

分布式 SDN 控制器放置问题研究

蔡 宁^{1,2}, 韩言妮^{1,2}, 安 伟¹, 徐 震¹

¹中国科学院 信息工程研究所 北京 中国 100093

²中国科学院大学 网络空间安全学院 北京 中国 100049

摘要 软件定义网络(Software-Defined Networking, SDN)通过控制平面与数据平面的分离和逻辑集中的控制构建了新的网络范式。考虑性能、可扩展性和可靠性等方面的需求,大规模网络通常采用分布式 SDN 控制平面,即通过放置多个控制器共同管理整个网络。这需要确定控制器的放置数量、放置位置以及交换机到控制器的分配,从而实现网络的性能、可靠性、成本与能耗等方面的优化,这就是控制器放置问题。本文系统研究了分布式 SDN 控制器放置问题。首先,对控制器放置问题的各种优化指标进行详细介绍;其次,根据优化目标以及目标之间的权衡方式对已有研究提出的控制器放置方案进行归类、分析与总结;最后,讨论控制器放置问题的未来研究趋势。

关键词 软件定义网络; 控制器放置问题; 性能; 可靠性; 成本与能耗
中图分类号 TP393.0 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2021.03.04

A Survey of Distributed SDN Controller Placement Problem

CAI Ning^{1,2}, HAN Yanni^{1,2}, AN Wei¹, XU Zhen¹

¹Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

²School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract Software-Defined Networking (SDN) builds a new network paradigm through the separation of the control plane from the data plane and the logically centralized control. Considering the requirements of performance, scalability and reliability, large-scale networks usually adopt a distributed SDN control plane, i.e., placing multiple controllers to manage the whole network. In order to optimize the performance, reliability, cost and energy consumption of the network, it is necessary to determine the number and locations of controllers and the assignment between switches and controllers; that is known as controller placement problem. This paper systematically studies distributed SDN controller placement problem. Firstly, the various optimization metrics of controller placement problem are introduced in detail. Secondly, the controller placement schemes proposed by existing researches are classified, analyzed and summarized according to the optimization objectives and the trade-offs among these objectives. Finally, we discuss the future research trends in controller placement problem.

Key words Software-Defined Networking; controller placement problem; performance; reliability; cost and energy consumption

1 引言

计算机网络通常包含大量的交换机、路由器以及防火墙、网络地址转换器和负载均衡器等多种类型的网络中间件。随着近年来互联网技术的快速发展和互联网业务需求日益增长,网络规模和数据流量的不断增长使得网络的管理和控制越来越复杂。特别是,随着网络虚拟化和云计算的广泛部署以及

移动计算趋势的兴起,网络环境变得更加动态,从而使流量模式和网络状况以更快速和显著的方式发生变化。

在传统网络架构中,每个网络设备不仅转发数据包,还要计算和维护其操作所需的状态;网络运营商需在命令行界面环境中使用有限的一组低级设备配置命令来执行日益复杂的管理策略和任务。现有传统网络架构采用控制与转发紧耦合的网络管理

通讯作者: 韩言妮, 副研究员, Email: hanyanni@iie.ac.cn。

本课题得到国家自然科学基金(No. 61771469)资助, 青年之星人才计划(No. Y7Z0091105)资助。

收稿日期: 2019-01-28; 修改日期: 2019-04-10; 定稿日期: 2020-12-20

方式,已经难以适应计算机网络的复杂性和动态性发展需要,具体面临如下4个问题:

(1) **异构设备增加了网络管理的复杂性:** 为了执行各种网络管理策略,并响应可能发生的各种网络事件(例如流量突变、网络攻击等),网络运营商需要为网络中大量的异构网络设备进行配置。每种设备都使用各自专用的配置工具和配置命令,并根据特定协议运行,因此增加了网络管理的复杂性。

(2) **频繁地人为配置增加了网络管理的复杂性:** 由于网络规模的快速增长和网络状态的不断变化,运营商需要不断地人工调整网络配置以响应不断变化的网络状况。为此,运营商借助外部工具或构建临时脚本以在事件发生时动态地重新配置网络设备。这种人为的频繁配置加剧了配置的复杂性,且容易引入配置错误^[1]。

(3) **分布式的低级配置方式增加了网络管理的复杂性:** 每个网络设备的控制平面与其数据平面紧耦合,且控制平面使用低级的配置命令进行设备配置,这种分布式的网络管理方式使得运营商需要将网络范围的高级管理策略转化为分布式的低级设备配置命令,随着日益复杂的网络策略的不断增加,网络管理也越来越复杂。

(4) **网络设备的专用配置方式和封闭式开发阻碍了网络的发展与创新:** 网络设备使用专用的配置工具和封闭式的开发环境,设备配置方法和命令的更新通常由设备供应商单方面决定,导致新协议和新技术的引入与部署变得极为困难,阻碍了网络的发展与创新。

由于传统网络专用的、分布式的、低级的、封闭的网络管理方式难以适应日益复杂和变化的网络发展趋势,网络的管理、发展和创新变得越来越困难。

1.1 软件定义网络

为了解决上述问题,研究人员提出了一种新型网络范式——软件定义网络(Software-Defined Networking, SDN)。SDN 将控制平面功能从各个网络设备移动到集中的控制器软件,网络设备只保留简单的分组转发功能。图 1 是由开放网络基金会(Open Networking Foundation, ONF)提出的 SDN 经典三层架构,包括数据平面、控制平面和应用平面。

(1) **数据平面**也被称为转发平面,由支持 SDN 南向接口的网络设备组成,用于转发数据包。网络设备可以是 SDN 硬件交换机、SDN 虚拟交换机或支持 SDN 的路由器等,后文统称为交换机。数据平面设备通过南向接口接收控制平面的控制器下发的控制

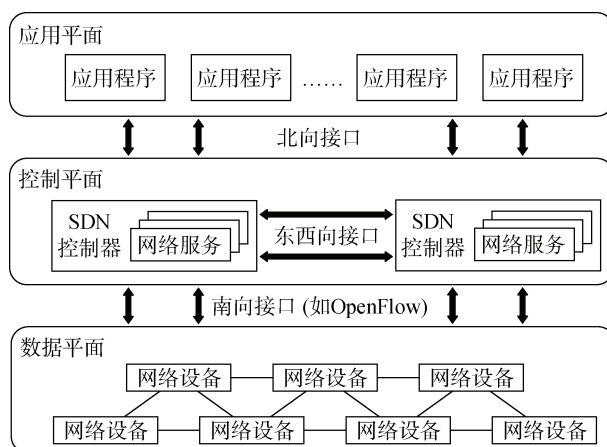


图 1 SDN 架构

Figure 1 SDN architecture

策略,并以流表项的形式存储于本地。当数据平面的数据包到达时,网络设备根据本地存储的流表项对数据包进行转发与处理,当找不到匹配的流表项时,网络设备会向控制器发送请求消息,由控制器决定如何处理该数据包并将处理指令下发给相关网络设备。数据平面设备还通过南向接口向控制平面的控制器主动上报设备状态和网络状态等信息。南向接口为上层提供统一的网络资源控制接口以简化网络资源控制工作,在多种南向接口协议中,OpenFlow^[2]受到了最为广泛的关注与使用,因此本文也主要关注基于 OpenFlow 的 SDN 解决方案。

(2) **控制平面**由集中的控制器软件组成,负责维护全网信息,为应用层提供全局网络抽象视图,同时将应用层的要求转换并下发至数据平面设备。控制平面通常使用单个控制器,这种控制平面被称为集中式 SDN 控制平面。由于可扩展性和可靠性等方面的需要,开发了分布式 SDN 控制平面,即使用多个控制器共同管理整个网络,控制器之间通过东西向接口进行通信以维持全网视图。控制器中内置了多种网络服务模块用于执行基本的网络资源管控。控制平面通过北向接口与应用平面中的应用程序进行通信,向应用层程序开放对底层网络资源的控制能力。

(3) **应用平面**包含使用 SDN 控制器北向接口实现的应用程序,这些应用程序通过组合调用北向接口,实现比控制器内置网络服务更加复杂和自动化的网络服务,包括路由、流量工程、负载均衡、防火墙和网络状态监测等。应用程序依据控制平面提供的全网视图,不需依赖于底层网络基础设施的执行细节,以集中的方式直接表达所需的目标和策略,由控制器接收并转化为南向特定的命令和指示各数

据平面设备行为的转发规则。

SDN 控制器具有网络的全局视图, 能够根据当前的网络状态(例如链路利用率、网络故障等)快速地做出恰当的路由决策, 提高网络资源利用率, 实现更高效的网络控制和管理; SDN 使用集中式的网络配置方式, 绕过底层网络细节的复杂性, 无需单独配置所有设备, 简化并加快了网络的配置; SDN 通过开放可编程接口, 提供自动化的网络管理能力, 降低网络设计、部署、管理及扩展开销, 减少运维耗时以及人工失误, 从而降低网络运维成本; 开发人员还可通过可编程接口快速部署各种新协议和应用程序, 加速网络的发展与创新。

1.2 集中式 SDN 控制平面

集中式 SDN 控制平面使用单个控制器(例如 NOX^[3]、Floodlight^[4]、Ryu^[5]和 Beacon^[6]等), 是 SDN 早期的部署方式, 主要应用于校园网和小型企业网, 因此集中式控制平面能够满足需求。但当 SDN 推广到数据中心和广域网(Wide Area Network, WAN)等大规模网络时, 集中式 SDN 控制平面在性能、可扩展性和可靠性等方面遇到如下挑战:

(1) **性能:** SDN 中控制平面与数据平面相分离, 交换机不再具有对新流做出决策的能力, 需要与控制器建立通信以接收关于新流的处理决定。此外, 控制器也会根据当前的网络状态以及上层应用程序制定的控制策略主动地下发转发规则。控制器与距离较远的交换机之间通信可能需要过多的时延, 这在地理分布的 WAN 中尤为严重。控制平面与数据平面之间较高的通信时延会降低控制平面的决策速度和数据平面的响应速度^[7], 也限制了期望的故障恢复能力以及横向的扩展^[8]。

(2) **可扩展性:** 随着网络规模的不断扩大和业务需求的不断增多, 控制器需要应对越来越多的网络事件和流请求, 这可能会使控制器由于其计算、内存和带宽等资源的限制成为瓶颈, 出现控制器过载的情况, 这在数据中心和大型企业网等大规模网络中尤为严重^[9]。例如, NOX 控制器^[3]每秒可处理 3 万个流请求^[7], 而具有 1500 台服务器的数据中心每秒平均产生 10 万条流^[10], 具有 100 个交换机的网络每秒最多可以产生 1000 万个流请求^[6], 远远超过了 NOX 控制器的处理能力。控制器的负载过高会增加控制器的处理时延, 影响网络的性能。为了解决该问题, 研究人员试图限制向 SDN 控制器发送流请求的数量, 例如, 将部分控制功能返还给交换机^[11-12], 这种方式虽然可以一定程度上提高网络的可扩展性, 但需要修改交换机的硬件且较为复杂, 并且该方式违背了

SDN 控制与转发分离的设计初衷。

(3) **可靠性:** 单个控制器本质上具有单点故障问题, 且易受拒绝服务攻击的影响。当控制平面失效时, 交换机将失去转发新流和应对网络其他动态变化的能力, 降低服务质量, 最终可能导致网络瘫痪。值得注意的是, 即使控制器正常运行, 由于链路或交换机的故障, 某些网络设备还是会与控制器发生断连。因此, 单控制器与转发平面的分层架构对网络的可靠性产生了重大影响。

集中式 SDN 控制平面不能保证大规模网络在性能、可扩展性和可靠性等方面的需求。

1.3 分布式 SDN 控制平面

为了增强在大规模网络部署时 SDN 的性能、可扩展性和可靠性, 研究人员提出了分布式 SDN 控制平面, 它使用多个物理分布、逻辑集中的控制器共同管理整个网络。逻辑集中是指全部或部分控制器具有全局的网络视图, 从而保证做出全局最优的控制决策。

在分布式 SDN 控制平面中, 控制器之间使用东西向接口进行通信。东西向接口(1)用于控制器之间共享本地网络状态信息从而构建全局网络视图; (2)用于控制器故障转移机制, 实时检测控制器的存活情况并做好数据备份; (3)用于控制器之间的负载均衡机制, 平均分配网络负载, 使网络获得最佳性能和效率。东西向接口若使用专用的链路传递消息则称为带外连接方式; 若使用网络中已有的链路传输消息则称为带内连接方式。

根据控制平面的物理组织结构, 分布式 SDN 控制平面大体分为扁平模型和分层模型两种类型, 如图 2 所示。

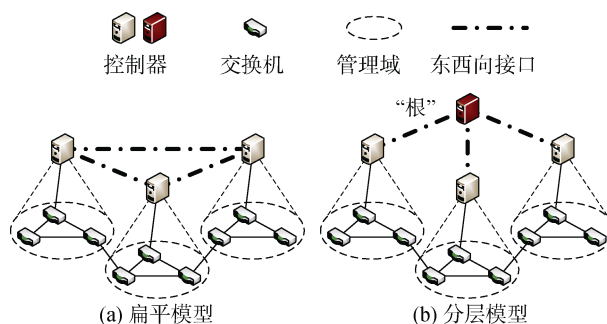


图 2 分布式 SDN 控制平面

Figure 2 Distributed SDN control plane

(1) **扁平模型:** 该模型将网络水平划分为多个区域, 每个区域由一个控制器负责管理, 该域称为该控制器的管理域。为了实现逻辑集中的控制, 控制器之间通过东西向接口进行通信, 交换各自管理

域的状态信息,从而维护全局的网络视图。典型的扁平 SDN 控制平台有 Onix^[13]、HyperFlow^[14]和 ONOS^[15]等。

(2) 分层模型: 该模型将控制平面根据所需服务垂直划分为多个层次。底层由本地控制器组成,本地控制器之间不相互连接,且只处理本地应用程序的需求。上层由通常称为“根”的控制器构成,根控制器与底层控制器相连,处理非本地应用的需求。与扁平模型不同的是,分层模型的底层控制器仅具有本地网络视图,只有上层控制器才具有网络的全局视图。典型的分层 SDN 控制平台有 Kandoo^[16]、谷歌的 B4^[17]和 Espresso^[18]等。

以上两种模型以及各种 SDN 控制平台的详细介绍可参考相应的综述文献[19-20]。

分布式 SDN 控制平面相比集中式 SDN 控制平面具有如下优势: (1)地理位置分布的控制平面可以降低控制器与交换机间的通信时延,提高网络性能; (2)多控制器可以解决单控制器的处理瓶颈,提高可扩展性; (3)多控制器可以解决单点故障问题,并可通过执行各种故障检测和恢复机制进一步提高网络的可靠性。例如, HyperFlow^[14]使用分布式存储系统 WheelFS^[21]提供的故障检测机制检测控制器故障,当某个控制器发生故障后将受影响的交换机重分配到临近的控制器; ONOS^[15]在网络构建时就将每个交换机连接到多个控制器来应对控制器故障。

1.4 控制器放置问题

分布式 SDN 控制平面在过去几年受到广泛关注并发展迅速,研究人员开发了多种分布式 SDN 控制平台^[13-18]解决了分布式 SDN 控制平面的设计与实现问题,但是分布式控制平面在实际部署时还需要考虑以下几个问题: (1)使用多少个控制器实例; (2)控制器实例的放置位置; (3)每个控制器管理哪些交换机。这些问题统称为控制器放置问题(Controller Placement Problem, CPP)。

控制器的放置对 SDN 的性能、可靠性和成本等方面产生重要影响。例如,控制器的放置位置和交换机到控制器的分配会影响控制平面与数据平面之间的时延;控制器与交换机之间的连接以及备用路径和备用控制器的选取也会影响网络的可靠性;控制器的数量以及交换机到控制器的分配和连接也会影响网络的成本与能耗。

控制器放置问题由 Heller 等人^[22]首次提出,他们研究了 WAN 中控制器的数量和位置对控制器与交换机之间传播时延的影响,目标是最小化传播时延,并通过将该问题等同于选址问题^[23],证明了

CPP 是 NP 难问题。WAN 是使用分布式 SDN 控制平面的一个重要场景,并且 WAN 的网络拓扑相比数据中心等网络的拓扑更加复杂和多样,研究人员提出了多种 WAN 中的控制器放置方案。因此,本文主要关注 WAN 中的控制器放置问题,具体模型如下:

WAN 拓扑由一个无向图 $G(V, E)$ 表示,其中, V 表示节点(即交换机)集合, E 表示边(即链路)集合。距离矩阵 D , $d_{v,u} \in D$ 表示节点 v 和 u ($v, u \in V$) 之间的最短距离,通常使用最短路径时延或者最小跳数表示。 k 表示控制器数量, P 表示 k 个控制器的放置位置集合,所以 $k = |P|$,控制器的放置位置满足 $P \subseteq V$,即放置于网络中的节点位置。控制器放置问题是在给定的网络 G 中,确定控制器的数量 k 、控制器的放置位置 P 以及交换机到控制器的分配 $A: V \rightarrow P$,从而使得预定义的目标函数 $f_i (i \in \{1, \dots, J\})$ 达到最优。

本文对控制器放置问题进行深入调研,主要贡献总结如下:

- (1) 介绍了 SDN 以及分布式 SDN 控制平面的优势及面临的挑战,并阐述控制器放置问题的重要性;
- (2) 归纳总结了控制器放置问题的优化指标,并从含义、重要性、分类、计算方法和影响因素等方面对各指标进行详细分析;
- (3) 根据多目标的权衡方式对已有研究提出的控制器放置方案进行归类、分析与总结;
- (4) 总结了控制器放置问题中仍然存在的问题,为该领域的后续研究提供参考。

其余部分的组织结构如下:第 2 节介绍了控制器放置问题的优化指标;第 3 节对已有的控制器放置方案进行分类和详细介绍;第 4 节给出未来的研究展望;第 5 节总结全文。

2 优化指标

控制器放置问题的优化目标包括最大化性能(主要指最小化时延,包括制器与交换机间时延和控制器间时延)、优化控制器负载、最大化可靠性以及最小化成本与能耗(包括部署成本、管理成本和能量消耗)等多个方面。本节从定义、重要性、分类、计算方法和影响因素等方面对相关指标进行详细介绍。

2.1 时延

网络信息的收集和指令的下发等多种 SDN 网络管控功能的执行都需要控制器与交换机之间频繁的通信,控制器之间也需要经常通信以进行状态同步和管理协调。控制器与交换机间时延和控制器间时延对 SDN 网络的性能有重要影响。时延主要包括以

下四部分:

(1) **传输时延**是指发送端将数据包发送到链路上的时间, 通常表示为数据包的长度与链路带宽的比值, 其中, 链路带宽由最小传输速率的网络接口决定。

(2) **传播时延**是指数据从链路的源端传播到目的端的时间, 可表示为链路的长度与链路中信号的传播速度的比值。

(3) **排队时延**包括交换机的排队时延和控制器的排队时延。交换机的排队时延是指数据包进入交换机之后在输入队列中等待处理以及在确定转发端口之后在输出队列中排队等待转发的时间。控制器的排队时延是指数据包进入控制器后等待处理的时间。排队时延与设备的负载和处理性能有关。

