# Abstract

分布式软件定义网络 (SDN) 方法已被用来解决与使用单个 SDN 控制器相关的可扩展性问题，尤其是在大型网络中。然而，这种方法给分布式SDN控制器带来了负载均衡不均衡的新挑战。一些研究试图解决这个问题，但他们只关注发生过载时的负载平衡，并没有持续实现负载平衡。在本文中，我们提出了多阈值负载平衡（MTLB）交换机迁移方案，以实现控制器之间的连续负载平衡。这种方法将负载分为几个渐变级别，代表了当控制器的负载与其他控制器的负载不同时交换机迁移的基础，从而导致阈值动态调整。此外，我们的方案通过采用控制器负载状态作为传播负载信息的指标，减少了控制器之间不需要的更新操作的开销。我们使用 Floodlight 控制器和 Mininet 模拟器来实现我们的方案。实验结果表明，我们的方案在响应时间、控制开销和吞吐率方面比其他方案具有更好的综合性能。

# 1.INTRODUCTION

软件定义网络（Software-Defined Networking，SDN）是一种新兴的网络生成模型，它将控制平面与数据平面分离，实现网络可编程、灵活管理和网络集中控制[1]。该模型成为一种很有前途的网络范式，并已广泛应用于大规模网络[2]。随着人工智能应用的快速发展和网络规模的大规模增长，单个控制器可能成为性能瓶颈，无法满足来自更复杂网络的巨大流量请求[3]。此外，控制器之间的控制开销太高。因此，为了解决可扩展性问题 [4] 并避免单点故障 [5]，引入了分布式逻辑集中控制平面 [6]。在分布式架构中，网络被划分为多个域，每个SDN控制器管理一组交换机并与其他控制器同步信息以获得全局网络视图。分布式架构在逻辑上是集中的，但在物理上分布在多个控制器上，这不仅受益于分布式架构的可扩展性和可靠性，而且还保持了简单性[7]。然而，交换机和控制器之间的静态关系导致控制器之间的负载分布不均匀，特别是在高网络流量波动的情况下[8]。因此，一些研究人员提出了多种SDN控制器的负载均衡方案。 ElastiCon [9] 首次提出了交换机迁移的概念，以解决控制器之间负载分布不均的问题。因此，如果交换机的主控制器负载很高，则可以将交换机从一个控制器迁移到另一个控制器。基于添加到 Openflow 协议 [10] 的新规则；每个交换机可以由多个不同角色的控制器控制：主、等和从。通常，交换机只有一个主控制器，它可以指示数据平面、获取交换机的状态并将规则写入其交换机。但是，这些研究并未关注如何为迁移操作选择合适的交换机和目标控制器。交换机迁移是将交换机角色主机从一个控制器更改为另一个控制器的操作。因此，它主要侧重于将一个或多个交换机的角色从高负载控制器转变为低负载控制器，并且交换机迁移问题的复杂性被证明是 NP-hard [11]。目前的研究提出了一些解决方案，但它们都有其局限性。在 [12,13] 方案中，控制器需要在不知道发生任何变化的情况下定期从其他控制器收集负载信息。不可避免地，由于大量的状态同步，这个过程会导致高开销，特别是如果网络状态不经常改变[14]。张等人。 [15]如果利用率在0.9-1范围内，则认为控制器过载，否则正常。 [16,17] 中的方案使用单个静态阈值来检测负载不平衡，当负载超过此阈值时，将发生交换机迁移。然而，单一阈值无法准确区分控制器负载状态，这将导致研究容错而不是负载平衡。因此，将无法获得足够高的性能。为了解决这个问题，[18] 的作者设计了一种双阈值方法来平均分配负载，Zhang 等人。 [19] 引入了多个阈值的概念来区分控制器状态，并对控制器进行分类并进行交换机迁移。但是，他们的算法不适用于大型系统，并且不稳健和可靠。然而，考虑单个静态阈值作为指标

