# Abstract

摘要：软件定义网络 (SDN) 中的分布式控制器因其在当前 SDN 环境中的可扩展和可靠部署而成为一种有前途的方法。由于网络流量随时间和空间变化，交换机和控制器之间的静态映射会导致控制器之间的负载分布不均匀。交换机的动态迁移方法可以在 SDN 控制器之间提供平衡的负载分配。最近，现有的用于动态开关迁移的强化学习 (RL) 方法（例如 MARVEL）正在将每个控制器的负载平衡建模为线性优化。即使它被广泛用于网络流建模，这种线性优化也不能很好地适应 SDN 控制器的实际工作负载，因为资源类型之间的相关性会出乎意料地不断变化。因此，使用资源利用的线性模型很难区分当前过载的资源类型。此外，这会产生很高的时间成本。在本文中，我们提出了一种基于强化学习的开关和控制器选择方案，用于开关迁移、开关感知强化学习负载平衡（SAR-LB）。 SAR-LB 使用控制器和交换机中各种资源类型的利用率作为神经网络的输入。它还将开关视为 RL 代理以减少学习的动作空间，同时考虑所有迁移情况。我们的实验结果表明，由于交换机迁移的准确决策，SAR-LB 在 SDN 控制器之间实现了更好（接近均匀）的负载分配。与现有方案相比，所提出的方案在分布式 SDN 控制器之间实现了更好的归一化标准偏差高达 34%。

# INTRODUCTION

SDN 提供了强大的可编程网络架构，可用于设计定义网络实体（例如，路由器、交换机、负载均衡器、防火墙和 VPN 等硬件设备）位置的逻辑拓扑结构 [1]。 SDN 将网络空间分为控制平面和数据平面。在数据平面中，交换机在主机之间传输实际数据，而控制平面定义了流规则，即在交换机之间路由数据的规则[2]。数据平面和控制平面的解耦使得网络逻辑的高效管理成为可能。 然而，当传入的数据包大量时，单个 SDN 控制器 [3] 可能成为性能瓶颈，所有 SDN 交换机都连接到单个 SDN 控制器.此外，控制器可能是单点故障 [4]，因为必须停止所有数据包处理。为了解决这两个主要问题，提出了分布式 SDN 控制器 [5]。在分布式 SDN 控制器中，每个 SDN 控制器管理自己的域网络，而 SDN 交换机静态映射到特定控制器。因此，控制器之间可能会出现负载不平衡，并且由于交换机和控制器之间的静态映射，控制器之间的负载分布不均匀变得很明显。

为了解决这个负载不平衡问题，已经研究了动态切换到控制器的映射变化，以实现分布式 SDN 控制器之间的负载均衡[6-12]。这些方法主要关注具有交换机迁移方案的分布式 SDN 控制器的负载平衡，它们基于重复优化方法，贪婪地选择用于迁移的交换机到控制器映射。此外，一些方法使基于强化学习 (RL) 的决策适应切换到控制器的映射变化 [13-15]。通常，基于 RL 的交换机迁移决策涉及在各种 SDN 环境中自适应地训练模型。然而，传统的基于 RL 的方法使用表格 Q 学习，这是一种众所周知的无模型 RL 方法，用于探索环境以避免局部最优问题。由于Q表[14,15]的大小有限，其状态定义不能将每个控制器的负载表示为连续值，因此很难利用控制器之间负载的动态变化。这种限制使得难以及时针对各个控制器负载的变化进行切换迁移决策。为了解决这个问题，最先进的基于 RL 的方法 MARVEL [13] 采用神经网络来近似连续状态空间中的动作值，并将控制器之间的负载平衡建模为线性优化，广泛用于建模网络流。为了模拟线性优化，他们将每个控制器的利用率定义为每个资源类型的利用率的线性总和，具有固定的权重。然而，最近的一项研究 [16] 表明，每个控制器的利用率无法通过线性优化有效建模，因为当 PACKET\_IN 的速率发生变化时，每种资源类型的利用率的相关性也会发生变化。此外，当控制器使用高网络带宽时，即使网络带宽的利用率一直很高 [17]，CPU 利用率也可能会意外抖动。因此，资源利用的线性模型很难区分当前过载的资源类型。此外，MARVEL 的简化动作空间使问题恶化。这是因为它的动作空间只考虑了有限数量的交换机迁移案例。因此，传统的 RL 方法对交换机迁移做出了优化程度较低的决策，从而导致控制器之间的负载不均衡。此外，将 switch 迁移到另一个域的时间成本相对较高 [9,18]。在这项研究中，我们提出了一种切换感知强化学习负载平衡（SAR-LB）方案来解决负载平衡的线性优化问题。我们将各种资源类型的利用率视为输入状态。此外，为了以小动作有效地考虑所有可能的 switch 迁移情况，我们将 switch 定义为 SAR-LB 中的 RL 代理。当我们将交换机定义为 RL 代理时，我们的 RL 模型的动作大小仅限于可能的目标控制器的数量，同时使我们能够在工作阶段考虑所有可能的交换机迁移情况。因此，SAR-LB 更准确地选择目标控制器。我们在模拟的 SDN 环境中评估了我们提出的方案。结果表明，我们提出的方案在分布式控制器之间实现了比其他方案更好的归一化标准偏差高达 34%。本研究的贡献如下：

