1}$ 。

[0064] 在网络环境中路由分组期间,学习模型500将基于强化学习算法不断地重复上述强化学习过程并执行迭代计算。从而,可以将学习模型500收敛得到的最优解或局部最优解作为从边缘节点到目的地节点的目标路径。

[0065] 根据本公开的实施例,学习模型500可以基于任何合适的强化学习算法来构建,无论是目前已知的还是将来开发的。本公开的范围在此方面不受限制。

[0066] 图6示出了根据本公开的实施例的用于路径优化的学习模型500的验证性测试示意图。如图6所示,网络的拓扑结构具有100个节点,并且在100个节点之间部署有500条链路,其中部署有学习模型500的边缘节点为节点100,目的地节点为节点199。在图6所示的示例中,SL0被设置为数值30.1,并且可以看出学习模型500最终收敛得到目标路径包括节点100、152、144、172、173、129、123、133、180以及199。

[0067] 应当注意,本公开的实施例还适用于与图6中所示出的不同的网络和SL0,并且可以将更多或更少的节点和链路纳入网络中。而且,本公开的实施例可以利用与图6中所示出的不同的学习模型。本公开的范围在此方面不受限制。

[0068] 在上文中已经参考图1至图6详细描述了根据本公开的方法的示例实现,在下文中将描述相应的装置的实现。

[0069] 图7示出了根据本公开的实施例的电子设备700的框图。如图7所示的电子设备700可以被实施在网络互连架构中的任意末端节点处,例如,图1所示的边缘节点110。电子设备700可以包括处理单元710和存储单元720。

[0070] 处理单元710可以例如是运行应用的任何主机、物理计算机、虚拟机、服务器等等。处理单元710可以向存储单元720发送输入/输出(I/O)请求,例如用于从存储单元720读取数据和/向存储单元720写入数据等。响应于接收到来自处理单元710的读请求,存储单元720读取数据,并且将所述读取的数据返回给处理单元710。响应于接收到来自处理单元710的写请求,存储单元720可以写入数据。存储单元720可以是任何目前已知的或者将来开发的非易失性存储介质,诸如磁盘、固态硬盘(SSD)或磁盘阵列等。

[0071] 在一些实施例中,处理单元710可以包括探测组件712和学习组件714。应当注意,

探测组件712也可以独立于处理单元710而作为分离的组件。而且,处理单元710还可以包括其他未示出的附加组件。本公开的范围在此方面不受限制。

[0072] 探测组件712可以进一步包括探测分组生成器,其被配置为生成探测分组,以用于探测网络和采集网络中的节点的位置和实时性能。

[0073] 探测组件712可以向网络中的节点发送探测分组,以及接收来自节点的针对探测分组的响应消息,如图3中的框310-320所述。在一些实施例中,探测组件712可以基于隧道协议、互联网控制报文协议(ICMP)或者定制的遥感协议来发送探测分组。

[0074] 探测组件712基于响应消息来获取节点的位置和性能,如图2中的框210以及图3中的框330所述。

[0075] 在一些实施例中,探测组件712通过电子设备700的环境改变应用程序接口(API) 716与学习组件714交互,以将所获取的位置和性能提供给学习组件714。

[0076] 学习组件714可以包括学习模型,诸如由图5所示的学习模型500。学习组件714可以被配置为利用强化学习算法来训练学习模型500。

[0077] 在一些实施例中,学习组件714可以基于节点110处所设置的SLO以及从探测组件712获取的节点的位置和性能来确定与网络中的节点相关联的性能指标,如图2中的框220所述。

[0078] 在一些实施例中,学习组件714还可以被配置为基于节点的位置和性能指标,确定用于从边缘节点向目的地节点发送分组的目标路径,如图2中的框230以及图4中的框410-440所述。作为示例,基于所确定的目标路径,边缘节点110可以使用源路由策略来在网络环境内路由边缘节点110处运行的应用所产生的网络流量。

[0079] 在一些实施例中,处理单元710还被配置为基于由学习组件714所确定的目标路径,将分组从边缘节点110向目的地节点130路由。

[0080] 在一些实施例中,学习组件714还被配置为在分组到达目的地节点130之后,获取目标路径中的节点的性能指标,以及基于目标路径中的节点的性能指标,计算与目标路径相关联的性能指标。