对于交换机迁移，这在某种程度上限制了过载的发生，但无法获得真正的负载平衡。单个阈值固定为一个值，无论负载情况如何都无法准确区分控制器负载状态。因此，控制器可能会经历长时间的高负载模式。此外，当控制器负载很高时（控制器负载即将达到控制器容量或接近其瓶颈），迁移操作成本高昂，这会有效地导致性能下降。此外，每个控制器的进程会定期将其负载信息与其他控制器同步，从而导致高通信开销，并影响性能，尤其是在大型网络中。

因此，由于上述缺点，我们提出了一种多级阈值负载平衡（MTLB）方案，以有效地解决控制器之间的负载不平衡问题，并预先排除控制器过载的发生。使用动态阈值，阈值取决于能够根据平均负载从一个值更改为另一个值的负载状态。 该方案直接准确区分负载状态，并且可以在负载水平增加或减少时动态调整阈值的值。 此外，该方案有效地调整了控制器之间的负载信息同步。 这项工作的主要贡献可以总结如下：

1. 我们提出了多级阈值交换机迁移负载平衡方案（MTLB），以实现分布式控制平面中控制器之间的持续负载平衡。该方案根据阈值将负载分为几个渐变级别。当一个控制器的负载与其他控制器明显不同时，执行交换机迁移。

2. 基于多级阈值概念设计负载信息同步策略。 MTLB 方案将负载分类为适当的状态阈值级别。网络中的每个控制器，当其负载超过或接近该阈值时，通知其他控制器用新状态更新负载信息。这大大减少了不需要的负载信息同步所导致的开销。

3. 为了确定合适的迁移选择，该方案通过仔细研究迁移交换机和目标控制器的行为，提供了一种为迁移操作选择合适的配对的方法。从而降低迁移频率并获得稳定性。

4. 对两种真实网络拓扑 Cernet 和 DFN 进行综合实验，以评估对比基线方案的性能。结果表明，MTLB方案比DDS、SMDM和EASM方案具有更好的综合性能。

## 1.1. Motivation

在这一部分，我们解释为什么需要多个阈值来

处理这些问题。负载均衡最重要的方面是准确判断控制器的负载状态，不同的评估标准会导致不同的负载状态。 MTLB 将负载分成几个渐变的间隔，其中每个控制器的负载水平相同，或者至少它们的负载水平非常接近，阈值的数量可以根据平均负载值增加或减少。因此，如我们在（第 4 节，A 小节系统模型）中说明的那样，每个级别代表一个单独的不同区间，并具有自己的阈值。控制器可能在每个时间间隔中过载，例如在第一个时间间隔中，如果其负载超过阈值一，等等，如果是这样，MTLB 方案可以有效地处理这种情况。场景应用 我们描述了我们提出的 MTLB 方案的功能。假设网络环境由 15 个节点组成，其中包括 3 个控制器。控制器C1、C2、C3与一组开关S={𝑠1、𝑠2、……、𝑠12}相连。每个控制器都有特定的负载能力。网络被划分为三个子域。每个交换机有一个作为主控制器运行的控制器和至少一个作为从控制器运行的控制器。在这种情况下，出现了两个同时考虑多控制器负载迁移的用例研究，如下所示：

在情况 1 中，让我们假设控制器 C2 的负载大约达到其容量的 65%，而控制器 c1 和 c3 分别为 35% 和 25%，如图 1（a）所示。根据 MTLB 方案，需要检测负载不平衡并进行交换机迁移，以平衡控制器之间的负载。 MLTB 方案将交换机 s7 从控制器 c2 迁移到控制器 c3，如图 1（b）所示。以平衡负载。

在情况 2 中，考虑控制器 C1 负载为其负载容量的 86%，而控制器 C2 和 C3 分别为 45% 和 50%，如图 2 所示。MTLB 方案发现负载分配不公平。由于控制器 c1 上的负载不平衡检测，然后将交换机 s5 从控制器 c1 迁移到 c2。并且负载正常返回。本文的其余部分安排如下。第 2 节描述了相关工作，而第 3 节详细描述了 MTLB 方案的设计。第 4 节描述了评估和实施设置并讨论了性能。最后，第 5 节介绍了我们的结论和未来的工作。