• 我们通过将控制器的各种类型的资源利用和要迁移的交换机作为RL 模型的输入状态，来增强交换机迁移决策方案，以应对资源类型之间相关性的变化；

• 我们通过在提议的方案中将开关建模为 RL 代理，有效地考虑了所有可能的迁移案例，其中动作大小（即控制器的数量）相对较小；

• 我们在随机和各种环境设置下对我们提出的方案和其他竞争方案进行性能评估。我们在两种网络条件（即，有/没有突发网络流量）和具有不同数量的交换机和控制器的五种情况下评估 SAR-LB。结果表明，当交换机数量远大于控制器数量时，我们的方案表现更好，这证实了我们的 SAR-LB 非常适合控制器管理数十个以上交换机的公共数据中心 [19-22 ]。本文的其余部分组织如下。我们在第 2 节中描述了我们的研究背景。在第 3 节中，我们介绍了相关工作。我们描述了我们提出的方案的细节，并分别在第 4 节和第 5 节中介绍了实验设置和评估结果。讨论在第 6 节进行，我们在第 7 节结束我们的论文。

# 2. Background

在本节中，我们描述了负载的交换机迁移的背景

分布式SDN控制器中的平衡。 然后，我们介绍了在分布式 SDN 控制器中负载均衡的交换机迁移域中使用的 RL。

## 2.1. Switch Migration in Load Balancing ofDistributed SDN Controllers

在分布式SDN控制器中，一个交换机可以被多个SDN控制器控制。但是，只有一个控制器可以管理交换机来处理PACKET\_IN请求和流规则的安装。交换机和控制器的关系按照南向接口协议规定。例如，OpenFlow [2] 提供 SDN 控制器到具有不同类型角色的交换机的多个连接：（i）“主”，（ii）“相等”，以及（iii）“从”。一个 SDN 交换机可以由多个具有“平等”和“从”角色的控制器控制。不同之处在于具有“从属”角色的 SDN 控制器对 SDN 交换机具有只读访问权限。但是，具有“平等”角色的控制器接收所有异步消息，并具有管理这些从 SDN 交换机发送的消息的完全访问权限。 “主人”角色类似于“平等”角色；然而，区别在于SDN交换机只有一个“主”来管理异步消息，其他控制器成为交换机的“从”。具有不同角色类型的控制器可以为 SDN 网络提供容错能力。此外，通过将交换机从过度使用的控制器迁移到未充分使用的控制器，并在控制器之间更改角色，可以实现分布式 SDN 控制器之间的负载分配。因此，来自交换机的 PACKET\_IN 请求可以由多个控制器控制。