[0081] 在一些实施例中,学习组件714的学习模型500可以通过强化学习API 718来与强化学习算法交互,从而执行强化学习过程。

[0082] 在一些实施例中,存储单元720可以被配置为存储探测组件714所获取的节点的位置和性能、节点的参考位置和参考性能、由学习组件714所确定的与节点相关联的性能指标、与目标路径相关联的性能指标等。

[0083] 根据本公开的实施例,环境改变API 716和强化学习API 718可以是适于与任何合适的软件或硬件交互的标准API。在一些实施例中,强化学习API可以包括、但不限于env.reset()、env.step(action)、env.change\_time()等函数。在一些实施例中,环境改变API可以包括但不限于env.add\_node()、env.update\_link()等函数。但是,应当理解,上述API仅仅是示例,电子设备700还可以包括其他任意合适的标准API。本公开的范围在此方面不受限制。

[0084] 根据本公开的实施例,提供了用于路径优化的电子设备,该电子设备利用强化学习过程来优化用于路由分组的路径,并且可以制定出面向端到端SLO的路由策略。以此方式,边缘节点侧也可以具有路径优化的决策权,满足了边缘节点处运行的许多应用及其产

生的网络流量对不同SL0及网络性能的需求。此外,由于根据本公开的实施例的电子设备提供了标准API的集合,使得电子设备可以与任何适当的强化学习算法和学习模型进行交互,从而实现了灵活、有效的路径优化。

[0085] 图8示出了可以用来实施本公开内容的实施例的示例设备800的示意性框图。例如,如图7所示的电子设备700可以由设备800实施。如图8所示,设备800包括中央处理单元(CPU)801,其可以根据存储在只读存储器(ROM)802中的计算机程序指令或者从存储单元808加载到随机访问存储器(RAM)803中的计算机程序指令,来执行各种适当的动作和处理。在RAM 803中,还可存储设备800操作所需的各种程序和数据。CPU 801、ROM 802以及RAM 803通过总线804彼此相连。输入/输出(I/O)接口805也连接至总线804。

[0086] 设备800中的多个部件连接至I/O接口805,包括:输入单元806,例如键盘、鼠标等;输出单元807,例如各种类型的显示器、扬声器等;存储单元808,例如磁盘、光盘等;以及通信单元809,例如网卡、调制解调器、无线通信收发机等。通信单元809允许设备800通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据。

[0087] 上文所描述的各个过程和处理,例如方法200、300和400可以由处理单元801执行。例如,在一些实施例中,方法200、300和400可以被实现为计算机软件程序,其被有形地包含于机器可读介质,例如存储单元808。在一些实施例中,计算机程序的部分或者全部可以由ROM 802和/或通信单元809而被载入和/或安装到设备800上。当计算机程序被加载到RAM 803并由CPU 801执行时,可以执行上文描述的方法200、300和400的一个或多个动作。

[0088] 本公开可以是方法、装置、系统和/或计算机程序产品。计算机程序产品可以包括计算机可读存储介质,其上载有用于执行本公开的各个方面的计算机可读程序指令。

[0089] 计算机可读存储介质可以是保持和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质例如可以是——但不限于——电存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或者上述的任意合适的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式压缩盘只读存储器(CD-ROM)、数字多功能盘(DVD)、记忆棒、软盘、机械编码设备、例如其上存储有指令的打孔卡或凹槽内凸起结构、以及上述的任意合适的组合。这里所使用的计算机可读存储介质不被解释为瞬时信号本身,诸如无线电波或者其他自由传播的电磁波、通过波导或其他传输媒介传播的电磁波(例如,通过光纤电缆的光脉冲)、或者通过电线传输的电信号。

[0090] 这里所描述的计算机可读程序指令可以从计算机可读存储介质下载到各个计算/处理设备,或者通过网络、例如因特网、局域网、广域网和/或无线网下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光纤传输、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或者网络接口从网络接收计算机可读程序指令,并转发该计算机可读程序指令,以供存储在各个计算/处理设备中的计算机可读存储介质中。