(4) **处理时延**包括控制器处理时延和交换机处理时延, 指设备收到数据包后进行分析头部、提取数据和查找适当路由等操作的时间, 由设备的负载与性能决定。

以上四类时延在不同类型的网络中具有不同的权重。在地理位置集中且带宽较小的局域网中, 传输时延占主导地位, 而传播时延可以忽略不计。在 WAN 中, 网络跨越较广的地理范围, 并使用具有高性能线速转发能力的交换机和高带宽容量的链路, 且流路径请求消息的长度相对较小, 因此传输时延、排队时延和处理时延可以忽略不计, 传播时延是主要因素, 通常使用传播时延作为控制器与交换机间时延以及控制器间时延的度量。

2.1.1 控制器与交换机间时延

控制器与交换机间时延会影响控制平面的决策速度和数据平面的响应速度, 进而影响业务, 尤其是实时业务(例如链路故障的发现与恢复), 因此需要最小化控制器与交换机间时延。控制器与交换机间时延包括以下 5 个指标:

(1) 无故障时平均传播时延

无故障时控制器与交换机间平均传播时延是指控制器与交换机间传播时延的均值, 如式(1)所示^[22,24-25], 该式隐含地假设交换机到控制器的分配基于最近分配。若明确指定了交换机到控制器的特定分配, 平均传播时延需要按照式(2)^[25]进行计算, 其中 $A(v)$ 表示为交换机 v 分配的控制器。优化目标是从所有可能的放置方案中寻找最优的放置使平均传播时延最小。平均时延隐藏了最大时延, 可能出现某些交换机与控制器间时延过大的情况。

$$L_{avg}(P) = \frac{1}{|V|} \sum_{v \in V} \min_{p \in P} d_{v,p} \quad (1)$$

$$L'_{avg}(P) = \frac{1}{|V|} \sum_{v \in V} \min_{p \in P} d_{v,A(v)} \quad (2)$$

(2) 无故障时最大传播时延

无故障时控制器与交换机间最大传播时延是指控制器与交换机间传播时延的最大值。与平均传播时延的计算类似, 式(3)^[22,24,26-28]基于最近分配, 而式(4)^[29]是更为一般的计算公式。最大传播时延反映了时延的最差情况, 能够保证所有时延不超过该最大值。因此最大传播时延常用来作为优化指标(即最小化最大传播时延)或作为其他优化目标的约束条件(例如在满足最大传播时延约束下最小化成本)。

$$L_{max}(P) = \max_{v \in V} \min_{p \in P} d_{v,p} \quad (3)$$

$$L'_{max}(P) = \max_{v \in V} d_{v,A(v)} \quad (4)$$

(3) 控制器故障时平均传播时延

当控制器发生故障时, 可以使用故障转移机制, 例如基于正常最短路径路由的备份分配, 将之前连接到该故障控制器的所有交换机重新分配到距这些交换机第二近的控制器。但是将交换机分配到新的控制器可能会导致更高的时延, 因此在控制器放置还应考虑控制器发生故障时的时延情况。考虑最多 $k-1$ 个控制器故障的所有可能场景(包括无故障场景), X 表示这些场景中无故障控制器的放置, 则控制器故障时控制器与交换机间平均传播时延如式(5)所示^[24]。

$$L_{avg}^X(P) = \frac{1}{|X|} \sum_{P \in X} \left(\frac{1}{|V|} \sum_{v \in V} \left(\min_{p \in P} d_{v,p} \right) \right) \quad (5)$$

(4) 控制器故障时最大传播时延

与无故障时的平均传播时延类似, 控制器故障时的平均传播时延反映的是整体情况, 会掩盖时延的最差情况, 因此需要考虑控制器故障时的最大传播时延, 如式(6)所示^[24,27]。

$$L_{max}^X(P) = \max_{P \in X} \max_{v \in V} \min_{p \in P} d_{v,p} \quad (6)$$

(5) 交换机或链路故障时最大传播时延

不同的交换机或链路故障构成不同的故障场景, 所有故障场景与无故障场景组成了网络场景集合, 由 T 表示。网络场景 t 发生的概率为 r_t , 交换机或链路故障时控制器与交换机间最大传播时延如式(7)所示^[30]。由于交换机或链路故障可能会导致某些交换机与所有控制器发生断连, 这些交换机与控制器间的时延可视为无限大, 为了便于分析, 为该情况增加一个惩罚性的时延值^[30]。

$$L_{max}^T(P) = \sum_{t \in T} \max_{v \in V} d_{v,A(v)} r_t \quad (7)$$

2.1.2 控制器间时延

在分布式 SDN 控制平面中, 控制器之间: (1)需要经常通信来获得全局一致的网络状态信息, 确保做出正确的管理决策; (2)需要进行协调以进行跨域的操作; (3)需要状态同步和协调, 用于网络故障的发现与恢复。因此, 需要最小化控制器间的时延以提高网络管理效率。控制器间的平均传播时延和最大传播时延分别如式(8)^[24]和(9)^[24,27]所示。

$$I_{avg}^{controller}(P) = \frac{1}{\left(\frac{|P|}{2}\right)} \sum_{p_1, p_2 \in P} d_{p_1, p_2} \quad (8)$$

$$I_{max}^{controller}(P) = \max_{p_1, p_2 \in P} d_{p_1, p_2} \quad (9)$$

2.2 控制器负载

控制器的负载主要包括四部分^[29]: (1)处理交换机的流路径请求事件(简称流路径请求负载); (2)维护本地网络视图; (3)与其他控制器通信获得全网视图; (4)将应用程序生成的策略转换成流表规则并下发至交换机。流路径请求负载通常被认为是总负载中最重要的部分, 其使用交换机的流路径请求率进行度量。

控制器的负载过大将显著增加控制器的处理时延^[7], 进而增加控制器与交换机之间的总通信时延。高负载的控制器由于缺少处理故障的资源, 更加可能遭受攻击, 因此具有更高的故障概率, 甚至可能导致其他控制器的级联故障^[31]。在控制器放置时要合理地给控制器分配负载, 相关指标包括控制器能力约束和控制器负载均衡两大类。

2.2.1 控制器能力约束

受处理器、内存和带宽等资源的限制, 控制器的处理能力是有限的。控制器的负载不应超过其能力^[29,32-34], 即交换机到控制器的分配应满足控制器能力约束, 如式(10)所示^[29], 其中 l_v 表示交换机的流路径请求率, $D(p)$ 表示控制器 p 所管理的所有交换机。为了与考虑备用的控制器能力约束相区分, 该约束也被称为不考虑备用的控制器能力约束。

$$\sum_{v \in D(p)} l_v \leq \alpha_p \quad (10)$$

为了应对突发流量或由于故障转移机制导致的负载增加, 可以为控制器预留一部分能力作为备用^[35]。预留比例由 β_p 表示, 则考虑备用的控制器能力约束如式(11)所示。

$$\sum_{v \in D(p)} l_v \leq (1 - \beta_p) \alpha_p \quad (11)$$

2.2.2 控制器负载均衡

控制器放置时, 不但要避免控制器负载超过其

能力, 还应该很好地平衡控制器间的负载分配, 保证控制平面高效地运行。与控制器负载均衡相关的指标包括以下四个。

(1) 考虑负载均衡的控制器负载约束

控制器负载分配最理想的情形是每个控制器都分配数量相同的负载, 但由于交换机可能具有不同的流路径请求率, 且控制器的负载通常是以交换机为单位进行分配, 很难使得各控制器间的负载完全均衡。因此, 可以利用考虑负载均衡的控制器负载约束来限制控制器的负载分配, 如式(12)所示^[36], τ 作为调节因子对约束进行微调。

$$\sum_{v \in D(p)} l_v \leq \frac{1}{k} \sum_{v \in V} l_v + \tau \quad (12)$$

(2) 无故障时控制器间负载最大差值

考虑负载均衡的控制器负载约束只设置了控制器负载的上界。在控制器放置过程中, 不仅要避免出现一部分控制器负载过大的情况, 还要避免一部分控制器负载过低, 以提高控制平面的资源利用率。控制器负载的不均衡情况使用控制器的最大负载与最小负载的差值来描述, 如式(13)所示^[24,25,27], 这里的目标是最小化控制器间负载最大差值。

$$B(P) = \max_{p \in P} n_p - \min_{p \in P} n_p \quad (13)$$

(3) 控制器故障时控制器间负载最大差值

当控制器发生故障时, 故障控制器管理的交换机会通过故障转移机制分配到其他正常运行的控制器。交换机到控制器的重新分配会影响控制器的负载。控制器故障时控制器间的负载情况也由负载最大差值反映, 如式(14)所示^[24,27]。其中, Y 表示最多 $k-2$ 个控制器发生故障时的无故障控制器。优化目标是最小化正常运行的控制器间的负载最大差值。

$$B^Y(P) = \max_{P \in Y} \left(\max_{p \in P} n_p - \min_{p \in P} n_p \right) \quad (14)$$

(4) 控制器负载率方差

控制器间负载最大差值反映的是控制器负载均衡情况, 但却没有考虑控制器的能力约束。将控制器负载与控制器能力的比值定义为控制器利用率, 也即控制器负载率^[37], 用来反映控制器的负载占用情况并设置其最大约束为 1。使用所有控制器负载率的方差来表示考虑控制器能力的控制器的负载均衡情况, 如式(15)所示。

$$BRV(P) = \frac{1}{k} \sum_{p \in P} \left(\frac{n_p}{\alpha_p} - \frac{1}{k} \sum_{p \in P} \frac{n_p}{\alpha_p} \right)^2 \quad (15)$$