# 2. Related work

在本节中，我们将描述分布式控制器中交换机迁移负载平衡方法的背景。 我们将其分为三部分，包括周期性同步部分、单阈值和多阈值部分。 首先，展示了使用周期性同步方法到达全局网络视图的负载均衡方案。 然后，解释基于单一阈值的交换机迁移的传统文献及其局限性。 最后展示了多重阈值的一些研究进展。

## 2.1. Periodic synchronization

在分布式 SDN 控制器中，一些算法会在发生过载之前从其他控制器收集负载信息。余等人。 [13]采用了基于负载通知策略的负载平衡机制。每个控制器都需要在不知道发生任何变化的情况下，定期向其他控制器报告其负载信息。假设过载的控制器在做出迁移决策之前不再需要收集所有其他控制器的负载信息。由于大量的状态同步，这个过程不可避免地会导致高开销。同样，周等人。 [22] 引入了一种机制来选择一组交换机进行迁移，而不是一个一个地迁移交换机。它根据多余的工作负载选择迁移交换机组，这减少了平衡时间。但是，它增加了迁移交换机的数量，从而带来了额外的不必要的迁移成本。此外，选择迁移交换机和目标控制器的阶段也用于负载均衡；例如，Lan 等人。 [23]使用基于分层架构的负载信息收集策略来避免控制平面中的单点问题。此外，[24] 提出了一种负载平衡机制，可以平衡分布式 SDN 控制器之间的负载。它有一个集中控制器作为协调器运行，它定期收集控制器之间的负载统计信息，并决定是否执行交换机迁移。然而，这个方案有几个可能会降低性能的缺点。由于与其他控制器频繁交换消息，这些缺点包括高内存、带宽和 CPU 能力。然而，由于控制器之间的大量状态同步，这些操作导致网络不稳定，这些方案都没有减少周期性同步开销。

## 2.2. Single threshold

在 [21] 中，多个控制器根据目标控制器的响应时间同时迁移其交换机。第一轮，高负载的控制器选择轻负载的控制器进行迁移操作，第二轮，所有被选中的交换机都被迁移。由于同时进行交换机迁移，该方案消耗高网络带宽。高等人。 [25] 介绍集中式和分布式启发式

用于在动态情况下选择负载最重的交换机和负载最轻的控制器作为迁移过程对的控制器之间的工作负载迁移。但是，这种方法可能会导致目标控制器再次过载。钟等人。 [26] 提出了一种新的基于 SDN 网络方差分析的负载均衡概率方法，用于支持海量移动用户的动态需求。与此同时，库马尔等人。 [27]通过为迁移过程选择一组最轻的交换机负载来考虑该方法中的响应时间、抖动和丢包。 [28]提出了一种基于控制器性能的负载分配机制，将过载的控制器分为正常阶段和重型阶段。这仅允许管理任务的控制器收集每个单独控制器的负载并做出迁移决策。 [15] 中的工作设计了分布式决策策略 (DDS)，以提高交换机迁移和平衡控制器负载的效率。根据负载情况构建分布式迁移决策域。 DDS 用于平衡多域网络中的控制器负载。迁移交换机由交换机选择概率决定。同时考虑数据收集、交换机迁移和控制器状态同步的成本。在 [16] 中，交换机迁移和决策（SMDM）方案

建议的效率模型在迁移成本和负载平衡率之间进行权衡。在该方案中，每个控制器计算其聚合负载和负载差异矩阵，然后将其与计算的单个阈值进行比较。超过负载阈值的最大负载控制器总是直接设置为过载，而不判断控制器负载是否接近其瓶颈。因此，如果两个控制器之间的负载差异大于阈值，则触发交换机迁移。因此，该操作允许至少一个具有最大负载的控制器迁移其负载。它总是导致频繁的交换机迁移。将交换机从一个控制器迁移到另一个控制器可能会导致正在进行的流中断，从而严重影响性能。 [17] 中的作者提出了用于负载平衡的效率感知交换机迁移 (EASM)。该方案引入了负载差异矩阵和触发因子来衡量控制器上的负载平衡。 EASM 由三种方法组成。首先，计算触发因素来衡量控制器的负载平衡性能。然后，如果触发因素超过阈值执行迁移，则将计算值与阈值进行比较。最后，通过解决迁移效率问题选择迁移交换机，同时表征负载均衡率和迁移成本。但是，它增加了负载均衡时间和迁移成本，有效地降低了网络的性能。