[0091] 用于执行本公开操作的计算机程序指令可以是汇编指令、指令集架构(ISA)指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、或者以一种或多种编程语言的任意组合编写的源代码或目标代码,所述编程语言包括面向对象的编程语言——诸如

Smalltalk、C++等,以及常规的过程式编程语言—诸如“C”语言或类似的编程语言。计算机可读程序指令可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络—包括局域网(LAN)或广域网(WAN)—连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。在一些实施例中,通过利用计算机可读程序指令的状态信息来个性化定制电子电路,例如可编程逻辑电路、现场可编程门阵列(FPGA)或可编程逻辑阵列(PLA),该电子电路可以执行计算机可读程序指令,从而实现本公开的各个方面。

[0092] 这里参照根据本公开实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述了本公开的各个方面。应当理解,流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合,都可以由计算机可读程序指令实现。

[0093] 这些计算机可读程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理单元,从而生产出一种机器,使得这些指令在通过计算机或其它可编程数据处理装置的处理单元执行时,产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中,这些指令使得计算机、可编程数据处理装置和/或其他设备以特定方式工作,从而,存储有指令的计算机可读介质则包括一个制品,其包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的各个方面的指令。

[0094] 也可以把计算机可读程序指令加载到计算机、其它可编程数据处理装置、或其它设备上,使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤,以产生计算机实现的过程,从而使得在计算机、其它可编程数据处理装置、或其它设备上执行的指令实现流程图和/或框图中的一个或多个方框中规定的功能/动作。

[0095] 附图中的流程图和框图显示了根据本公开的多个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或指令的一部分,所述模块、程序段或指令的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或动作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0096] 以上已经描述了本公开的各实施例,上述说明是示例性的,并非穷尽性的,并且也不限于所披露的各实施例。在不偏离所说明的各实施例的范围和精神的情况下,对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择,旨在最好地解释各实施例的原理、实际应用或对市场中的技术的技术改进,或者使本技术领域的其它普通技术人员能理解本文披露的各实施例。