2.3 可靠性

在 SDN 中, 交换机与控制器通过标准的传输层

安全协议(TLS)或传输控制协议(TCP)连接进行通信, 控制器之间也需要连接通信以进行控制平面的协调。交换机与控制器以及控制器之间的连接路径统称为控制路径, 所有控制路径构成 SDN 的控制网络^[13]。

网络运行过程中, 由于软件缺陷、恶意攻击、配置错误、硬件故障和断电等原因会导致网络组件发生故障, 可能会使部分控制器路径发生故障。控制路径故障可能会造成控制平面与数据平面的断连, 从而使控制平面失去对部分数据平面的管控能力, 可能会导致数据平面严重的丢包和性能下降。此外, 控制路径故障也可能会导致控制平面中控制器之间的断连, 从而影响控制器之间的协同。因此, 在控制器放置时应该最大化控制网络的可靠性, 进而提高整个 SDN 的可靠性和性能。

网络组件故障主要分为两种类型: **(1)数据平面故障**包括链路故障和交换机故障。链路故障表示经过故障链路的流量不能再通过该链路进行传输, 交换机故障表示该交换节点无法发送、响应或转发任何数据包。**(2)控制平面故障**是指控制器实例不再具有任何控制功能。

网络组件发生故障后, 可以通过故障转移机制避免或减小故障对控制网络造成的影响。故障转移机制主要包括以下两种: **(1)备用路径**: 交换机与分配的控制器之间使用多条路径进行连接, 在主控制路径发生故障后(由数据平面故障导致), 交换机可以通过备用路径与控制器进行连接, 从而维持控制平面与数据平面的正常通信, 提高控制网络的可靠性。可以使用到达控制器的任何路径作为备用路径(例如节点不相交路径、最短路径或最宽路径)。**(2)备用控制器**: OpenFlow 1.2 提出多控制器方案, 为控制器赋予了不同角色, 使交换机可以通过主/从连接的方式连接到多个控制器。当交换机与主控制器失去连接时(主控制器发生故障或备用路径全部失效), 备用(从)控制器可以接管该交换机, 保证对网络的正常管控。备用控制器可以是多个控制器, 备用控制器构成的备用控制器列表指定了交换机连接到这些控制器实例的顺序。

根据是否使用故障转移机制, 控制网络的可靠性分为无故障转移机制的可靠性和有故障转移机制的可靠性两类。

2.3.1 无故障转移机制的可靠性

无故障转移机制的可靠性指不考虑故障转移机制时控制网络正常运行的能力。在网络中, 交换机、链路或控制器都有一定的故障概率, 进而会导致控

制路径故障。故障的控制路径越多, 对控制网络的影响越严重。因此在控制器放置时需要最小化控制路径的故障比例或概率, 从而提高控制网络的可靠性。无故障转移机制的可靠性包括以下 5 个指标:

(1) 故障控制路径预期比例

在物理网络中, 故障场景多种多样, 故障场景之间彼此独立, 每个故障场景都可能影响控制网络中部分控制路径的状态, 可以采用基于原因的可靠性分析模型^[38]进行可靠性分析。网络场景集合(包括无故障场景和所有故障场景)由 S 表示, 场景 s 发生的概率为 r_s , 且会导致 q_s 条控制路径中断。使用故障控制路径的预期比例来描述控制网络的可靠性, 如式(16)所示^[39], 目标是 minimized 故障控制路径预期比例, 其中 m 表示控制路径的总数。

$$\delta(P) = \frac{1}{m} \sum_{s \in S} q_s r_s \quad (16)$$

(2) 有效控制路径预期比例

有效控制路径预期比例是指在所有网络场景中无故障控制路径的平均比例, 如式(17)所示^[40], 优化目标是最大化有效控制路径预期比例, 与式(16)的关系是 $\varepsilon(P) = 1 - \delta(P)$ 。

$$\varepsilon(P) = 1 - \frac{1}{m} \sum_{s \in S} q_s r_s \quad (17)$$

(3) 控制路径故障概率

与前两个指标基于原因的分析不同, Zhang 等人^[41]分析单条控制路径的故障概率, 控制路径的故障概率与该路径中节点和链路的故障概率有关。 Z_{e_i} 表示链路 e_i 的故障概率, Z_{v_j} 表示节点 v_j 的故障概率, 节点 a 到 b 之间控制路径 $R_{a,b}$ 的故障概率 $Z_{a,b}$ 如式(18)所示^[41]。优化目标是 minimized 控制路径平均故障概率或最小化控制路径最大故障概率。

$$Z_{a,b}(P) = 1 - \prod_{e_i, v_j \in R_{a,b}} (1 - Z_{e_i})(1 - Z_{v_j}) \quad (18)$$

(4) 控制网络有效概率

控制路径故障概率反映的是单条控制路径的可靠性, 将所有控制路径正常运行概率的乘积定义为控制网络有效概率, 来评价整个网络的可靠性, 如式(19)^[37]所示, 目标是最大化控制网络有效概率。

$$Y(P) = \prod_{v \in V} \prod_{e_i, v_j \in R_{v,A(v)}} (1 - Z_{e_i})(1 - Z_{v_j}) \quad (19)$$

(5) 断连交换机数

交换机与控制器间的控制路径故障会使交换机与控制器之间发生断连, 交换机将不再受控制器的管控, 使用断连的交换机数来反映控制网络的可靠性。在所有网络场景 S 中, 平均断连交换机数和最大

断连交换机数分别如式(20)和(21)所示。其中, $e_{v,A(v)}^s$ 表示在场景 s 中, 交换机 v 是否与控制器 $A(v)$ 发生断连, 若是则为 1, 否则为 0, 目标是最小化平均或最大断连交换机数。

$$N_{avg}(P) = \frac{1}{|S|} \sum_{s \in S} \frac{1}{|V|} \sum_{v \in V} e_{v,A(v)}^s \quad (20)$$

$$N_{max}(P) = \max_{s \in S} \sum_{v \in V} e_{v,A(v)}^s \quad (21)$$

2.3.2 有故障转移机制的可靠性

有故障转移机制的可靠性是指使用故障转移机制时控制网络面对故障仍能维持正常运行的能力, 即维持交换机与控制器以及控制器之间连接的能力, 也被称为弹性、恢复性或生存性。有故障转移机制的可靠性的度量指标包括以下 4 种:

(1) 交换机与控制器间的连通性

交换机与控制器间的连通性是指交换机与控制器间不相交路径数, 这些路径均可以作为交换机与控制器间的备用路径。 $w_{p,v}$ 表示控制器 p 与交换机 v 之间不相交路径的数量, 若交换机 v 不属于控制器 p 的管理域, 则该值为 0。交换机与其分配的控制面间不相交路径数的平均值如式(22)所示^[35]。交换机与控制器间的连通性越大, 表示可用的备用路径数越多, 从多种故障场景中维持交换机与控制平面连通的可能性越大, 因此在控制器放置时要最大化交换机与控制器间的连通性。

$$\pi(P) = \frac{1}{|V|} \sum_{p \in P} \sum_{v \in V} w_{p,v} \quad (22)$$

(2) 控制路径平均长度

为了减少故障转移机制的执行时间, 备用路径需要提前进行配置, 且主控制路径和备用控制路径的选择需要同时优化, 控制路径(包括主控制路径和备用控制路径)的平均长度如式(23)所示^[42], 其中 $u_{v,A(v)}$ 表示交换机 v 与其分配的控制面之间的主控制路径的长度, $m_{v,A(v)}$ 表示交换机 v 与其分配的控制面之间的备用控制路径的长度。目标是最小化控制路径平均长度。考虑到传播时延与链路长度是线性关系, 最小化控制路径长度等同于最小化控制器与交换机间时延。链路的故障概率通常也与链路长度成正比, 因此最小化控制路径长度也可视为最小化控制路径的故障概率。

$$\frac{1}{|V|} \sum_{v \in V} (u_{v,A(v)} + m_{v,A(v)}) \quad (23)$$

(3) 故障控制路径预期比例

当网络发生故障时, 即使使用故障转移机制,

还是可能出现交换机与控制器或控制器之间没有可用连接的情况, 将这种情况称为广义的控制路径故障。考虑所有网络场景 S , p_s 表示受场景 s 中的网络故障影响的控制路径的条件故障概率, 与使用的故障转移机制有关, θ_s 表示以故障节点为端点的控制路径数量, ϕ_s 表示包含故障节点(非端点)的控制路径数量, 使用故障转移机制时故障控制路径的预期比例如式(24)所示^[43]。目标是最小化故障控制路径预期比例。

$$\gamma(P) = \sum_{s \in S} r_s \frac{\theta_s + \phi_s p_s}{m} \quad (24)$$

(4) 无控制器节点数

控制器故障时, 受影响的交换机会连接到备用控制器, 只要存在至少一个正常运行的控制器, 数据平面都可以维持与控制平面之间的连接。当数据平面发生故障时, 网络拓扑发生改变, 可能导致某些交换机与控制平面发生断连。这些断连的交换机(至少与另一个断连交换机相连)虽仍然可以继续转发网络中已有的流量, 但是不能再转发新的流量或接收控制器的其他指令, 这种交换机被称为无控制器节点。对于特定的控制器放置 P , 考虑所有网络场景 S , 最大无控制器节点数如式(25)所示^[24,27], 其中, E^s 为连通性矩阵。在场景 s 中, 若交换机 v 能到达控制器 p , 则矩阵 E^s 的项 $e_{v,p}^s$ 为 1, 否则为 0。优化目标是最大化最大无控制器节点数。

$$N(P) = \max_{s \in S} \sum_{v \in V} \min_{p \in P} e_{v,p}^s \quad (25)$$

2.4 成本与能耗

在控制器放置时还需要考虑部署成本、管理成本和能量消耗等指标, 这些对于分布式 SDN 的实际部署和商用非常重要。

2.4.1 部署成本

控制器的数量影响资本支出(CAPEX)与运营支出(OPEX), 因此大多数研究考虑的是使用较少的控制器实现特定的优化目标。对于采用带外连接方式的控制网络, 还需要考虑链路的成本。控制器与链路的部署成本根据使用场景分为以下两类:

(1) 静态部署成本

静态部署成本 C 指控制网络构建初期的成本, 包括控制器的安装成本 C_c 、控制器与交换机之间链路的安装成本 C_l 以及控制器之间链路的安装成本 C_t , 如式(26)所示^[32]。其中, 控制器的安装成本与控制器的价格有关, 链路的安装成本与链路的单价和链路的长度有关。优化目标是最小化静态部署成本。

$$C(P) = C_c + C_l + C_t \quad (26)$$

(2) 网络扩张成本

为了满足用户和业务增长的需要, 网络需要不断扩张, 即为数据平面添加更多的网络设备。为此, 控制网络也需要进行调整以满足网络控制在性能和可靠性等方面的需求。控制网络变化的成本定义为网络扩张成本 C' , 包括控制器的安装和移除成本 C'_c 、控制器与交换机间链路的安装或移除成本 C'_l 以及控制器之间链路的安装或移除成本 C'_t , 如式(27)所示^[34]。目标是最小化网络扩张成本。

$$C'(P) = C'_c + C'_l + C'_t \quad (27)$$

2.4.2 管理成本

管理成本主要是指网络管理的通信成本, 包括静态管理成本和动态管理成本两类。

(1) 静态管理成本

静态管理成本是指控制网络不发生改变时网络管理的通信成本, 主要包括以下三部分:

统计信息收集成本 C_v : 控制器需要定期地收集其管理域内交换机的端口、流表等统计信息。收集成本与交换机和控制器间的距离(通常由时延或跳数表示)以及查询交换机信息的比特率有关。

流路径设置成本 C_q : 当新流进入交换机时, 由于没有匹配的转发规则, 交换机会向控制器发送路径设置请求消息, 控制器计算该流的转发路径并将相应的转发规则下发至相关交换机。因此, 流路径设置成本包括初始路径设置请求成本和规则安装成本。若该流进入其他控制器的管理域, 还需要再次执行路径设置请求, 将该成本称为中间路径设置请求成本。则流路径设置成本是以上三个成本之和, 与交换机的流路径请求率、请求消息、流表安装消息以及交换机与控制器间的距离有关。

控制器同步成本 C_s : 指控制器之间信息同步的成本, 与控制器之间的距离、同步数据量以及同步策略有关。

静态管理成本 C_O 是上述三类成本的加权和, 如式(28)所示^[44]。权重由运营商根据需求来决定, 优化目标是最小化静态管理成本。

$$C_O(P) = a_1 C_v + a_2 C_q + a_3 C_s \quad (28)$$

(2) 动态管理成本

动态管理成本是指控制网络发生变动时的通信成本。由于网络流量变化等原因需要实时调整交换机到控制器之间的分配。交换机到控制器的重分配成本由 C_r 表示, 与交换机到新分配的控制器之间的距离有关。动态管理成本 C'_O 是静态管理成本与重分

配成本的加权和, 如式(29)所示^[33], 优化目标是最小化动态管理成本。

$$C'_O(P) = a_1 C_v + a_2 C_q + a_3 C_s + a_4 C_r \quad (29)$$

2.4.3 能量消耗

降低控制网络的能耗不仅有利于减少运营成本支出, 而且还能减少碳排放有助于环境保护。控制网络的能耗主要包括控制器能耗和控制链路能耗。

(1) 控制器能耗

控制器的能耗与控制器的数量有关, 在控制器放置时应尽可能减少控制器的数量从而降低控制器能耗。此外, 可以根据网络中流量负载的实时变化动态地调整控制器的数量^[45], 在流量负载降低时减少运行控制器的数量从而降低控制器能耗。

(2) 控制链路能耗

控制链路能耗与控制网络中使用的链路数有关, 如式(30)所示^[46], 优化目标是最小化控制链路的能耗, 其中, 二元变量 χ_e 表示链路 e 是否属于控制链路, 若是则为 1, 否则为 0, ρ_e 表示链路 e 的能量消耗。

$$Q(P) = \sum_{e \in E} \rho_e \chi_e \quad (30)$$

为了进一步减少控制链路能耗, 在网络运行的非高峰时期, 可以将某些交换机端口置为睡眠模式以使对应的链路进入睡眠状态^[47]。

2.5 小结

SDN 控制器放置问题的优化指标及其主要影响因素如表 1 所示, 所有优化指标均与网络的拓扑有密切的关系。控制器与交换机间时延以及控制器间时延的指标包括平均时延和最大时延两大类。平均时延反映时延的整体情况, 但隐藏了最差的情况, 而最大时延可以保证每个控制器与交换机间的时延或控制器间的时延符合预期的需求。WAN 中的时延通常只考虑传播时延, 在后文中如果不做特别说明, 时延泛指传播时延, 且指无故障时的传播时延。并且控制器能力约束泛指不考虑备用的控制器能力约束。接下来, 首先分析这些优化指标之间的内在关系, 然后对已有的 SDN 控制器放置方案进行归类并详细介绍。

3 控制器放置方案

SDN 控制器放置问题通常是一个多目标优化问题, 且大多数目标之间具有竞争关系。竞争关系是指对一个目标进行优化时, 优化过程会对另一个目标产生负面影响。由于竞争关系的存在, 不存在对所有

表 1 SDN 控制放置问题的优化指标及其主要影响因素

Table 1 Optimization indexes and main influencing factors of SDN controller placement problem

优化指标		主要影响因素
时延	无故障时平均传播时延	控制器与交换机间距离
	无故障时最大传播时延	控制器与交换机间距离
	控制器故障时平均传播时延	控制器与交换机间距离
	控制器故障时最大传播时延	控制器与交换机间距离
	交换机或链路故障时最大传播时延	控制器与交换机间距离
控制器间时延	平均传播时延	控制器间距离
	最大传播时延	控制器间距离
控制器能力约束	不考虑备用的控制器能力约束	控制器能力, 连接到控制器的交换机个数和各交换机的流路径请求率
	考虑备用的控制器能力约束	控制器能力, 连接到控制器的交换机个数和各交换机的流路径请求率
	考虑负载均衡的控制器负载约束	连接到控制器的交换机个数和各交换机的流路径请求率
	无故障时控制器间负载最大差值	连接到控制器的交换机个数和各交换机的流路径请求率
	控制器故障时控制器间负载最大差值	连接到控制器的交换机个数和各交换机的流路径请求率
控制器负载均衡	控制器负载率方差	控制器能力, 连接到控制器的交换机个数和各交换机的流路径请求率
	故障控制路径预期比例	网络组件故障概率
	有效控制路径预期比例	网络组件故障概率
	控制路径故障概率	网络组件故障概率
	控制网络有效概率	网络组件故障概率
可靠性	断连交换机数	控制器与交换机间距离
	交换机与控制器的连通性	控制器与交换机间距离
	有故障转移机制的可靠性	控制器与交换机间距离, 故障转移机制
	故障控制路径预期比例	网络组件故障概率, 故障转移机制
	无控制器节点数	控制器与交换机间距离, 故障转移机制
成本与能耗	部署成本	控制器的类型、价格和数量, 链路的类型、价格和长度
	网络扩张成本	控制器的类型、价格和数量, 链路的类型、价格和长度
	静态管理成本	控制器与交换机间距离, 控制器间距离, 交换机的流路径请求率
	动态管理成本	控制器与交换机间距离, 控制器间距离, 交换机的流路径请求率
	能量消耗	控制器数量 链路数量

目标均为最优的控制器放置方案, 决策者需要对这些目标进行权衡。以下是 5 个典型的竞争关系和三类权衡方式。

(1) 优化目标的竞争关系

最小化无故障时控制器与交换机间时延, 最小化控制器故障时控制器与交换机间时延: 为了最小化无故障时控制器与交换机间时延, 控制器需要分散地放置在网络中。为了最小化控制器故障时控制器与交换机间时延, 所有控制器都趋向于放置在网络中心的位置, 当大部分控制器发生故障后, 无故障控制器可以接管相应的交换机并使控制器与交换机间时延尽可能小。

最小化控制器与交换机间时延, 最小化控制器间时延: 将控制器分散地放置在网络中会降低控制器与交换机间时延, 但是会导致控制器间时延增加。为了最小化控制器间时延而将控制器聚集在一起的

放置会导致控制器与交换机间时延的增加。

最小化控制器与交换机间时延, 最大化控制器负载均衡: 由于网络具有不规则的拓扑, 将交换机连接到距离最近的控制器以最小化控制器与交换机间时延的方案难以保证控制器的负载均衡。为了最大化控制器的负载均衡, 可能需要将交换机连接到距离较远的控制器, 从而导致控制器与交换机间时延增加。

最大化可靠性, 最小化时延: 为了提高可靠性, 控制器放置时主要考虑网络组件的故障概率和控制路径的故障概率, 可能会导致控制器与交换机以及控制器间的时延增加。此外, 为了进一步提高可靠性, 在故障发生时利用故障转移机制将交换机通过当前可用的路径连接到可用的控制器, 可能会进一步增加控制器与交换机间的时延。

最小化成本与能耗, 最小化时延(或最大化可靠

性): 为了最小化成本与能耗, 需要尽量减少放置控制器的数量以及控制路径中使用的链路数量。但是为了降低时延或提高可靠性, 趋向于在网络中部署大量的控制器并使用大量的控制路径来实现性能或可靠性的最优化。