## 2.3. Multiple threshold

为了解决频繁切换迁移的问题，方案[29]在没有找到合适的目标控制器进行迁移时动态调整源控制器的阈值。但是，需要更多的计算来调整阈值。为了减轻多集群方案中超级控制器的负载，HyperFlow [18] 引入了双阈值方法来处理多控制器负载平衡，通过采用集中和分布式架构来减少超级控制器的过载。 HybridFlow 应用类似集群的机制来构建网络，每个集群有多个控制器。此外，所提出的模型可以同时处理集群内部和集群之间的负载。但是，该方案仅在分布式分层方法中进行。Wang 等。 [30] 制定了使用控制器带宽的负载调整算法。该算法测量负载能力并将其划分为三个子区间。尽管如此，这些间隔不会影响与 70% 阈值相关的迁移决策。为了解决低效节点和过载节点之间的冲突，Zhang 等人。 [19]，提出了一种多阈值迁移策略负载均衡方案。该算法首先准确选择控制器负载描述参数，然后使用多个阈值区分控制器状态，选择交换机迁移到空闲控制器。该算法在试图使每个控制器工作在正常模式下的同时，精确地确定了控制器的状态。负载计算与 CPU 和 RAM 利用率相关，考虑了四个阈值。然而，该方案忽略了讨论他们的算法和实现细节。以及它不够健壮，不适合大规模网络。在本文中，为了解决上述问题，我们提出了多级阈值的概念，以在控制器之间分配负载并预先排除控制器过载的发生。一些负载均衡工作已在表 1 中列出。表 1 说明了各种软件定义的网络负载均衡方法的分析比较。该表还解释了每种技术的优缺点。

# 3. Multiple-level threshold load balancing scheme

在本节中，我们将介绍我们提出的方案。

如第 1 节所述，MTLB 方案引入了多级阈值方法，使用触发因子而不是定期更新来更新控制器之间的负载信息。 MTLB 方案最初将负载分类为适当的状态阈值级别，以处理信息同步和交换机迁移。当网络中的控制器负载超过或接近阈值水平时，它会通知其他控制器用其新状态更新它们的负载信息。此外，当控制器负载水平与其他控制器明显不同时，该方案会执行交换机迁移。最后，该方案通过仔细研究迁移交换机和目标控制器的行为来选择合适的对进行迁移操作。因此，它极大地减少了不需要的负载信息同步开销。对此，该方案分三个阶段进行。首先检查是否更新控制器的负载信息的方法。其次，检测控制器之间的负载不平衡并做出迁移决策的方法。最后，选择合适的交换机和控制器进行迁移操作的方法。此处使用的符号总结在表 2 中。

## 3.1. System model

在这项工作中，我们考虑由 𝑁 组成的 SDN 网络𝐺

控制器𝐶 = {𝑐1, 𝑐2,..., 𝑐𝑛} 和𝐾 开关𝑆 = {𝑠1, 𝑠2,..., 𝑠𝑘}。因此，我们描述了我们的架构设计，如图 3 所示。正如文献 [31] 中假设的那样，负载由传入 𝑃𝑎𝑐𝑘𝑒𝑡−𝐼𝑛 的累积数量组成。因此，我们认为流在时间段通过开关𝑠𝑖产生是𝑙𝑖作为唯一的组分控制器load.𝑓𝑙𝑜𝑤的=Σ𝐾𝑖=1𝑙𝑖其中𝑓𝑙𝑜𝑤表示总到达𝑃𝑎𝑐𝑘𝑒𝑡 - 𝐼𝑛到控制器𝑖𝐿𝑜𝑎𝑑𝑐𝑖=𝑓𝑙𝑜𝑤𝑐𝑎𝑝𝑎𝑐𝑖𝑡𝑦× 100 (2)