## 2.2. Reinforcement Learning

强化学习是一种学习，关注在动态环境中做什么[23]。与其他学习类别（例如，监督学习、无监督学习）不同，RL 的学习是通过与环境交互来处理的。这种交互由 RL 框架的基本元素组成：代理（即，它从环境中获得奖励并将动作发送回环境），状态（即，它代表环境的当前情况），以及奖励（即来自环境的反馈以改变状态）。 RL 可以处理复杂的问题。例如，RL 被用作在医疗保健、机器人和网络自动化等复杂系统中训练 AI 模型的工具 [24]。图 1 展示了深度 Q 网络 (DQN) 的概览。 DQN [25] 是最著名的深度强化学习方法，它有两个特点。第一个特征是重放缓冲区。通过使用重放缓冲区，DQN 可以存储和利用当前训练步骤中的先前经验。第二个是目标Q网络。目标 Q 网络是 Q 网络的支持网络，由 Q 网络缓慢训练，而 Q 网络则直接通过重放缓冲区中的样本进行训练。通过设置目标Q网络，由于目标Q网络的保守变化，可以更稳定地训练Q网络。

算法 1 描述了 DQN 的整个过程。 在第 4 行，DQN 代理通过 -decay 策略选择一个动作。 在-衰减策略中，动作是随机选择的或由Q网络选择的，随机动作或神经网络选择动作的概率由的值决定。 选择动作后，由状态、动作、下一个状态、奖励和完成（即定义情节结束的布尔值）组成的体验被保存到重放缓冲区。 DQN 的训练过程在第 11-19 行定义。 特别是，DQN 的损失函数是训练的重要组成部分。 为了定义损失函数，yx 由奖励和下一状态的最大目标 Q 值的组合定义。 这个 yx 被用作监督学习的标记值。 这样，Q network 和 yx 值的均方误差就成为 DQN 的损失函数。

## 3.1. Switch Migration Schemes for Load Balancing in SDN Environment

以往研究中的交换机迁移方案集中在PACKET\_IN处理、资源利用和交换机组迁移以提高迁移效率。在[6-8]的研究中，交换机迁移方案的决定是根据对分布式SDN控制器的PACKET\_IN请求的数量来考虑的。在[6]中，作者设计了一种采用分布式跳频算法（DHA）的迁移算法。他们提出的算法从过度使用的控制器中随机选择一个开关，并在每次迭代中将即将到来的迁移活动广播到邻居控制器。因此，随着迁移后全局网络视图的状态同步，通信开销增加。 Al-Tam 等人。 [7] 提出了一种具有“移位”和“交换”移动的启发式算法，用于切换迁移。然而，他们提出的方法导致迁移的决策时间增加。这是因为他们提出的“转移”和“交换”移动会反复发生，直到确定最佳开关到控制器对以在 SDN 控制器之间均匀分配负载。杰等人。 [8]还提出了一种基于PACKET\_IN请求响应时间的交换机迁移方案。响应时间取决于南向通信的网络状况。例如，交换机和控制器之间的链路故障可能会影响响应时间的延迟。在 [9,10] 中，作者还提出了一种交换机迁移方案，重点是最大化分布式 SDN 控制器的资源使用。在他们的方法中，过度使用的控制器选择负载较小且距离最远的开关来消除控制器的过度使用状态。在程等人。 [9]，每个未充分利用的控制器独立玩游戏，直到只有一个控制器（即目标控制器），表明控制器迁移。交换机迁移的决策重复发生，直到为迁移选择了最佳目标控制器。萨胡等人。 [10] 还提出了一个称为 ESMLB 的框架。他们提出的框架在决策之前对每对切换到控制器使用初步迭代计算。因此，初步计算的计算开销可能会影响负载均衡的响应时间以及分布式 SDN 控制器环境中的可扩展性。在 [11] 和 [12] 中研究了带有交换机组迁移的交换机迁移决策。其中，交换机组从过度使用的控制器迁移到未充分使用的控制器是迭代执行的。交换机组迁移增加了迁移成本。在大提琴等人。 [11]，作者还提出了一种基于图划分技术的决策算法。在他们的方法中，交换机迁移决策是在交换机和控制器之间短距离内的分区之间进行的。因此，他们的方法适用于小型网络。吴等人。 [12]通过利用具有交换机迭代迁移的非合作博弈设计了基于交换机迁移的负载平衡方案。在他们的方法中，交换机迁移活动在每一轮游戏中都贪婪地发生。而且，交换机组迁移增加了交换机迁移的额外开销。