(2) 优化目标的权衡方式

权衡方式 1: 设置约束。通常侧重于对其中一个目标进行优化, 而对其他指标设置约束, 例如设置最大时延约束、控制器可以管理的交换机个数、控制器间负载最大差值约束或故障容忍率等。这种权衡方式的缺点是需要大量的网络知识来提前设置约束值, 约束值的设置不当也会影响其他目标的优化结果。

权衡方式 2: 设置权重。根据优化目标的重要程度设置对应的权重, 将多目标优化问题转化为带权重的单目标优化问题。与权衡方式 1 类似, 权重的设置需要大量的先验知识, 并对结果产生重要影响。

权衡方式 3: Pareto 最优。计算所有 Pareto 最优结果, 构成 Pareto 边界, 之后, 网络运营商可以根据需求选择合适的方案。通常, 称放置方案 P_1 是 Pareto

最优的, 当且仅当不存在放置方案 P_2 使得对于所有的目标函数 $f_i (i \in \{1, \dots, J\})$ 有 $f_i(P_2)$ 不亚于 $f_i(P_1)$ 且至少有一个 i 使得 $f_i(P_2)$ 优于 $f_i(P_1)$ 。这种权衡方式的优点包括: 避免先验地对一些指标设置约束, 这些约束可能导致没有可行的放置方案; 能清晰地展示优化目标之间的权衡; 决策者可根据不同需求选择适合的方案。但是该方式由于需要返回 Pareto 边界, 相比权衡方式 1 和 2 更加耗时。

在已有的控制器放置问题的解决方案中, 若采用权衡方式 1, 则可根据优化目标的侧重将这些放置方案分为以性能为主要优化目标、以可靠性为主要优化目标和以成本与能耗为主要优化目标的三类放置方案。其中, 性能优化主要是指最小化时延, 且根据是否考虑控制器能力约束可进一步分为最小化无能力约束的时延和最小化有能力约束的时延这两类方案。若放置方案使用权衡方式 2 或 3, 则这些方案统称为多目标优化的控制器放置方案, 根据使用的权衡方式, 可进一步分为带权重和 Pareto 最优两类多目标放置方案。图 3 展示了控制器放置方案的分类, 下面对各类方案的已有研究进行详细介绍。

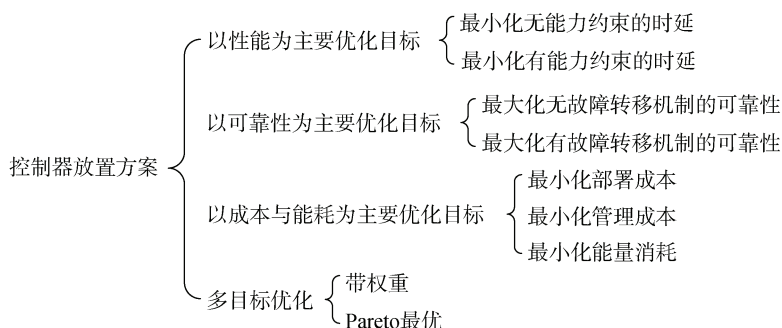


图 3 SDN 控制器放置方案分类

Figure 3 Classification of SDN controller placement schemes

3.1 性能优化

首先, 介绍以性能为主要优化目标的控制器放置方案, 包括最小化无能力约束的时延和最小化有能力约束的时延两类, 从时延、控制器负载、可靠性、成本与能耗以及算法等方面对已有方案进行分析, 如表 2 和 3 所示。

3.1.1 最小化无能力约束的时延

Heller 等人^[22]首次提出控制器放置问题, 分别以最小化控制器与交换机间平均时延和最小化控制器与交换机间最大时延为优化目标, 研究控制器的放置对控制器与交换机间时延的影响。最小化平均时延对应的优化问题是最小 k -median 问题^[48], 最小化最大时延对应的优化问题是最小 k -center 问题。他

们采用穷举法搜寻最优的控制器放置方案, 研究结果表明最小化平均时延的放置方案倾向于将控制器放置在交换机密度较高的位置, 而最小化最大时延的控制器放置倾向于地理分布较为中心的位置。

Wang 等人^[28]提出使用网络划分的方法解决控制器放置问题, 目标是最小化控制器与交换机间最大时延。他们基于标准的 k -means 聚类算法和 Katsavounidis 等人^[54]提出的初始化方法, 提出一个优化的 k -means 算法, 将网络划分为 k 个子网, 并使每个子网内的控制器与其管理的交换机间的最大时延最小。仿真结果表明, 相比标准的 k -means 算法, 所提的优化算法可以显著减小控制器与交换机间的最大时延。

表 2 最小化无能力约束的时延的控制器放置方案

Table 2 Controller placement schemes of minimizing the latency without capacity constraint

作者	时延	控制器负载	可靠性	成本与能耗	算法
Heller 等人 ^[22]	最小化控制器与交换机间平均时延	-	-	-	穷举法(最优解)
	最小化控制器与交换机间最大时延	-	-	-	穷举法(最优解)
Wang 等人 ^[28]	最小化控制器与交换机间最大时延	-	-	-	聚类算法
Wang 等人 ^[49]	最小化控制器与交换机间最大传播时延和控制器排队时延, 控制器与交换机间最大时延约束	-	-	-	聚类算法
Guo 等人 ^[30]	最小化链路故障时控制器与交换机间最大时延	-	可靠性约束	-	贪心算法(误差不大于 17.6%)
Alshamrani 等人 ^[50]	最小化控制器故障时控制器与交换机间平均时延	-	可靠性约束	-	穷举法(最优解)
	最小化控制器故障时控制器与交换机间最大时延	-	可靠性约束	-	穷举法(最优解)

表 3 最小化有能力约束的时延的控制器放置方案

Table 3 Controller placement schemes of minimizing the latency with capacity constraint

作者	时延	控制器负载	可靠性	成本与能耗	算法
Yao 等人 ^[29]	最小化控制器与交换机间最大时延	控制器能力约束	-	-	整数规划(最优解)
Hu 等人 ^[36]	最小化控制器与交换机间平均时延	考虑负载均衡的控制器负载约束	-	-	整数规划(最优解)
	最小化控制器与交换机间最大时延	考虑负载均衡的控制器负载约束	-	-	整数规划(最优解)
Killi 等人 ^[51]	最小化交换机与距离最近的第 N 个控制器间最大时延	控制器能力约束	可靠性约束	-	整数规划(最优解)
Killi 等人 ^[52]	最小化交换机与主控制器和主控制器与备用控制器间最大时延	控制器能力约束	可靠性约束	-	模拟退火算法(近优解)
Killi 等人 ^[53]	最小化链路故障时控制器与交换机间最大时延	控制器能力约束	可靠性约束	-	整数规划(最优解)

Wang 等人^[49]在之前研究的基础上, 提出最小化控制器与交换机间最大传播时延和控制器排队时延的控制器放置方案。分布式控制平面的模型基于 ONOS 控制平台^[15], 即交换机通过调度器与多个控制器连接, 调度器用于将交换机负载分布到不同的控制器, 交换机的流路径请求在到达调度器之前需要暂存在调度器前的队列中, 暂存的时间被称为控制器排队时延, 与控制器的处理能力和负载有关。在 WAN 中, 相比控制器排队时延, 传播时延在总时延中占主导地位, 因此需要分别处理这两种时延: 首先, 使用优化的 k-means 算法^[28]进行网络划分以减小每个子网内控制器到交换机间的最大传播时延; 之后, 在具有大量交换机的子网内放置多个控制器以减小控制器排队时延, 最终使控制器与交换机间时延满足最大时延约束。研究结果表明所提方案相比传统的 k-means 和 k-center 算法可以更有效地减小控制器与交换机间的时延。

Guo 等人^[30]提出两个控制器放置模型, 即考虑所有链路故障场景的控制器放置以及只考虑单链路故障的控制器放置。在链路故障发生后将受影响的

交换机通过当前可用路径连接到控制器, 最小化交换机与控制器间的最大传播时延。他们提出一个贪心算法对单链故障的控制器放置进行求解, 该算法每次迭代选择一个控制器放置位置, 使得当前的放置方案产生最小的链路故障时控制器与交换机间最大传播时延。在 Internet2 OS3E 拓扑^[55]和 Cernet2 拓扑^[56]上的仿真结果显示所提算法与最优解之间的误差不尽相同, 且最大误差达到 17.6%, 但随着放置控制器数量的增加, 所提算法的误差在两个拓扑上均呈现减小的趋势。

Alshamrani 等人^[50]在控制器放置时为每个交换机分配 3 个距离最近的控制器以应对最多 2 个控制器故障的场景。目标是最小化控制器故障时控制器与交换机间最大时延和平均时延。实验结果表明, 当控制器数量较少时, 为每个交换机分配多个控制器的方式会导致控制器与交换机间时延显著增加, 随着放置控制器数量的增加, 增加的时延逐渐减小, 可以在可靠性和性能之间取得平衡。

Heller 等人^[22]和 Alshamrani 等人^[50]都同时研究了不同性能优化目标的控制器放置: 由于网络拓扑

的不规则性, 最小化平均时延和最小化最大时延的控制器放置结果通常是不同的。Guo 等人^[30]和 Alshamrani 等人^[50]分别考虑了链路故障和控制器故障场景: 相比于不考虑故障场景的控制器放置, 考虑故障场景的控制器放置在提高可靠性的同时会明显增加控制器与交换机间的时延; 随着控制器数量的增加, 增加的时延会不断减小, 因此为了满足可靠性约束, 需放置尽可能多的控制器以减小控制器与交换机间时延的增加。