其中 𝑐𝑎𝑝𝑎𝑐𝑖𝑡𝑦 表示控制器 𝑖 在时间 𝑡 能够服务的总量 𝑃𝑎𝑐𝑘𝑒𝑡−𝐼𝑛。根据主/从约束，交换机可以由多个从控制器控制，但一次只能由一个主控制器控制。关系矩阵是 𝑋𝑖𝑗 ，其中 𝑋𝑖𝑗 = 0,1。在给定时间将每个开关连接到一个主控制器的条件如下描述： ∑𝑁 𝑗=1𝑋𝑖𝑗 = 1 𝑥𝑖𝑗 = {0, 1} ∀𝑖, 𝑗

### 3.1.1. The multiple threshold

该方案将负载分为几个渐变级别，阈值基于四分之一方法[32]。 第一季度平均控制器负载选择第一阈值，平均值选择第二阈值，第二季度平均控制器负载选择第三阈值。 该模型设置了四个状态级别和三个阈值。 阈值以最高阈值“过载”和最低阈值“阈值0”为界。 我们建立如图 4 所示的区间如下

(15, 35] , 𝜇1 = 25, 𝜀1 = 35

(35, 60] , 𝜇2 = 50, 𝜀2 = 60

(60, 85] , 𝜇3 = 75, 𝜀3 = 85

一旦负载状态达到阈值即为过载状态；表明当前控制器负载已完全超过其自身的工作能力。交换机迁移操作需要立即触发。高负载状态表示控制器处于异常工作状态，处于高负载状态，但仍可工作一段时间。如果其他控制器仍处于正常或空闲状态，则迁移交换机。正常状态表示控制器可以正常工作，并且仍然有能力处理一些突发情况。空闲状态控制器的事务远没有它的处理能力，控制器仍然有能力“引导”更多的开关。空闲控制器作为接收交换机的最高优先级。当一个控制器过载时，它会选择几个迁移开关𝑆𝑚进行迁移。如果其负载接近控制器负载和𝜇之间的差值，则该开关将被选为迁移开关𝑆𝑚。它与 𝜇 的差值计算如下：

𝜈𝑐𝑚 = 𝑙𝑜𝑎𝑑𝑐𝑚 − 𝜇 where 𝜈𝑐𝑚

(4) 是控制器负载和𝜇 之间的差异（

控制器负载与水平 i 平均负载之间的差异）。其余容量越大控制器大于或等于负载变化ν𝑐𝑚将被选择作为目标控制器：𝐿𝑜𝑎𝑑\*𝑐𝑡=𝐿𝑜𝑎𝑑𝑐𝑡+ν\*𝑐𝑚，ν\*𝑐𝑚=（ν𝑐𝑚×𝐶𝑚）𝐶𝑡

(5)

其中𝐿𝑜𝑎𝑑𝑐𝑡 代表目标控制器负载，𝜈∗ 𝑐𝑡

𝑐𝑚 来自目标控制器的负载变化，而𝐿𝑜𝑎𝑑∗ 是新的目标控制器

添加负载变化后的负载。 所提出模型的流程图如图 5 所示。

## 4.1. Experimental platform and physical devices

我们在 Java 中实现了我们提出的多阈值负载平衡方案，并使用 Floodlight SDN 控制器 [33] 作为实验平台。我们使用 Mininet Topology [34] 作为网络模拟测试平台进行模拟。 Floodlight 是一种基于 Java 的 OpenFlow 控制器，由包括 Big Switch 工程师在内的开发人员社区提供支持，可以轻松调整软件和开发应用程序。 Mininet 是斯坦福大学开发的网络模拟器，它创建了一个由虚拟主机、交换机、控制器和链接组成的网络。我们通过扩展 Floodlight 控制模块来支持分布式控制器协作机制来实现 MTLB 方案。实验平台如图6所示。物理设备配置（Intel Core i7-7567U CPU 3.5 GHz 16 GB RAM）。我们将六个节点视为具有 MTLB 方案和开放流交换机的控制器（Floodlight）。有七个具有相同实验配置的虚拟机，每个虚拟机运行 Ubuntu 16.04 LTS 操作系统和 JAVA 8。六个虚拟机运行 Floodlight 控制器，其他虚拟机运行 Mininet。我们使用这些控制器（𝐶1、𝐶2、…、𝐶6）来部署分布式 SDN 网络。每个控制器控制 Mininet 上的一部分网络拓扑。多个开关连接到控制器𝐶𝑖作为主控制器，同时控制器𝐶𝑗作为从控制器。控制器𝐶𝑖、𝐶𝑗 是相邻的控制器。