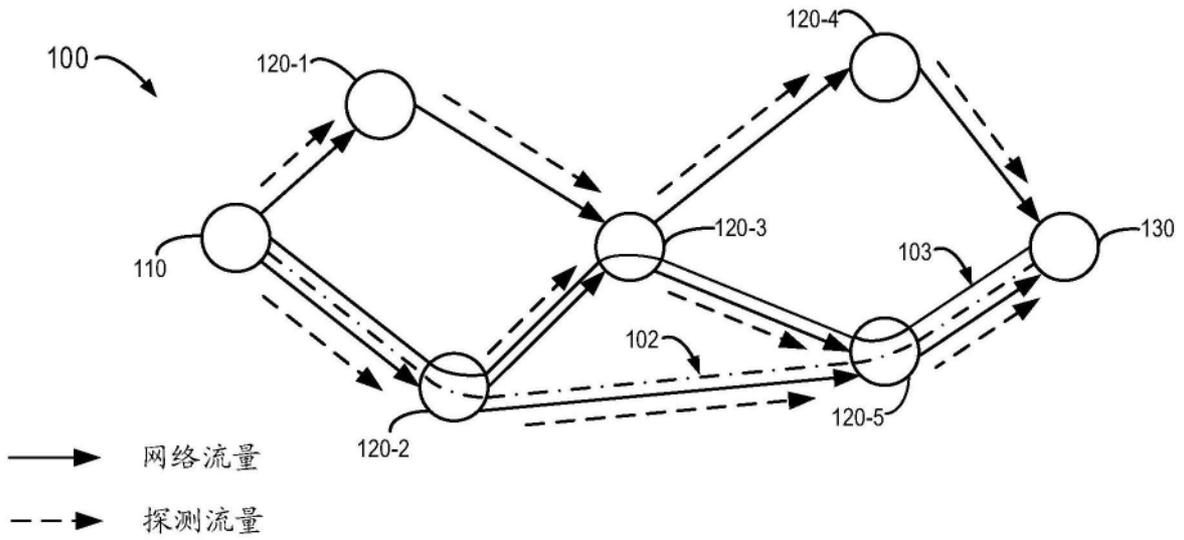


图1

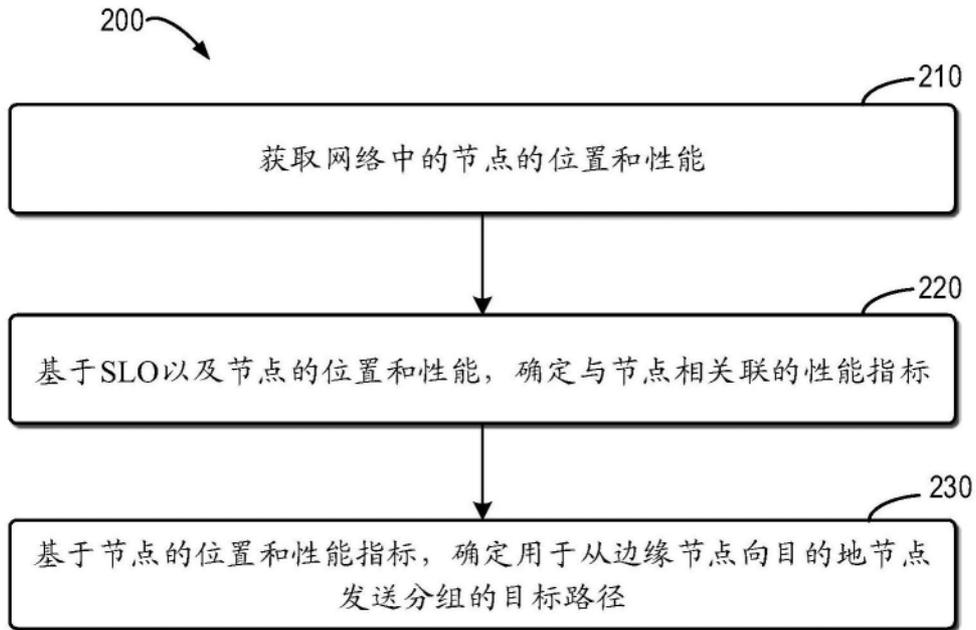


图2

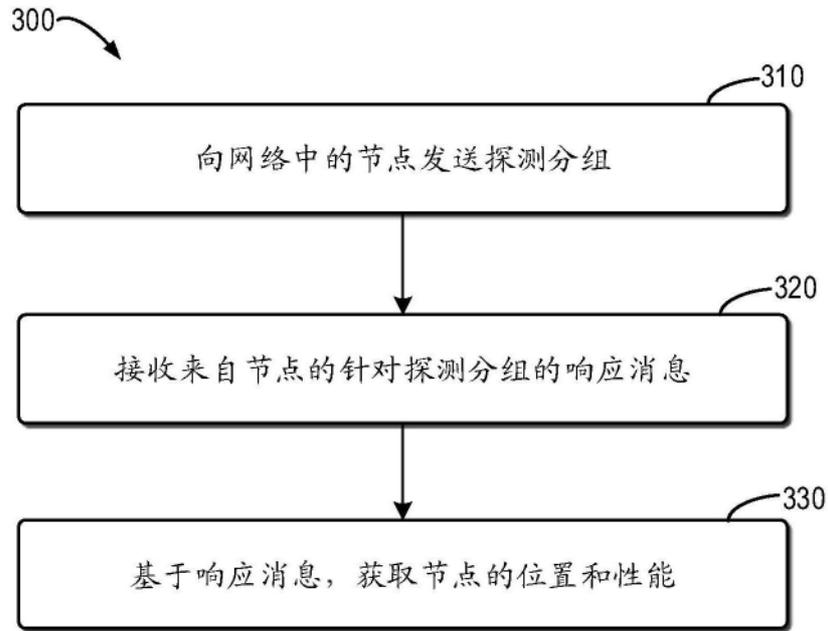


图3

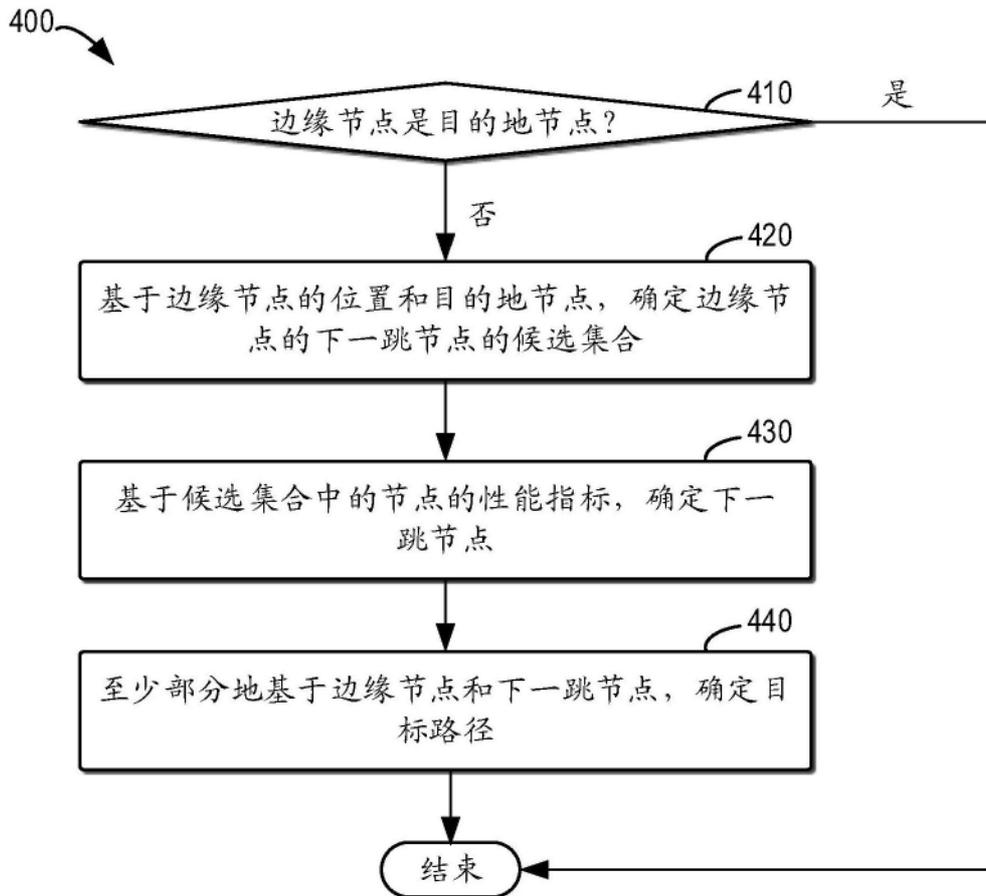


图4

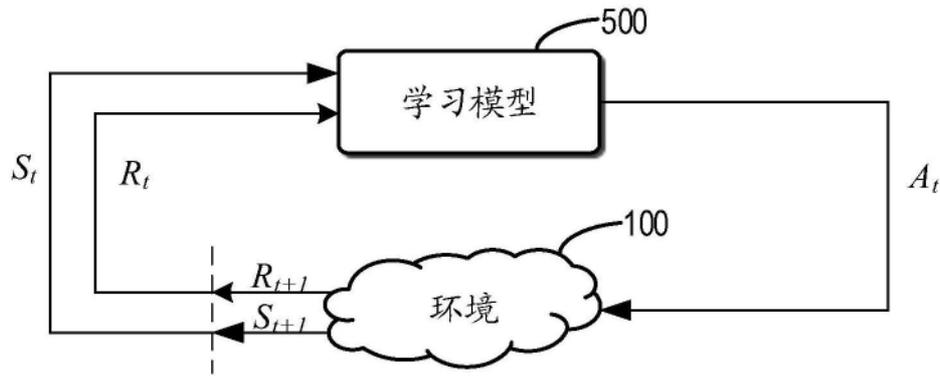


图5

```
SLOs:  
30.1  
State:  
state: 100 total_rtl: 0  
Final Episode:  
state: (152, 0) action: 3 tc: 2 reward: -50.0  
state: (144, 0) action: 5 tc: 2 reward: -100.0  
state: (172, 0) action: 5 tc: 0 reward: -110.0  
state: (173, 0) action: 3 tc: 0 reward: -120.0  
state: (129, 0) action: 0 tc: 0 reward: -130.0  
state: (123, 0) action: 0 tc: 2 reward: -180.0  
state: (133, 0) action: 2 tc: 2 reward: -230.0  
state: (180, 0) action: 5 tc: 1 reward: -260.0  
state: (199, 0) action: 2 tc: 1 reward: -519.9
```

