



(12)发明专利申请

(10)申请公布号 CN 110365568 A
(43)申请公布日 2019.10.22

(21)申请号 201910527463.6

(22)申请日 2019.06.18

(71)申请人 西安交通大学

地址 710049 陕西省西安市咸宁西路28号

(72)发明人 曲桦 赵季红 李明霞 石亚娟
王娇 边江

(74)专利代理机构 西安通大专利代理有限责任
公司 61200

代理人 安彦彦

(51)Int.Cl.

H04L 12/46(2006.01)

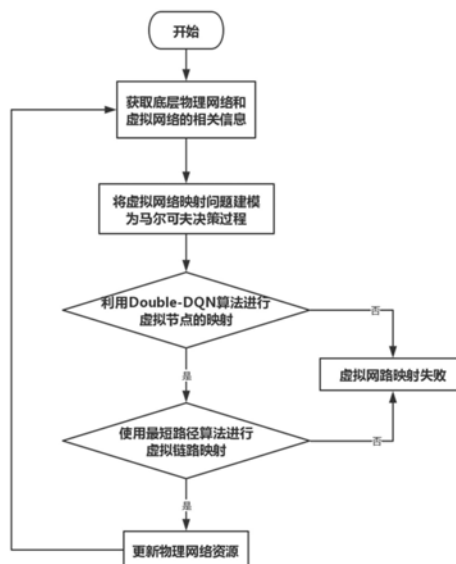
权利要求书2页 说明书6页 附图1页

(54)发明名称

一种基于深度强化学习的虚拟网络映射方法

(57)摘要

一种基于深度强化学习的虚拟网络映射方法,包括以下步骤:步骤1,将虚拟网络映射中节点映射问题建模为马尔可夫决策过程;步骤2,在马尔可夫决策过程基础上,利用DDQN算法进行虚拟节点的映射;步骤3,利用最短路径算法进行虚拟网络链路映射;步骤4,更新物理网络资源,包括CPU资源和链路带宽资源。通过这种自适应学习方案,可以通过节约能耗和提高VNR接收率来获得全局最优的映射方法,比传统方法相比,有更好的灵活性。实验表明,该方法能够降低能耗,提高请求接受率并提高长期平均收益。



1. 一种基于深度强化学习的虚拟网络映射方法,其特征在于,包括以下步骤:

步骤1,将虚拟网络映射中节点映射问题建模为马尔可夫决策过程;

步骤2,在马尔可夫决策过程基础上,利用DDQN算法进行虚拟节点的映射;

步骤3,利用最短路径算法进行虚拟网络链路映射;

步骤4,更新物理网络资源,包括CPU资源和链路带宽资源。

2. 根据权利要求1所述的基于深度强化学习的虚拟网络映射方法,其特征在于,步骤1的具体过程如下:

2.1 定义状态集合 S_t

假设在给定状态 S_t 中,代理选择一个物理节点 $n^s \in N^{\psi_s}$ 映射到虚拟节点 $n^v \in N^{\psi_v}$,然后进入下一个状态 S_{t+1} ,其中 N^{ψ_s} 是能够承载虚拟节点 n^v 的所有物理节点的集合, N^{ψ_v} 是所有未被映射的虚拟节点的集合; t 时刻状态集合被定义为:

$$S_t = \left((N_t^{\psi_v} = N_{t-1}^{\psi_v} \setminus \{n_{t-1}^v\}), (N_t^{\psi_s} = N_{t-1}^{\psi_s} \setminus \{n_{t-1}^s\}) \right)$$

其中 n_{t-1}^s 是承载先前虚拟节点 n_{t-1}^v 的物理节点,在初始状态,没有节点被映射,故 $N_1^{\psi_v} = N^{\psi_v}, N_1^{\psi_s} = N^{\psi_s}$;

2.2 定义动作集合 A_t

代理选择节点 $n_t^s \in \{N_t^{\psi_s} \cap N_t^{\psi_s}(n_t^v)\}$ 的动作 A_t 按如下方式定义:

$$A_t = \{\varepsilon\} \cup \left\{ (n_t^v - n_t^s) : n_t^s \in \{N_t^{\psi_s} \cap N_t^{\psi_s}(n_t^v)\} \right\}$$