## 3.2. Reinforcement Learning-Based Load Balancing in SDN Environment

在最近的研究中，人工智能 (AI) 的集成已经解决了分布式 SDN 控制器中的负载平衡问题。 RL 是 SDN 环境中广泛使用的 AI 方法。特别是，它通过由 RL 代理训练的最优策略用于负载平衡。在最近的研究中，SDN 环境中基于 RL 的研究可以分为两类：(i) 在数据平面的负载平衡中使用 RL，以及 (ii) 在交换机迁移中使用 RL 以实现控制平面的负载平衡.在 RL 用于数据平面的负载均衡的情况下，Mu 等人。 [26] 检测

大象提前流动以减少控制器和交换机之间的流量。他们采用了两种 RL 算法（即传统强化学习和深度强化学习算法）来更好地控制开销，表命中率分别提高了 60% 和 14%。在[27]中，吴等人。提出了一种深度Q网络（DQN）授权的动态流数据驱动的控制器放置问题（D4CPP）方法，该方法充分考虑了流量波动，减少延迟和负载平衡。他们的模拟结果表明，在动态流量波动的 SDN 环境中，D4CPP 在延迟方面优于传统方案 13%，在负载平衡方面优于传统方案 50%。托苏尼迪斯等人。 [28] 专注于使用 CNN 和 Q-learning 的预测方法进行动态负载平衡。这些方法试图预测流量以及控制器和网络的状态，以最大限度地提高网络的可用性。他们提出的方法仅将其方法的性能与传统方法（例如循环法、加权循环法）进行比较，而没有与其他基于 RL 的解决方案进行比较。在 RL 用于控制平面负载平衡的情况下，分布式 SDN 控制器的基于交换机迁移的负载平衡已通过称为 MARVEL [13] 的多代理 RL 解决。 MARVEL 用泊松过程产生的控制器负载训练代理，揭示了分布式 SDN 控制器在现实世界 SDN 控制器负载中造成的控制器过度使用。 MARVEL 通过将动作空间定义为三种情况（即 STAY、IMPORT 和 EXPORT）来简化 switch 迁移动作。但是，简化的动作空间在当前步骤中没有考虑许多其他可能的开关迁移情况。李等人。 [14] 还提出了基于 RL 的交换机迁移决策，用于 SDN 控制器负载平衡。他们提出的方法侧重于选择未充分利用的控制器，在交换机和控制器之间具有最少的跳数。这限制了未充分利用的控制器的选择空间，无法提供分布式 SDN 控制器架构的可扩展性。敏等人。 [15] 使用 Q 学习算法解决了分布式 SDN 控制器的交换机迁移问题。在他们的论文中，作者描述了对所提出算法的分析。但是，在对其算法的评估上，并没有与其他交换机迁移算法进行比较