3.1.2 最小化有能力约束的时延

考虑控制器能力约束时, 最小化控制器与交换机间最大时延对应的优化问题是能力约束的 k -center 问题^[57]。Yao 等人^[29]将该问题描述为整数规划模型, 在二分查找法中通过解决整数规划的线性松弛确定控制器与交换机间距离的下界, 之后使用整数规划的方法进行求解, 并通过遍历 k 值寻找满足控制器能力约束的最小控制器数。在仿真中使用随机(交换机的流请求率服从均匀分布)和热点(20%的交换机为热点交换机)等两种不同的流模型来模拟不同的负载场景。研究表明, 使控制器不发生过载所需的最少控制器数在不同的负载场景中略有不同; 相比不考虑控制器能力约束的放置, 考虑控制器能力约束的放置可以使用较少的控制器保证控制器不发生过载, 并且产生较小的控制器与交换机间时延。

Hu 等人^[36]在考虑控制器负载均衡时分别最小化控制器与交换机间平均时延和最小化控制器与交换机间最大时延。他们使用考虑负载均衡的控制器负载约束调整控制器间的负载均衡情况, 并评估考虑负载均衡时所付出的时延代价, 在 SDN-lib^[58]中的拓扑上进行大量仿真。研究表明, 对于只考虑最小化时延的控制器放置方案, 只需增加较小的时延即可实现控制器之间良好的负载均衡, 特别地, 在 Abilene 拓扑和 Cost266 拓扑上分别使用 5 个和 2 个控制器进行放置时, 可以在不增加时延的情况下实现控制器间的负载均衡。

Killi 等人^[51]在控制器放置时为每个交换机分配距离最近的 N 个控制器。当最多 $N-1$ 个控制器发生故障时, 故障控制器管理的交换机可以重新分配至距其最近的一个备用控制器, 避免交换机与控制平面发生断连。优化目标是最小化交换机与距离最近的第 N 个控制器间的最大时延, 同时使控制器的负载满足其能力约束。研究结果显示, 控制器发生故障后, 将受影响的交换机连接到当前可用的控制器虽然可以保证交换机到控制平面的持续连接, 但是可能会导致交换机与控制平面间的时延过大; 在控制

器放置时提前考虑可能的故障场景并为交换机分配备用控制器, 可以避免故障发生之后交换机与控制平面间时延的急剧增长。

Killi 等人^[52]研究单控制器故障场景的控制器放置, 为每个交换机分配一个距离最近的控制器作为该交换机的主控制器, 选择距离该主控制器最近的控制器作为该交换机的一个备用控制器。优化目标是最小化交换机与主控制器以及主控制器与备用控制器间的最大时延, 并使主控制器与备用控制器均满足能力约束。他们将该问题建模为混合整数线性规划问题, 使用模拟退火算^[59]法进行求解。模拟退火算法是基于蒙特卡洛方法的启发式算法, 将当前解以一定的接受概率移动到更差解, 从而避免陷入局部最优, 接受概率在算法迭代初期时设置较大以允许更广泛的搜索空间, 随着迭代次数的增加, 接受概率逐渐减小, 有助于算法的收敛。通过经验以及大量的仿真实验为模拟退火算法针对特定的问题模型设置合适的相关参数。研究表明, 由于同时考虑了无故障和控制器故障场景, 所提方案会略微增加交换机与主控制器间时延, 但会显著降低交换机与备用控制器间时延; 与通过 CPLEX 优化器^[60]获得的最优解相比, 模拟退火算法可以使用少于一半的时间获取近优解。

Killi 等人^[53]在控制器放置时还考虑单链路故障场景, 将受影响的交换机通过备用路径连接到控制器, 目标是最小化链路故障时控制器与交换机间最大时延, 并使控制器的负载满足控制器能力约束。优化目标中单链路故障场景的最大时延和无故障场景的最大时延的权重由网络运营商决定。仿真结果表明: 无故障场景的最大时延和故障场景的最大时延具有竞争关系; 在控制器放置时为交换机设置备用路径的方式可以降低链路故障发生后控制器与交换机间的最大时延, 同时也降低了控制器间平均时延和最大时延。

由于控制器的处理能力有限, 保证控制器负载不超过其能力的控制器放置更加符合实际场景。Killi 等人^[51-53]在控制器放置时还分别考虑了控制器故障场景和链路故障场景。提前设计备用控制器和备用路径的方案会显著减小网络组件故障带来的控制器与交换机间时延急剧增大, 但也需要使无故障场景中的时延不受过多影响。通常使用网络场景的发生概率作为同时优化不同网络场景时延的权重。

3.2 可靠性优化

按照是否采用故障转移机制, 以可靠性为主要优化目标的控制器放置方案分为最大化无故障转移机制的可靠性和最大化有故障转移机制的可

靠性两类。

3.2.1 最大化无故障转移机制的可靠性

从故障场景、可靠性、时延、控制器负载、成

本与能耗以及算法等方面对最大化无故障转移机制的可靠性的已有控制器放置方案进行分析, 如表 4 所示。

表 4 最大化无故障转移机制的可靠性的控制器放置方案

Table 4 Controller placement schemes of maximizing the reliability without failover mechanism

作者	故障场景	可靠性	时延	控制器负载	成本与能耗	算法
Zhang 等人 ^[41]	交换机或链路故障导致控制器与交换机间控制路径故障	最小化控制器与交换机间控制路径平均故障概率	交换机与控制器通过最短路径连接	-	-	贪心算法, 基于最小割的启发式算法
Hu 等人 ^[39,40]	单个交换机或单条链路故障导致控制器与交换机或控制器间控制路径故障	最小化故障控制路径预期比例	交换机与控制器以及控制器之间通过最短路径连接	-	-	贪心算法, 模拟退火算法(近优解)
Guo 等人 ^[61]	交换机、链路或控制器故障导致控制器与交换机间控制路径故障	最大化故障时控制器连接的交换机数	交换机与控制器通过最短路径连接	-	-	基于贪心的社区发现算法
Jiménez 等人 ^[26]	交换机或链路故障导致控制器与交换机间控制路径故障	最小化控制器与交换机间故障控制路径数	控制器与交换机时延约束, 控制器间时延约束, 交换机与控制器以及控制器之间通过最短路径相连	-	-	聚类算法

Zhang 等人^[41]研究分离架构的可靠性, 考虑交换机或链路故障导致控制器与交换机间控制路径故障的场景, 目标是在控制器放置时最小化控制器与交换机间控制路径平均故障概率。他们首先使用基于最小割的方法将网络划分为 k 个子网, 目标是最小化每个子网内交换机到图心的距离, 同时最小化子网之间边的数目, 以提高子网内交换机间的路径多样性。之后, 将控制器放置在每个子网的中心交换机位置, 并将子网内的交换机通过最短路径与该控制器相连。他们在环型、星型和胖树等不同类型和规模的网络拓扑上进行大量的仿真, 以寻找可靠性指标与网络拓扑之间的关系, 并提出一个贪心放置方案用于算法的性能对比。仿真结果表明: 基于最小割的控制器放置方案相比随机放置和贪心放置方案具有更低的路径故障概率; 控制器与交换机间控制路径的长度越长, 控制器与交换机间连接丢失的概率越高。

Hu 等人^[39-40]采用基于原因的可靠性分析模型分析控制网络的可靠性, 考虑单个交换机或单条链路故障导致的控制器与交换机或控制器间控制路径故障的场景, 目标是最小化故障控制路径的预期比例。他们提出两个启发式算法: (1)贪心算法: 基于交换机故障率选出控制器放置的候选位置, 每次从候选位置中选择一个可以产生最小故障控制路径预期比例的位置放置一个控制器, 设置回溯步长进行回溯来寻找比当前放置更优的放置位置以避免陷入局部最优; (2)模拟退火算法: 是一个通用算法, 通过大量仿

真确定该算法对可靠控制器放置问题的优化配置参数。仿真结果证实了以下结论: (1)模拟退火算法的结果优于贪心算法; (2)若放置的控制器数量过少, 较多的交换机将通过相同的链路与控制器的连接, 单个网络组件(交换机或链路)故障会导致较多的控制路径故障; 若放置的控制器数量过多, 会产生大量的控制器间的控制路径, 组件故障也很有可能会导致多条控制路径故障; (3)最大化控制网络可靠性的控制器放置具有较高的控制器与交换机间时延, 但满足 Heller 等人^[22]提出的时延上限。

Guo 等人^[61]按照网络功能将 SDN 分为两个相互依赖的网络: 数据转发网络(交换机与交换机之间构成的网络)和控制网络, 并使用级联故障分析的方法分析控制器放置对控制网络可靠性的影响。研究链路或节点(控制器或交换机)故障导致控制器与交换机之间断连的情况, 目标是在网络组件故障时使控制器连接尽可能多的交换机。他们使用基于贪心的社区发现算法^[62]将网络划分为 k 个子网, 在每个子网内选择到其他节点具有最大接近中心性的节点放置控制器。实验结果发现网络的可靠性与网络拓扑的平均路径长度成反比, 因此, 环型网络拓扑具有较低的可靠性, 而随机网络由于其平均路径长度较短而具有较高的可靠性。

Jiménez 等人^[26]将控制器与交换机间控制路径构成的网络视为一组以控制器为根的树, 在控制器放置时考虑如何构建一组健壮树, 使所需的控制器数量最小, 并最小化交换机或链路故障导致的控制

器与交换机间故障控制路径的数量,同时满足控制器与交换机间时延约束和控制器间时延约束。他们提出名为 **k-Critical** 的聚类算法,将具有较大节点度且周围一定跳数范围内的节点数较多的节点作为控制器的候选节点,构建以控制器候选节点为根的树,最小化该树的深度以降低网络故障对控制路径造成的影响,同时使控制器与交换机间时延满足最大时延约束。之后,对于与其他控制器时延超过时延约束的控制器,在其管理域内添加一个控制器使控制器间时延满足约束。研究表明:使用多个控制器进行放置可以减小控制器与交换机间时延,但是过多的控制器是不划算的,因为时延的减少是微不足道的;**k-Critical** 算法以增加控制器与交换机间的时延为代价提高了网络的可靠性,由于时延满足最大时延约束,因此该代价可以接受;相比同规模的稀疏