其中 ε 表示能够到达终端状态的任意动作;当代理选择当前虚拟节点 n_t^v 的物理节点 n_t^s 时,它转换到下一个状态 S_{t+1} ;

2.3 定义状态转移概率矩阵 P_t

当处于状态 S_t 时,代理选择节点 $n_t^s \in \{N_t^{\psi_s} \cap N_t^{\psi_s}(n_t^v)\}$ 的动作 A_t 转换到下一个状态 S_{t+1} 的状态转移可能性被定义为:

$$P_t(S_{t+1} | n_t^v - n_t^s, S_t) = 1$$

2.4 定义回报函数 R

$$R_j = \begin{cases} R_j, & i_{S_{end_j}} \text{ is true} \\ R_j + \gamma Q'(\phi(S_j'), \underset{a'}{\operatorname{argmax}} Q(\phi(S_j'), a; \theta); \theta'), & i_{S_{end_j}} \text{ is false} \end{cases}$$

其中, j 表示第 j 个样本, γ 为衰减因子, Q 为当前 Q 网络, Q' 为目标 Q 网络, a' 为在当前 Q 网络中最大 Q 值对应的动作。

3. 根据权利要求2所述的基于深度强化学习的虚拟网络映射方法,其特征在于,步骤2的具体过程如下:

3.1 随机初始化所有状态和动作对应的价值 Q ,随机初始化当前 Q 网络的所有参数 θ ,初始化目标 Q 网络的参数 $\theta' = \theta$,清空经验回放集合 D ;

3.2 初始化状态集合 S 为当前状态的第一个状态,得到状态集合 S 的特征向量,并用 $\phi(S)$

表示;

3.3在当前Q网络中使用状态集合S的特征向量 $\phi(S)$ 作为输入,得到当前Q网络的所有动作对应的Q值输出,在当前Q值输出中选择对应的动作A;

3.4在状态S执行当前动作A,得到新状态 S' 、特征向量 $\phi(S')$ 、奖励R以及终止状态标志is_end;

3.5将 $\{\phi(S), A, R, \phi(S'), is_end\}$ 这五元组存入经验回放集合D;

3.6令初始状态 $S=S'$;

3.7从经验回放集合D中采集m个样本 $\{\phi(S_j), A_j, R_j, \phi(S'_j), is_end_j\}, j = 1, 2 \dots m$,计算当前价值Q值 y_j :

$$y_j = \begin{cases} R_j, & is_end_j \text{ is true} \\ R_j + \gamma Q'(\phi(S'_j), \underset{a'}{\operatorname{argmax}} Q(\phi(S'_j), a; \theta); \theta'), & is_end_j \text{ is false} \end{cases}$$

3.8使用均方差损失函数 $\frac{1}{m} \sum_{j=1}^m (y_j - Q(\phi(S_j), A_j, \theta))^2$,更新当前Q网络的所有参数 θ ;

3.9重复步骤3.1-步骤3.8多次,如果重复次数达到目标网络的更新频率C,则更新目标网络参数 $\theta' = \theta$;如果重复次数没有达到目标网络的更新频率C,则继续重复步骤3.1-步骤3.8;

3.10判断终止状态标志is_end是否为终止状态;如果终止状态标志is_end是终止状态,则进行步骤4,否则转到步骤3.2。

4.根据权利要求3所述的基于深度强化学习的虚拟网络映射方法,其特征在于,步骤3.3中,使用 ϵ -贪婪法在当前Q值输出中选择对应的动作A。

5.根据权利要求3所述的基于深度强化学习的虚拟网络映射方法,其特征在于,步骤3.8中,通过神经网络的梯度反向传播更新当前Q网络的所有参数 θ 。

6.根据权利要求3所述的基于深度强化学习的虚拟网络映射方法,其特征在于,步骤3的具体过程如下:

4.1对于虚拟链路 l_{uw} ,首先排除 $ReqBWL(l_{uw}) > BWL(l_{mn})$ 的物理链路;其中, $ReqBWL(l_{uw})$ 为虚拟链路 l_{uw} 请求的带宽, $BWL(l_{mn})$ 为物理链路 l_{mn} 的剩余可用带宽;

4.2初始化计数变量count=0;

4.3对于连接两个终端节点 u^v, w^v 的虚拟链路 l_{uw} ,首先找到虚拟链路 l_{uw} 终端映射的物理节点 m^s, n^s ,然后寻找 m^s, n^s 间的最短路径 p^s ;

4.4若 m^s, n^s 间的最短路径 p^s 存在,则技术变量count++,继续执行步骤4.1-步骤4.3,直到技术变量count等于虚拟链路总数时执行步骤5,否则输出虚拟网络映射失败。

7.根据权利要求6所述的基于深度强化学习的虚拟网络映射方法,其特征在于,步骤4.3中,使用Floyd算法寻找 m^s, n^s 间的最短路径 p^s 。

一种基于深度强化学习的虚拟网络映射方法

技术领域

[0001] 本发明涉及软件定义网络中虚拟网络的映射问题,特别涉及一种基于深度强化学习的虚拟网络映射方法。

背景技术

[0002] 随着云计算、移动互联网等技术的快速发展,多租户网络的需求越来越多和灵活,传统的以IP为核心的基础框架存在着扩展能力差和核心功能单一等问题,不能满足多租户网络的业务需求。

[0003] 网络虚拟化技术是解决上述问题的有效方法,它能融合现有的通信和计算资源虚拟化技术,采用分层的技术手段解决此问题,是未来互联网应具备的关键特性之一。网络虚拟化技术使用解耦合和复用的方法使底层网络的物理资源被多个虚拟网络共享,为共享的异构上层网络提供差异化服务。可自组织、重构的上层虚拟网络能够动态的映射到底层物理网络之上,实现快速的异构网络部署。

[0004] 但是在现有环境下实现网络虚拟化存在很大的局限性,于是一种新兴的网络体系架构—SDN应运而生。它将控制与转发分离,并且实现了可编程的集中控制,支持网络虚拟化,二者相辅相成,是两种具有高度相关性的技术的有效结合,可以很好的满足未来网络对组网的灵活性,网络的集中管理等需求。

[0005] 虚拟网络映射或者虚拟网络嵌入是实现网络虚拟化的一项重要挑战,它能够实现将底层网络资源分配给虚拟网络,即在节点上分配CPU资源,在链路上分配带宽资源。VNE是一个NP难问题,即使所有的虚拟节点已被映射后,映射带有带宽资源约束的虚拟链路仍然是NP难的。面向软件定义网络的虚拟网络映射是对底层网络设备的虚拟化,使其可以支持更多的可扩展应用,同一个物理基础设施可以运行多样的虚拟网络。

[0006] 由于传统的深度强化学习无法解决Q学习算法的固有缺点-过优估计,过优估计指的是估计的值函数比真实的值函数偏大,如果过优估计在所有状态都是均匀的,那么根据贪心策略,依然能够找到值函数的最大动作,但是往往过优估计在各个状态不是均匀的,因此过优估计会影响到策略决策,从而导致获取的不是最优策略。

发明内容

[0007] 本发明的目的在于针对软件定义网络中虚拟网络映射问题,提出一种基于深度强化学习的虚拟网络映射方法。

[0008] 为了达到上述目的,本发明采用了以下技术方案:

[0009] 一种基于深度强化学习的虚拟网络映射方法,包括以下步骤:

[0010] 步骤1,将虚拟网络映射中节点映射问题建模为马尔可夫决策过程;

[0011] 步骤2,在马尔可夫决策过程基础上,利用DDQN算法进行虚拟节点的映射;

[0012] 步骤3,利用最短路径算法进行虚拟网络链路映射;

[0013] 步骤4,更新物理网络资源,包括CPU资源和链路带宽资源。

[0014] 本发明进一步的改进在于,步骤1的具体过程如下:

[0015] 2.1定义状态集合 S_t

[0016] 假设在给定状态 S_t 中,代理选择一个物理节点 $n^s \in N^{\psi_s}$ 映射到虚拟节点 $n^v \in N^{\psi_v}$,然后进入下一个状态 S_{t+1} ,其中 N^{ψ_s} 是能够承载虚拟节点 n^v 的所有物理节点的集合, N^{ψ_v} 是所有未被映射的虚拟节点的集合;t时刻状态集合被定义为:

$$[0017] \quad S_t = \left((N_t^{\psi_v} = N_{t-1}^{\psi_v} \setminus \{n_{t-1}^v\}), (N_t^{\psi_s} = N_{t-1}^{\psi_s} \setminus \{n_{t-1}^s\}) \right)$$

[0018] 其中 n_{t-1}^s 是承载先前虚拟节点 n_{t-1}^v 的物理节点,在初始状态,没有节点被映射,故 $N_1^{\psi_v} = N^{\psi_v}$, $N_1^{\psi_s} = N^{\psi_s}$;

[0019] 2.2定义动作集合 A_t

[0020] 代理选择节点 $n_t^s \in \{N_t^{\psi_s} \cap N_t^{\psi_s}(n_t^v)\}$ 的动作 A_t 按如下方式定义:

$$[0021] \quad A_t = \{\varepsilon\} \cup \left\{ (n_t^v - n_t^s) : n_t^s \in \{N_t^{\psi_s} \cap N_t^{\psi_s}(n_t^v)\} \right\}$$

[0022] 其中 ε 表示能够到达终端状态的任意动作;当代理选择当前虚拟节点 n_t^v 的物理节点 n_t^s 时,它转换到下一个状态 S_{t+1} ;

[0023] 2.3定义状态转移概率矩阵 P_r

[0024] 当处于状态 S_t 时,代理选择节点 $n_t^s \in \{N_t^{\psi_s} \cap N_t^{\psi_s}(n_t^v)\}$ 的动作 A_t 转换到下一个状态 S_{t+1} 的状态转移可能性被定义为:

$$[0025] \quad P_r(S_{t+1}|n_t^v - n_t^s, S_t) = 1$$

[0026] 2.4定义回报函数R

$$[0027] \quad R_j = \begin{cases} R_j, & iS_{end_j} \text{ is true} \\ R_j + \gamma Q'(\phi(S_j), \underset{a'}{\operatorname{argmax}} Q(\phi(S_j), a; \theta); \theta'), & iS_{end_j} \text{ is false} \end{cases}$$

[0028] 其中,j表示第j个样本, γ 为衰减因子,Q为当前Q网络, Q' 为目标Q网络, a' 为在当前Q网络中最大Q值对应的动作。

[0029] 本发明进一步的改进在于,步骤2的具体过程如下:

[0030] 3.1随机初始化所有状态和动作对应的价值Q,随机初始化当前Q网络的所有参数 θ ,初始化目标Q网络的参数 $\theta' = \theta$,清空经验回放集合D;

[0031] 3.2初始化状态集合S为当前状态的第一个状态,得到状态集合S的特征向量,并用 $\phi(S)$ 表示;

[0032] 3.3在当前Q网络中使用状态集合S的特征向量 $\phi(S)$ 作为输入,得到当前Q网络的所有动作对应的Q值输出,在当前Q值输出中选择对应的动作A;

[0033] 3.4在状态S执行当前动作A,得到新状态 S' 、特征向量 $\phi(S')$ 、奖励R以及终止状态标志 iS_{end} ;

[0034] 3.5将 $\{\phi(S), A, R, \phi(S'), is_end\}$ 这五元组存入经验回放集合D;

[0035] 3.6令初始状态 $S=S'$;

[0036] 3.7从经验回放集合D中采集m个样本 $\{\phi(S_j), A_j, R_j, \phi(S'_j), is_end_j\}, j = 1, 2 \dots m$, 计算当前价值Q值 y_j :

$$[0037] \quad y_j = \begin{cases} R_j, & is_end_j \text{ is true} \\ R_j + \gamma Q'(\phi(S'_j), \underset{a'}{\operatorname{argmax}} Q(\phi(S'_j), a; \theta); \theta'), & is_end_j \text{ is false} \end{cases}$$