# 4. Design of SAR-LB

我们提出的交换机感知强化学习负载平衡 (SAR-LB) 是一种基于强化学习的交换机和控制器选择方案，用于交换机迁移。它将各种资源类型的利用率作为 RL 代理的输入状态，并将交换机定义为 RL 代理，以限制动作大小的数量，同时考虑所有可能的交换机迁移情况。我们将 SAR-LB 的设计分为三部分：（i）RL 通信通道的定义，（ii）训练和工作阶段，以及（iii）DQN 代理的 DNN 模型。为了详细描述 SAR-LB，将我们提出的方案 SAR-LB 与开关迁移中的现有 RL 方法进行比较将是有效的。表 1 显示了传统的基于 RL 的开关迁移方法与我们的 SAR-LB 之间的比较。 MARVEL 和 SAR-LB 都使用 DNN 模型来近似当前 SDN 环境状态下的动作值，而 [14,15] 中的其他方法在 Q 值的计算中使用表格 Q 学习。如果这些方法使用表格 Q 学习，由于 Q 表的空间有限，只能将有限数量的状态定义为状态。在 Q 表中，行索引表示选择作为初始控制器的控制器，列索引表示选择的交换机迁移案例。当动作被 Q 表选择时，它被奖励被动更新。因此，难以适应负载的动态变化。为了及时适应控制器负载的动态变化，每个控制器的负载应该用作神经网络的输入状态。或者应该使用机器学习方法来近似控制器的每个负载信息中的动作值。 MARVEL 和 SAR-LB 在他们的方法中考虑了这些因素。但是，MARVEL 和 SAR-LB 之间存在差异。与 SAR-LB 不同，MARVEL 基于多智能体强化学习 (MARL)，它是 RL 的分布式版本。但是，MARVEL 并没有考虑每种资源类型的资源利用率；它将每个控制器的资源利用率建模为线性优化。由于每种资源类型的利用率并非始终相关，因此这会导致优化程度较低的交换机迁移。例如，控制器的内存和 CPU 利用率随着 PACKET\_IN 消息数量的增加而呈指数增加 [16]。

因此，具有恒定权重的资源类型的线性模型几乎不可能对各种资源类型之间动态变化的相关性进行建模。 此外，当控制器的网络带宽使用率高时（即网络带宽利用率高），控制器的 CPU 利用率可能会意外抖动，而网络带宽的利用率仍然保持不变 [17]。 因此，如果未定义每种资源类型的利用率，则无法阐明过度使用的原因。 因此，最好将控制器的负载建模为每种资源类型的利用率。 此外，应考虑所有可能的交换机迁移案例，以便为交换机迁移做出准确的决策

## 4.1. The Definition ofRL Communication Channel for Switch Migration

SAR-LB 使用各种资源类型的资源利用率作为训练阶段的输入，并且可以通过考虑更多开关迁移案例来提供更精确的动作。我们为我们的研究构建了 RL 框架，如图 2 所示。为了组成 RL 的框架，需要定义两个实体和三个通信渠道。这两个实体被定义为代理和环境。代理被定义为用于迁移的候选 Open vSwitch [29]，环境被定义为具有目标控制器以迁移候选交换机的 SDN 环境。三个沟通渠道（即状态、奖励和行动）是代理学习过程中的重要因素。因为状态和动作空间定义了 RL 代理的输入和输出空间。此外，奖励是一个重要的术语，因为在计算每个智能体的目标函数时会使用奖励。为了评估每个输入状态下动作的价值，神经网络的输出被定义为当前控制器的价值SDN 环境状态。因此，开关的目标控制器由具有最大值的输出元素的索引选择。图 3 定义了 RL 模型的状态。该状态是控制器的资源利用率和选择迁移的交换机的资源利用率的线性特征向量。在图 3 中，我们以表格格式表示线性特征向量，以便更好地理解状态定义。每行定义每个控制器对每种资源类型的资源利用率。控制器的这种资源利用率定义为五个元组； Ci.ID是用于对每个控制器进行分类的标识符号，零（即0）是指示当前信息是关于控制器的布尔标志。此外，下面三个 Ci.Util 定义了控制器对每种资源类型（例如 CPU、RAM 和网络带宽）的资源利用率。控制器的这些信息与其余控制器的信息合并，这些信息的定义与我们在本段中描述的格式相同。 在描述了控制器信息之后，将有关选定要迁移的交换机的附加信息添加到特征向量中，如图3表的最后一行所示，交换机上的信息与控制器信息的描述类似。但是交换机信息中的控制器ID代表的是交换机所在的初始控制器的ID。在此上下文中，布尔标志为1（即1），表示该信息是关于交换机的，后面三个Sselected.Util 表示被选中交换机迁移时在初始控制器中的资源利用率。交换机迁移可以定义为选择目标控制器的过程，其中将添加所选交换机的负载。在我们对交换机迁移的定义中，初始控制器还需要与学习环境中的其他控制器一起被视为候选控制器。在这种情况下，如图 3 的蓝色箭头所示，应从初始控制器 Ci 的负载中移除所选开关的负载，以便与其他控制器进行公平比较，因为目标控制器的选择是基于比较环境中控制器之间的负载。为了在奖励中反映初始控制器和目标控制器的资源利用率之间的差异，奖励计算为上一步和当前时间步之间每种资源类型的资源利用率的平方差的变化。它的计算方法如等式 (1) 中所述。