## 4.2. Topology selection

我们模拟了两个真实的网络拓扑 Cernet [35]（36 个节点

和 53 个链接）和来自 Internet Topology Zoo [37] 的 DFN [36]（58 个节点和 87 个链接）。 Cernet是中国最大的教育和研究网络。 DFN 是由科学家组织的德国国家科学和研究教育网络。 为了模拟实际的分布式网络流量，我们使用 Iperf [38] 来生成 TCP 流，控制器的平均流请求在 100 到 600 KB/s 之间。 我们采用不同低级别的控制器容量（网络中的控制器有各种低级别值），最大容量的控制器在DFN拓扑中为12MB，在Cernet拓扑中为6MB。 我们使用 Cbench [39] 工具进行性能评估。 表3，比较不同作品的属性，其中表示方案满足属性，×表示方案不满足属性，-表示方案不考虑属性。

## 4.3. Simulation results and analysis

为了验证 MTLB 方案的性能，我们对以下方案进行了比较：用于平衡软件定义网络 (EASM) 中控制器负载的效率感知交换机迁移 [17]，用于平衡负载的基于交换机迁移的决策方案 在SDN（SMDM）[16]和基于交换机向SDN控制平面（DDS）方案迁移的分布式决策策略[15]。 为了验证我们的方案，我们将性能指标与表 3 中的工作进行了比较。评估包括流设置时间、控制器负载、网络开销、响应时间、吞吐率、负载平衡率和迁移成本。 我们根据以下组讨论结果； 负载平衡、开销和性能评估。 我们通过改变负载和网络大小来多次运行模拟，以验证负载平衡时间的准确性。 在 Cernet 和 DFN 中捕获的超过 6 小时的流量负载数据如图 1 和图 2 所示。 分别参见 7(a) 和 7(b)。

### 4.3.1. Load balancing

我们根据负载平衡和流量完成来评估我们方案的负载平衡。控制器的负载：为了负载均衡，我们记录了两个网络拓扑中的负载变化，应用我们的方案之前和之后的负载均衡率。图 8 显示了在没有负载平衡机制的情况下控制器之间负载差异随时间变化的结果。我们观察到，与正常运行的控制器𝐶3、𝐶6 和𝐶6 相比，控制器𝐶1、𝐶2 和𝐶5 的负载通常随着时间的推移而增加，特别是在30 分钟后。因此，控制器 𝐶1 表现出最差的性能，特别是在 53 分钟时，负载约为 580 数据包/秒。为了验证 MTLB 方案，我们在将我们的方案应用于所有控制器 𝐶1、𝐶2、.... 𝐶6 六个小时后运行实验。我们观察到在控制器𝐶1、𝐶2和𝐶5中传入的𝑝𝑎𝑐𝑘𝑎𝑔𝑒\_𝑖𝑛消息的数量显着增加，而在其他控制器中𝐶3、𝐶6和𝐶6保持不变。图 9 显示了应用 MTLB 方案后控制器的负载平衡，该方案检测到控制器 𝐶1、𝐶2 和 𝐶5 中的负载不平衡，它们的负载超出阈值 𝜀2。因此，该方案干扰并执行了交换机迁移。

流量设置时间：我们记录流量设置时间，如图

无花果。 图 10(a) 和 10(b) 分别用于 Cernet 和 DFN，可以观察到 DDS 的最大流步时间随着流量负载的增加而增加。 然而，如图 10 所示，在 SMDM 和 EASM 方案的情况下，流建立时间会根据流请求而波动。 MTLB方案在整个流量负载变化期间具有最小的稳定流建立时间。 不同拓扑的模拟显示了相似的曲线模式。 在大规模拓扑中，平均流建立时间在 SMDM、EASM 和 DDS 场景下明显增加。 在这个模型中，当控制器𝐶𝑖的负载与其他控制器𝐶𝑗或𝐶𝑘不同时，它会触发负载平衡，将一个开关从高负载控制器动态迁移到低负载控制器，使得控制器之间的负载变化很小 .