图6

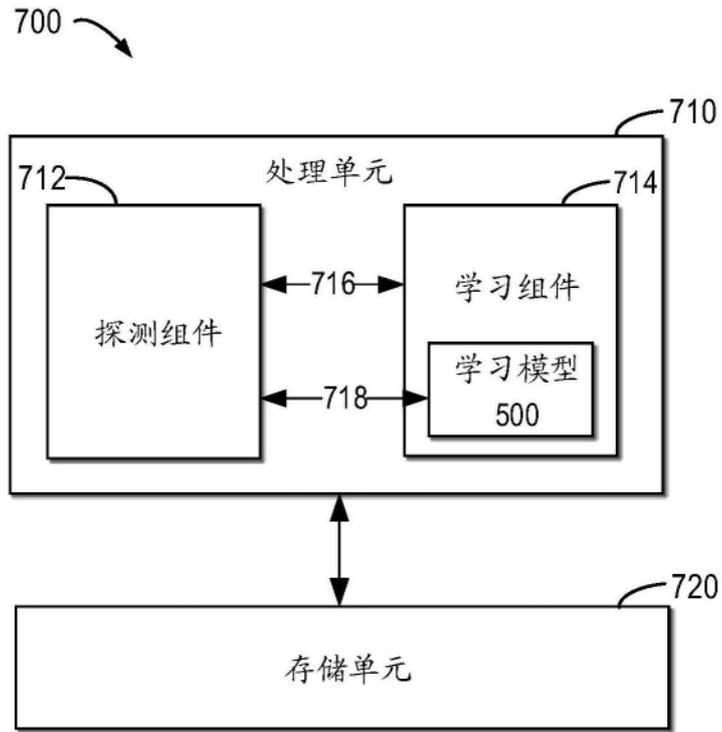


图7

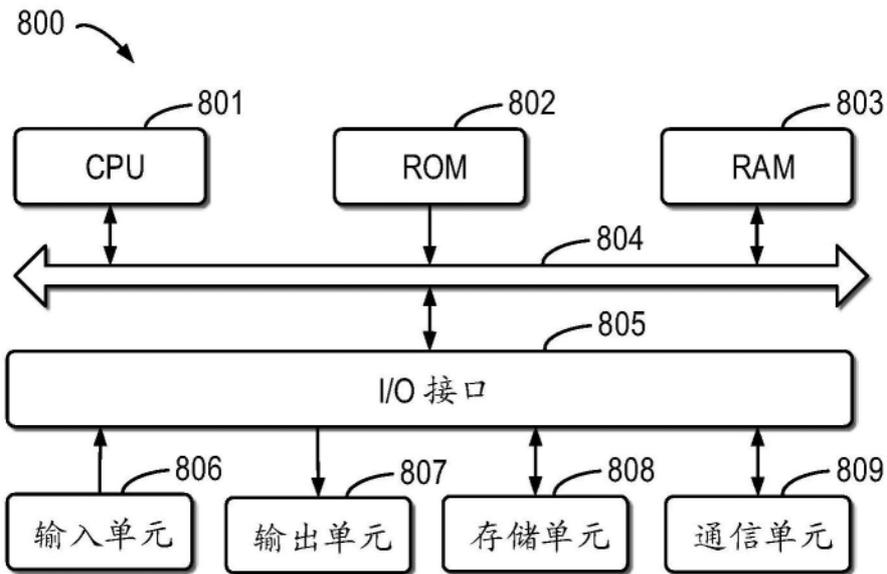


图8