## 4.2. Training and Working Phases ofSAR-LB

对于分布式 SDN 控制器的负载平衡，我们分别在算法 2 和算法 3 中提出了具有训练和工作阶段的交换机迁移决策算法。算法 2 描述了我们提出的使用 DQN 代理的交换机迁移方案 [25]。重复次数在算法2的第1行中设置。初始化SDN环境后，根据控制器和目标交换机的信息创建初始状态值以进行迁移（算法3中的第2行至第4行）。 DQN 代理与 SDN 环境之间的通信和动作执行过程从第 6 行到第 12 行进行。第 13 行到第 14 行定义了 DQN 代理的学习过程。在算法 2 中，我们提出了通过强化学习进行交换机迁移的训练阶段.然而，训练阶段由于其随机切换选择而难以直接利用。为了有效利用训练好的模型，算法3中描述了交换机迁移的工作阶段。在算法3中，通过第2行到第5行选择资源利用率最高的控制器。在第7行到第10行，具有控制器中每个开关的资源利用率Cmax.Sj，Cmax，作为神经网络的输入。此后，收集控制器 Cmax 中每个开关的动作和奖励。在第 11 行到第 12 行中，通过奖励最高的动作执行 switch 迁移。在算法 2 中，我们提出了通过强化学习进行切换迁移的训练阶段。 然而，训练阶段由于其随机切换选择而难以直接利用。 为了有效利用训练好的模型，算法3中描述了交换机迁移的工作阶段。在算法3中，通过第2行到第5行选择资源利用率最高的控制器。在第7行到第10行，具有 控制器中每个开关的资源利用率Cmax.Sj，Cmax，作为神经网络的输入。 此后，收集控制器 Cmax 中每个开关的动作和奖励。 在第 11 行到第 12 行中，通过奖励最高的动作执行 switch 迁移。

## 4.3. DNN Model for DQN Agent

DQN 代理通过两种类型的层进行配置，即 LSTM [30] 层和全连接（FC）层，以识别传入负载的变化。长短期记忆（LSTM）是一种循环神经网络（RNN）[30]。 RNN 是一种神经网络，旨在操作层之间的顺序信息。此外，其次，FC 层是简单的前馈神经网络。 FC 层通常构成网络中的最后几层。因此，全连接层的输入是最终池化层或卷积层或 LSTM 层的输出。在 LSTM 和 FC 的组合中，FC 层中输入特征的数量就是 LSTM 中隐藏单元的数量。如图 4 所示，历史长度定义了 LSTM 序列长度，它定义了在 DQN 重放缓冲区中采样经验时的连续经验数量。此后，将 LSTM 层的结果值连接到全连接层，从而可以确定考虑连续体验的行为。

# 5. Evaluation and Results

## 5.1. Experimental Environment

为了评估我们提出的方案 SAR-LB，我们使用基于 Python 的 SDN 环境定义了一个模拟环境，我们使用了带有 Intel Xeon Skylake（Gold 6230）/2.10 GHz（20 核）处理器、384 GB RAM 的主机 和 V100 GPU，如表 2 所示。来自模拟 SDN 环境的每个交换机通过泊松分布 [31] 为每个剧集创建工作负载。 此外，每种资源类型（即 CPU、RAM、网络带宽）的负载也为每个环境创建而不同，这使我们提出的方案能够在动态负载环境中学习。 基于 PyTorch [32] 的 DQN-agent 通过配置超参数来执行学习，如表 3 所示。