### 4.3.2. Performance evaluation

我们在响应时间、吞吐量、丢包和延迟方面评估了我们的方案在负载平衡方面的性能。响应时间：正常控制器和过载控制器之间的控制器响应时间存在巨大差异。如果发生负载不平衡，响应时间将急剧减少。图 11 显示了 MTLB、SMDM、EASM 和 DDS 方案在控制平面的平均响应时间方面的比较。可以看出，MTLB 方案可以在变化的交通流量下保持稳定的响应时间。 MTLB方案的响应时间平均只有40ms，符合其最小化响应时间的目标。相比之下，DDS 的响应时间随着流请求计数的变化而波动最大。 SMDM 和 EASM 的波动范围较小。这个结果是合理的，因为 DDS 必须涉及多个 SDN 子域来进行迁移决策以及贪心算法的缺点。 SMDM 和 EASM 侧重于通过迁移效率条件降低迁移成本，而忽略了负载平衡。 MTLB 方案实现了多阈值负载均衡策略，确保持续迁移并快速降低控制器负载。吞吐量：控制器吞吐量是评估性能的重要因素。更大的控制器吞吐量对应于更好的负载平衡。图 12 显示了 MTLB、SMDM、EASM 和 DDS 方案在 Cernet 和 DFN 拓扑场景下的吞吐率。 DDS 的吞吐率最低，低于每秒 300 个数据包，因为它涉及多个用于迁移决策的 SDN 子域。第二个 SMDM 和 EASM 方案在控制器因迁移而过载时实现交换机迁移。因此，这些算法显示出类似的平均吞吐量结果，接近每秒 800 个数据包。 MTLB 方案在控制器负载与其他控制器的负载不同时实现交换机迁移，并且可以在没有过载的情况下做出交换机迁移决策。我们观察到，在 Cernet 和 DFN 中，MTLB 方案的平均吞吐量分别接近每秒 1300 和 5000 个数据包。这显示了所有其他方案中的最佳性能。包延迟：是等待包被服务的总时间的比率。它是用于绩效评估的重要关键指标。图 14 显示了 MTLB、SMDM、EASM 和 DDS 方案的数据包延迟比的结果。我们的 MTLB 方案的平均数据包延迟与其他方案的平均数据包延迟之间存在显着差异，并且观察到 SMDM 和 EASM 处于中间。 DDS 具有最大和剧烈的时间波动和最高峰值，而 MTLB 方案最低。

### 4.3.3. Overhead

流量开销：我们记录了 MLTB 方案的控制器之间的通信开销以及两种拓扑上其他方案的通信开销值，结果如图 1 和图 2 所示。 13(a) 和 13(b)。我们提出的 MTLB 方案记录了最低的通信开销。 SMDM 和 EASM 保持在中间，而 DDS 则随着流量负载的变化表现出很大的波动。这是因为MTLB方案采用控制器负载状态作为在控制器之间传播负载信息的指标。控制器在其负载状态发生变化时通知其他控制器。因此，MTLB 方案产生最低的通信开销。丢包：图15 显示了在MTLB、SMDM、EASM 和DDS 方案的迁移算法下迁移成本和丢包计数的结果。我们可以看到，SMDM 的迁移成本更高。由于迁移决策的相似性，EASM 和 DDS 的迁移成本有些相似。 MTLB 方案的迁移成本和丢包率最低。

### 4.4. Deployment environment

MTLB 方案适合在任何大型网络中实施。 抗高流量，适用于系统高峰期。 因此，可以部署在数据中心网络中以管理海量流量并提高性能。 并能在灾害管理网络、军事网络中工作。

# 5. Conclusion

在本文中，我们提出了一种多级阈值负载平衡方案，该方案有效地解决了分布式 SDN 控制器之间的负载不平衡问题，并预先排除了控制器过载的发生。 该方案直接准确区分负载状态，并在负载水平增加或减少时动态调整阈值。 此外，该方案通过有效地调整负载信息同步，减少了控制器之间频繁更新操作带来的开销。 模拟结果