[0038] 3.8使用均方差损失函数 $\frac{1}{m} \sum_{j=1}^m (y_j - Q(\phi(S_j), A_j, \theta))^2$, 更新当前Q网络的所有参数 θ ;

[0039] 3.9重复步骤3.1-步骤3.8多次, 如果重复次数达到目标网络的更新频率C, 则更新目标网络参数 $\theta' = \theta$; 如果重复次数没有达到目标网络的更新频率C, 则继续重复步骤3.1-步骤3.8;

[0040] 3.10判断终止状态标志is_end是否为终止状态; 如果终止状态标志is_end是终止状态, 则进行步骤4, 否则转到步骤3.2。

[0041] 本发明进一步的改进在于, 步骤3.3中, 使用 ϵ -贪婪法在当前Q值输出中选择对应的动作A。

[0042] 本发明进一步的改进在于, 步骤3.8中, 通过神经网络的梯度反向传播更新当前Q网络的所有参数 θ 。

[0043] 本发明进一步的改进在于, 步骤3的具体过程如下:

[0044] 4.1对于虚拟链路 l_{uw} , 首先排除 $\operatorname{ReqBWL}(l_{uw}) > \operatorname{BWL}(l_{mn})$ 的物理链路; 其中, $\operatorname{ReqBWL}(l_{uw})$ 为虚拟链路 l_{uw} 请求的带宽, $\operatorname{BWL}(l_{mn})$ 为物理链路 l_{mn} 的剩余可用带宽;

[0045] 4.2初始化计数变量 $\operatorname{count} = 0$;

[0046] 4.3对于连接两个终端节点 u^v, w^v 的虚拟链路 l_{uw} , 首先找到虚拟链路 l_{uw} 终端映射的物理节点 m^s, n^s , 然后寻找 m^s, n^s 间的最短路径 p^s ;

[0047] 4.4若 m^s, n^s 间的最短路径 p^s 存在, 则技术变量 $\operatorname{count}++$, 继续执行步骤4.1-步骤4.3, 直到技术变量 count 等于虚拟链路总数时执行步骤5, 否则输出虚拟网络映射失败。

[0048] 本发明进一步的改进在于, 步骤4.3中, 使用Floyd算法寻找 m^s, n^s 间的最短路径 p^s 。

[0049] 与现有技术相比, 本发明具有的有益效果: 本发明通过马尔可夫决策过程(MDP模型)获取当前的VNE环境, 在每次映射开始时, 第一个映射的节点被随机传送到某个物理节点, 然后基于DDQN算法获得奖励值R, 更新网络参数, 然后移动到下一个状态 S_{t+1} 。本发明中使用的DDQN算法, 它是一种将Double Q-learning引入到DQN, 即将action的选择和评估解耦开的算法。DDQN中存在两个结构完全相同但是参数却不同的网络, 用当前Q网络中Q值选择 $\underset{a'}{\operatorname{argmax}} Q(\phi(S'_j), a; \theta)$ 对应的action, 在目标Q网络中评估这个action中的Q值。目标Q网络中网络参数不需要迭代更新, 而是每隔一段时间从当前Q网络中复制过来, 实现延时更新, 这样可以减少两个网络的相关性。因为DDQN算法的核心思想是将目标动作的选择和目标动作的评估分别使用不同的值函数来实现, 所以可以有效地解决过优估计问题。通过这种自

适应学习方案,可以通过节约能耗和提高VNR接收率来获得全局最优的映射方法,比传统方法相比,有更好的灵活性。实验表明,该方法能够降低能耗,提高请求接受率并提高长期平均收益。

附图说明

[0050] 图1为本发明中所用DDQN算法的流程图。

具体实施方式

[0051] 下面结合附图1和具体实施方式对本发明进行详细的描述。

[0052] 本发明提出的一种SDN场景下基于深度强化学习的虚拟网络映射方法,具体包括以下步骤:

[0053] 步骤1,获取底层物理网络和虚拟网络的相关信息:

[0054] 衬底网络拓扑使用无向图表示: $G^s = (N^s, L^s, A_N^s, A_L^s)$,其中 N^s 表示底层网络中的节点集合; L^s 表示衬底网络中链路的集合; A_N^s 表示衬底节点的属性集合,即CPU资源等等; A_L^s 表示衬底链路的属性集合,包含带宽资源、时延信息等。衬底网络中所有非闭合环路路径记作 P^s ,衬底节点的剩余容量记为 $R_N(n^s)$,衬底链路的剩余容量记为 $R_L(l^s)$ 。

[0055] 虚拟网络同样使用加权无向图表示: $G^v = (N^v, L^v, C_N^v, C_L^v)$ 其中 N^v 表示虚拟节点的集合, L^v 表示虚拟链路的集合, C_N^v 表示 $\forall n^v \in N^v$ 的CPU资源请求, C_L^v 表示 $\forall l^v \in L^v$ 的链路带宽资源请求。

[0056] VNE问题可描述为: $M(G^v) : (N^v, L^v, C_N^v, C_L^v) \rightarrow (N', L', R_N, R_L)$,其中 $N' \in N^s, P^s \in P'$, R_N, R_L 分别表示分配给虚拟网络请求的节点资源和链路资源。

[0057] 步骤2,将虚拟网络映射中节点映射问题建模为马尔可夫决策过程:

[0058] 2.1定义状态集合 S_t

[0059] 假设在给定状态 S_t 中,代理选择一个物理节点 $n^s \in N^{\psi_s}$ 映射到虚拟节点 $n^v \in N^{\psi_v}$,然后进入下一个状态 S_{t+1} ,其中 N^{ψ_s} 是可以承载虚拟节点 n^v 的所有物理节点的集合, N^{ψ_v} 是所有未被映射的虚拟节点的集合。 t 时刻状态集合被定义为:

$$[0060] \quad S_t = \left((N_t^{\psi_v} = N_{t-1}^{\psi_v} \setminus \{n_{t-1}^v\}), (N_t^{\psi_s} = N_{t-1}^{\psi_s} \setminus \{n_{t-1}^s\}) \right)$$

[0061] 其中 n_{t-1}^s 是承载先前虚拟节点 n_{t-1}^v 的物理节点,在初始状态,没有节点被映射,故 $N_1^{\psi_v} = N^{\psi_v}, N_1^{\psi_s} = N^{\psi_s}$ 。

[0062] 2.2定义动作集合 A_t

[0063] 代理选择节点 $n_t^s \in \{N_t^{\psi_s} \cap N_t^{\psi_s}(n_t^v)\}$ 的动作 A_t 按如下方式定义:

$$[0064] \quad A_t = \{\varepsilon\} \cup \left\{ (n_t^v - n_t^s) : n_t^s \in \{N_t^{\psi_s} \cap N_t^{\psi_s}(n_t^v)\} \right\}$$

[0065] 其中 ε 表示可以到达终端状态的任意动作。当代理选择当前虚拟节点 n_t^v 的物理节点 n_t^s 时,它转换到下一个状态 S_{t+1} 。

[0066] 2.3定义状态转移概率矩阵 P_r

[0067] 当处于状态 S_t 时,代理选择节点 $n_t^s \in \{N_t^{\psi_s} \cap N_t^{\psi_s}(n_t^v)\}$ 的动作 A_t 转换到下一个状态 S_{t+1} 的状态转移可能性被定义为:

$$[0068] \quad P_r(S_{t+1}|n_t^v - n_t^s, S_t) = 1$$

[0069] 2.4定义回报函数R

$$[0070] \quad R_j = \begin{cases} R_j, & is_{end_j} \text{ is true} \\ R_j + \gamma Q'(\phi(S_j), \underset{a'}{\operatorname{argmax}} Q(\phi(S_j), a; \theta); \theta'), & is_{end_j} \text{ is false} \end{cases}$$