## 5.2. Competing Schemes to Evaluate with SAR-LB

为了评估我们提出的方案，我们设置并测试了以下方案作为比较。为了统计评估我们的方案，我们在随机 SDN 网络环境中训练了 SAR-LB 和 MARVEL。这种随机环境意味着每一集的网络环境随机产生不同的网络开销。此外，我们对四种不同的网络模拟的评估结果进行了平均，这些模拟是通过设置不同的随机种子生成的。我们的 SAR-LB 与其他方法的比较如下： 循环：在循环 (RR) 方案中，控制器中负载最高的交换机从过度使用的控制器中选择，并在其中选择迁移的目标控制器一种循环方式。 ElastiCon [34]：ElastiCon 是众所周知的动态开关迁移研究。为了平衡控制器之间的负载，他们的方案比较了每种交换机迁移情况，并根据控制器之间利用率的标准偏差选择最佳迁移情况。为了与 MARVEL 和 SAR-LB 进行公平比较，我们修改了他们的算法以使用初始控制器和目标控制器之间的利用率差异找到最佳迁移案例，如公式（1）所示，在两个 MARVEL 中都用作奖励函数和 SAR-LB。 MARVEL [13]：在 MARVEL 中，控制器运行多个代理，RL 模型将控制器资源利用率作为输入并决定迁移的最佳切换到控制器映射。然而，我们没有将我们的方案与 [14,15] 进行比较，因为我们认为 MARVEL 是最先进的方案，它利用 RL 进行切换迁移，而不是以前的研究，并且 [14,15] 的研究难以处理开关负载的动态变化。 SAR-LB：在我们提出的方案中，我们为具有最高负载的控制器收集开关的动作（即由 DQN 选择的目标控制器），然后将具有最大奖励的开关迁移到目标控制器。与 MARVEL 相比，我们的方案利用新颖的输入状态定义来表示每种资源类型的利用率。此外，我们方案的动作空间可以考虑可以迁移此开关的每个可能的目标控制器。

## 5.3. Evaluation Results and Analysis

在我们的模拟中，我们在 5 个场景中的 300 个时间步长内进行了实验：三种仅增加开关数量的情况和两种同时增加开关和控制器数量的情况。 此外，我们在这五个场景中对两种网络状况进行了评估：（i）没有突发流量的网络状况和（ii）有突发流量的网络状况。 有/没有突发流量的网络条件的平均工作负载如图 5 所示。每个控制器中的工作负载随时间动态变化。 因此，图 5 显示了不均匀的负载分布。图 5b 显示了我们在评估中使用的突发流量的平均工作负载分布。 在这个图中，每个控制器的网络开销意外地急剧增加。

### 5.3.1. Comparison of Load Balancing Rate

我们将我们提出的交换机迁移决策方案 SAR-LB 应用于所有控制器

在具有两种网络条件的五种场景中，如上所述。应用我们方案前后的负载均衡率如图6和图7所示。图中x轴表示时间步长，y轴表示工作负载分布的标准差。如图 6 和图 7 所示，应用 SAR-LB 后控制器的负载均衡（图 6 和图 7 中的“红”线与没有负载均衡的负载分配模式相比，控制器之间的负载分配更加均匀。此外，我们还在测试平台中应用了替代方案（即 RR、ElatiCon 和 MARVEL）来评估我们提出的方案的性能，这些方案中的工作负载分布结果分别以不同颜色显示在图 6 和图 7 中。 ，当具有四个控制器和 80 个交换机的情况下，我们的方法的负载分布标准偏差优于其他竞争方案 表 4 和 5 显示了 SAR-LB 和竞争方案在五种场景，分别在两种网络条件下，RR的标准偏差也导致高于b中没有负载均衡的平均标准偏差其他场景。在有四个控制器和 20 个交换机的环境中，没有突发网络流量的测试平台，MARVEL、ElastiCon 和 SAR-LB 分别实现了比没有负载均衡方案更好 112%、121% 和 126% 的负载均衡。这种趋势在仅增加开关数量的情况下更为明显。然而，当交换机和控制器的数量都增加时，我们提出的方案 SAR-LB 的负载平衡性能越来越差。因此，SAR-LB 的归一化标准偏差比 ElastiCon 高 20%，并且在五个评估方案中的四个评估方案中优于其他方案。然而，与 ElastiCon 和 MARVEL 相比，它在具有 16 个控制器和 80 个开关的场景中显示了 26–24% 的归一化标准偏差。突发网络流量评估结果如表5所示。

结果与表 4 相似。但是，随着开关或控制器数量的变化，标准偏差的差异变得更大。因此，在四个控制器和 20 个交换机的情况下，SAR-LB 表现出比 ElastiCon 好 1% 的归一化标准偏差，并且这种差异比没有突发流量的网络条件要大得多。因此，SAR-LB 的归一化标准偏差比 ElastiCon 高 34%。因此，当控制器数量少于 8 个时，SAR-LB 的表现优于 MARVEL 和 ElastiCon，并且当开关数量增加时，这种趋势更加强烈。然而，当控制器数量增加时，SAR-LB 的负载均衡性能不如其他方案。这是因为当控制器数量增加时，我们方案的输入状态和动作空间的维度也增加了，这使得我们方案中的 RL 代理难以学习 SDN 环境的当前状态并选择动作。

### 5.3.2. Comparison of Decision Time per Time-Step

表 6 描述了每个时间步长决策时间的比较，它描述了五个场景的实验结果。与 MARVEL 相比，SAR-LB 显示出 53-104% 的决策时间。这一结果是通过优化输入管道得出的，该优化将控制器中每个开关的输入状态合并为一个输入状态。此外，SAR-LB 和 MARVEL 显示出比 ElastiCon 和 RR 更慢的决策时间。这是由于使用 DNN 模型的 MARVEL 和 SAR-LB 的推理开销。然而，与总交换机迁移时间相比，推理开销相对较小，如 [9,18] 所示。因此，我们假设推理开销是可以接受的，以选择更有效的交换机迁移案例。尽管表 6 中 SAR-LB 显示出与 MARVEL 相似且更便宜的时间成本，但我们的算法很难在控制器之间分配。这个问题是我们算法的主要缺点。因此，我们将在第 6 节中讨论我们方案的可扩展性问题。

# 6. Discussion

我们提出的方案 SAR-LB 表明，与其他竞争方案相比，它在控制器之间的归一化标准偏差提高了 34%。改进主要是因为我们对沟通渠道、培训和工作阶段以及 DQN 模型的有效设计。此外，SAR-LB 的性能随着开关数量的增加而变得更好。因此，基于这些实验结果，我们声称 SAR-LB 可以用于数据中心网络中的常见 SDN 环境，因为数据中心中的每个控制器通常管理数十个交换机 [19-22]。然而，SAR-LB 在其可扩展性方面仍有改进的空间。我们建议将特征向量转换为固定大小的图像格式。由于输入大小与控制器和开关的数量无关，而且如果我们使用卷积层，我们可以在神经网络的深处堆叠层，因此可以利用这种转换作为优势。因此，此转换过程将提高可扩展性并产生更准确的决策。将我们的工作阶段算法并行化到多个控制器中以增强可扩展性是很难实现的。因此，我们建议分布式深度学习的数据并行性。它可以通过将工作阶段算法的推理开销分配到多个控制器上并利用集体通信库（例如 MPI [35]）收集推理结果来实现。另一种降低推理开销的方法是设置推理阈值，即只有对具有利用率阈值的控制器占用大量资源的交换机才能进行推理过程。

# 7. Conclusions

在本文中，我们提出了基于 RL 的交换机迁移决策 SAR-LB 的交换机迁移方案设计，该方案在分布式 SDN 控制器之间实现了比其他方案更均衡的负载分配。 与现有的 RL 方法不同，我们在状态信息中考虑了各种资源类型的资源利用率，并作为 RL 代理进行切换以有效地考虑所有切换迁移情况。 因此，我们提出的方案提供了更好的负载分配，因为它对交换机迁移做出了准确的决策。 我们在具有五种场景的模拟测试台中评估了 SAR-LB 的性能。 在每种情况下，我们都会改变控制器或交换机的数量。 结果表明，在五种场景中的四种情况下，我们提出的 SAR-LB 在负载分配方面的性能优于竞争方案，并且分布式 SDN 控制器之间的负载分配改进比现有方案高出 34%。