[0071] 其中,j表示第j个样本, γ 为衰减因子,Q为当前Q网络, Q' 为目标Q网络, a' 为在当前Q网络中最大Q值对应的动作。

[0072] 步骤3,利用DDQN算法进行虚拟节点的映射;参见图1,具体如下:

[0073] 3.1随机初始化所有状态和动作对应的价值Q,随机初始化当前Q网络(MainNet)的所有参数 θ ,初始化目标Q网络(TargetNet)的参数 $\theta' = \theta$,清空经验回放集合D。

[0074] 3.2初始化状态集合S为当前状态的第一个状态,得到状态集合S的特征向量,并用 $\phi(S)$ 表示。

[0075] 3.3在MainNet中使用状态集合S的特征向量 $\phi(S)$ 作为输入,得到MainNet的所有动作对应的Q值输出,使用 ε -贪婪法在当前Q值输出中选择对应的动作A。

[0076] 3.4在状态S执行当前动作A,得到新状态 S' ,特征向量 $\phi(S')$ 和奖励R,以及终止状态标志 is_end ;

[0077] 3.5将 $\{\phi(S), A, R, \phi(S'), is_end\}$ 这五元组存入经验回放集合D。

[0078] 3.6令初始状态 $S = S'$;

[0079] 3.7从经验回放集合D中采集m个样本 $\{\phi(S_j), A_j, R_j, \phi(S'_j), is_{end_j}\}, j = 1, 2 \dots m$,计算当前价值Q值 y_j :

$$[0080] \quad y_j = \begin{cases} R_j, & is_{end_j} \text{ is true} \\ R_j + \gamma Q'(\phi(S'_j), \underset{a'}{\operatorname{argmax}} Q(\phi(S'_j), a; \theta); \theta'), & is_{end_j} \text{ is false} \end{cases}$$

[0081] 3.8使用均方差损失函数 $\frac{1}{m} \sum_{j=1}^m (y_j - Q(\phi(S_j), A_j, \theta))^2$,通过神经网络的梯度反向传播来更新当前Q网络的所有参数 θ 。

[0082] 3.9重复步骤3.1-步骤3.8多次,如果重复次数达到目标网络的更新频率C,则更新目标网络参数 $\theta' = \theta$ 。如果重复次数没有达到目标网络的更新频率C,则继续重复步骤3.1-步骤3.8。

[0083] 3.10判断终止状态标志 is_end 是否为终止状态;如果终止状态标志 is_end 是终止

状态,则进行步骤4,否则转到步骤3.2。

[0084] 步骤4,利用最短路径算法进行虚拟网络链路映射:

[0085] 4.1对于虚拟链路 l_{uw} ,首先排除 $\text{ReqBWL}(l_{uw}) > \text{BWL}(l_{mn})$ 的物理链路;其中, $\text{ReqBWL}(l_{uw})$ 为虚拟链路 l_{uw} 请求的带宽, $\text{BWL}(l_{mn})$ 为物理链路 l_{mn} 的剩余可用带宽;

[0086] 4.2初始化计数变量 $\text{count}=0$;

[0087] 4.3对于连接两个终端节点 u^v, w^v 的虚拟链路 l_{uw} ,首先找到虚拟链路 l_{uw} 终端映射的物理节点 m^s, n^s ,然后使用Floyd算法寻找 m^s, n^s 间的最短路径 p^s 。

[0088] 4.4若 m^s, n^s 间的最短路径 p^s 存在,则 $\text{count}++$,继续执行步骤4.1-步骤4.3,直到 count 等于虚拟链路总数时执行步骤5,否则输出虚拟网络映射失败。

[0089] 步骤5,更新物理网络资源,包括CPU资源和链路带宽资源。

[0090] 本发明通过马尔可夫决策过程(MDP模型)获取当前的VNE环境,例如物理网络的资源情况、链路连接状态以及虚拟网络的请求量。在每次映射开始时,第一个映射的节点被随机传送到某个物理节点,然后基于DDQN算法获得奖励值 R ,更新网络参数,然后移动到下一个状态 S_{t+1} 。

[0091] 本发明中使用的DDQN算法,它是一种将Double Q-learning引入到DQN,即将action的选择和评估解耦开的算法。DDQN中存在两个结构完全相同但是参数却不同的网络,用当前Q网络中Q值选择 $\arg\max_{a'} Q(\phi(S'_j), a; \theta)$ 对应的action,在目标Q网络中评估这个action中的Q值。目标Q网络中网络参数不需要迭代更新,而是每隔一段时间从当前Q网络中复制过来,实现延时更新,这样可以减少两个网络的相关性。

[0092] 因为DDQN算法的核心思想是将目标动作的选择和目标动作的评估分别使用不同的值函数来实现,所以可以有效地解决过优估计问题。通过这种自适应学习方案,可以通过节约能耗和提高VNR接收率来获得全局最优的映射方法,比传统方法相比,有更好的灵活性。

[0093] 该方法首先将虚拟节点映射问题建模为马尔可夫决策过程,使用DDQN算法进行虚拟节点映射,最后使用最短路径算法完成虚拟链路的映射。实验表明,该方法能够降低能耗,提高请求接受率并提高长期平均收益。

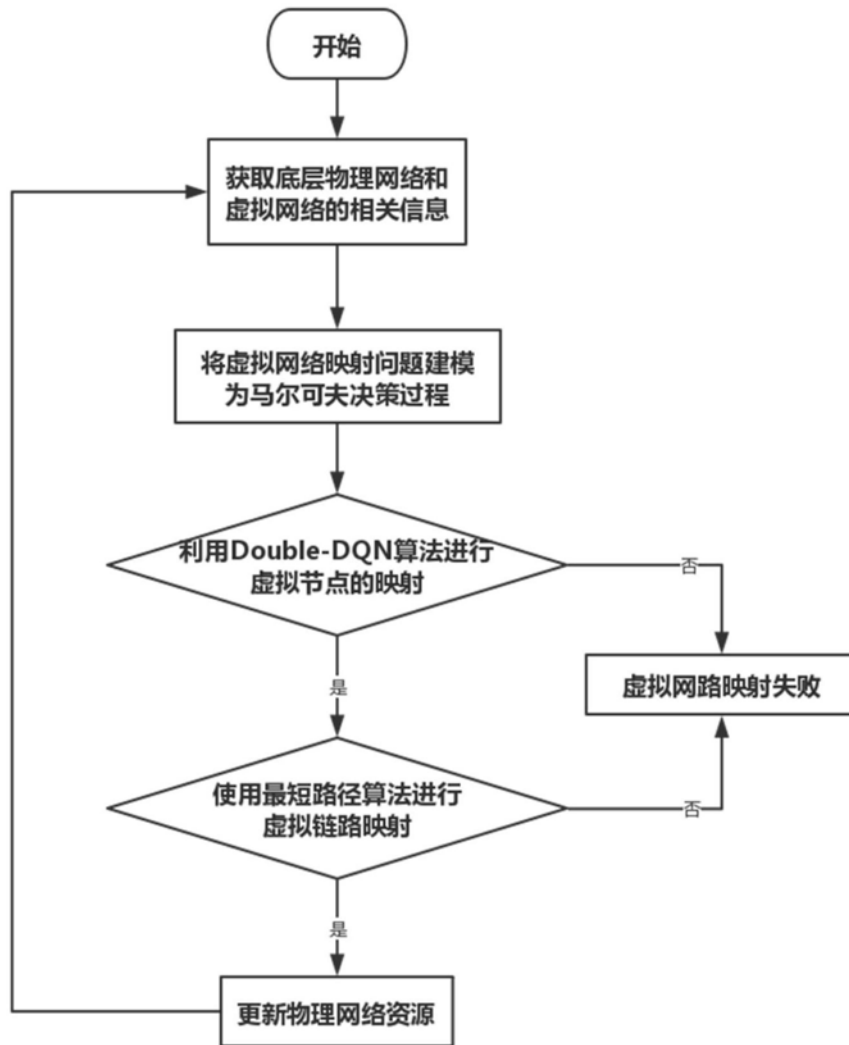


图1