型网络,密集型网络的路径长度较短,因此具有更小的控制器与交换机间时延。

Zhang 等人^[41]和 Jiménez 等人^[26]在控制器放置时考虑交换机或链路故障对控制器与交换机间控制路径的影响。Hu 等人^[39-40]研究了交换机或链路故障对整个控制网络的影响。Guo 等人^[61]还考虑了控制器故障场景。控制路径的长度越短,控制路径的故障概率越低。最大化可靠性的控制器放置方案通常具有较高的控制器与交换机间时延,可以通过设置最大时延约束进行限制。

3.2.2 最大化有故障转移机制的可靠性

从故障场景、故障转移机制、可靠性、时延、控制器负载、成本与能耗以及算法等方面对最大化有故障转移机制的可靠性的控制器放置方案进行分析,如表 5 所示。

表 5 最大化有故障转移机制的可靠性的控制器放置方案
Table 5 Controller placement schemes of maximizing the reliability with failover mechanism

作者	故障场景	故障转移机制	可靠性	时延	控制器负载	成本与能耗	算法
Hock 等人 ^[27]	最多两个交换机或两条链路故障导致出现无控制器节点	备用路径和备用控制器	最小化最大无控制器节点数	交换机与控制器通过最短路径连接	-	-	穷举法(最优解)
Hu 等人 ^[43]	单个交换机或单条链路故障导致控制器与交换机或控制器间控制路径故障	不限定具体的故障转移机制	最小化故障控制路径预期比例	交换机与控制器以及控制器之间通过最短路径连接	-	-	贪心算法,模拟退火算法(平均误差为 0.02%)
Müller 等人 ^[35]	交换机、链路或控制器故障导致控制器与交换机间控制路径故障	备用路径和备用控制器	最大化交换机与控制器间连通性	-	考虑备用的控制器能力约束	-	启发式算法
Vizarreta 等人 ^[42]	交换机、链路或控制器故障导致控制器与交换机间控制路径故障	备用路径和备用控制器	最小化控制器与交换机间主控制路径和备用控制路径的平均长度	最小化控制器与交换机间主控制路径和备用控制路径的平均长度	-	-	混合整数线性规划(最优解)

Hock 等人^[27]考虑最多两个交换机或两条链路同时发生故障的场景,给定控制器的数量,通过穷举法寻找最小化最大无控制器节点数的放置方案。无控制器节点数通常会随着控制器数量的增加而减少,因此还需要寻找不出现无控制器节点的最少控制器数的放置,搜寻方法基于以下假设:在两个相邻的无控制器节点位置至少放置一个控制器以保证在故障发生时这两个交换机节点不会成为无控制器节点。仿真结果表明:避免出现无控制器节点的控制器放置会导致控制器与交换机间时延增加;在 **Topology Zoo**^[56]的大多数拓扑中,超过 20%的交换机位置需要放置控制器从而在任何双交换机或链路故障时保证所有交换机与控制平面的持续连接。

Hu 等人^[43]研究单个交换机或单条链路故障导致

的控制器与交换机或控制器间控制路径故障的场景。不同的故障转移机制可能会导致不同的可靠性结果,使用条件故障概率表示由不同的网络组件故障和故障转移机制使控制路径发生故障的概率。他们证明了可靠的控制器放置问题是 **NP** 难问题,为了便于分析,假定所有控制路径的条件故障概率均相同,并使用之前研究^[39]提出的贪心算法和模拟退火算法进行求解。大量的仿真表明:模拟退火算法最接近最优解,平均误差为 0.02%;放置过多或过少的控制器都会降低控制网络的可靠性;可靠性和平均时延以及可靠性和最大时延之间通常不可能同时达到最优,但最大化可靠性的放置结果所增加的时延是可以接受的。

Müller 等人^[35]在控制器放置时考虑交换机与控

制器间路径的多样性(即连通性)以及备用控制器,提出一种弹性的控制器放置策略,具体分为两步: (1)为了最大化交换机与控制器间的连通性,选择使交换机与控制器之间节点不相交路径数最多的位置作为控制器的放置位置。同时使控制器的负载满足其能力约束,并为控制器预留部分能力以应对故障转移机制造成的负载增大的情况。使用整数线性规划方法寻找最优的放置结果。(2)为每个交换机选择一组备用控制器:可以使用基于邻近的方式选择距离交换机最近的一组控制器作为其备用控制器,或者使用基于剩余能力的方法选择剩余能力最多的一组控制器作为备用控制器。仿真结果表明:在控制器放置时考虑交换机与控制器间路径的多样性有助于降低交换机与控制平面之间断连的概率;故障发生后,基于剩余能力的备用控制器选择方法可以保证控制器不发生过载,而基于邻近的备用控制器选择方法虽然在第一步为每个控制器的部分能力进行预留,但还是会出现某些控制器过载的情况。

Vizarreta 等人^[42]提出两个控制器放置方案:方案一将交换机通过两条不相交的控制路径连接到同一个控制器;方案二将交换机通过两条不相交的控制路径分别连接到两个不同的控制器。考虑到链路故障概率与链路长度成正比,因此放置目标是最小化控制器与交换机间控制路径(包括主控制路径和备用控制路径)的平均长度。由于传播时延与路径的长度是线性关系,因此控制路径长度的优化也等同于控制器与交换机间时延的优化。他们将两个放置方案建模为混合整数线性规划问题,使用 Gurobi 求解

器^[63]进行求解。仿真结果表明,相比无故障转移机制且最小化控制器与交换机间平均控制路径长度的方案,所提方案以增加较小的控制器与交换机间时延为代价,显著提高了控制路径在不同故障场景中的恢复性: (1)在单链路或双链路故障场景中,故障控制路径预期比例降低了三个数量级; (2)当节点(交换机或控制器)故障是主要故障时,控制路径平均可用性也有明显改善,由于方案二提供了对控制器故障的保护,因此具有更优的结果。

Hock 等人^[27]提出不出现无控制器节点的控制器放置方案,由于无控制器节点不能通过任何故障转移机制与控制平面进行通信,因此是网络故障导致的最严重的后果之一。Hu 等人^[43]在控制器放置时最小化故障控制路径预期比例,但没有指明采用的具体故障转移机制。Müller 等人^[35]在控制器放置时考虑控制器能力约束,并最大化交换机与控制器间的连通性,但没能确保对每条控制路径的保护。Vizarreta 等人^[42]对控制路径的保护进行了明确的规划,但提出的两个放置方案适用于不同类型的故障场景,并不能很好地同时应对所有故障场景。

3.3 成本与能耗优化

以成本与能耗为主要优化目标的控制器放置方案分为最小化部署成本、最小化管理成本和最小化能量消耗三类。从成本与能耗、时延、控制器负载、可靠性以及算法等方面对不同的放置方案进行分析。

3.3.1 最小化部署成本

最小化部署成本的控制器放置方案如表 6 所示。

表 6 最小化部署成本的控制器放置方案
Table 6 Controller placement schemes of minimizing the deployment cost

作者	成本与能耗	时延	控制器负载	可靠性	算法
Rath 等人 ^[64]	最小化控制器数量	控制器与交换机间时延约束	控制器利用率约束	-	非零和博弈
Ros 等人 ^[65,66]	最小化控制器数量和每个交换机连接的控制器数量	-	-	可靠性约束	启发式算法
Sallahi 等人 ^[32]	最小化静态部署成本	流路径设置时延约束,交换机与控制器以及控制器之间通过最短路径连接	控制器能力约束	-	线性规划(最优解)
Sallahi 等人 ^[34]	最小化网络扩张成本	流路径设置时延约束,交换机与控制器以及控制器之间通过最短路径连接	控制器能力约束	-	线性规划(最优解)

Rath 等人^[64]根据网络中控制器负载的实时变化动态地调整控制器的数量以及交换机与控制器之间的分配,使控制器与交换机间时延满足给定时延约束、控制器利用率处于给定范围,最终目标是最小化控制器数量,即最小化控制器成本。他们使用分布式解决方案,在每个控制器上执行非零和博弈来解决

该优化问题。每个控制器试图最大化其利用率并最小化控制器与交换机间时延,与相邻控制器进行交互以执行调整交换机分配、增加或删除控制器等操作,具体逻辑如下:当控制器利用率低于最低阈值时,将该控制器管理域内的所有交换机分配给其相邻控制器,然后将该控制器删除或关闭;当控制器

利用率高于最大阈值时, 将该控制器的部分交换机分配给其相邻控制器以降低该控制器的负载; 当控制器利用率高于最大阈值, 但所有相邻控制器均不能在满足时延和控制器利用率约束条件下接收该控制器的负载时, 需增加一个新的控制器来分担过多的负载。

Ros 等人^[65-66]将交换机连接到多个控制器以提高控制网络的可靠性, 目标是在满足可靠性约束的条件下最小化每个交换机连接的控制器数量以及最小化控制器的总数, 即最小化控制器的成本(包括控制器的部署成本和为交换机提供服务的成本)。可靠性约束是指交换机通过不相交路径与多个控制器进行连接时, 交换机连接到控制平面的概率不小于给定门限值。他们提出一个启发式算法进行求解: 根据网络中节点的邻居节点的度数对节点进行由大到小排名, 排名较高的节点作为控制器的候选节点, 其余节点通过不相交路径连接到排名较高的候选节点以减少控制器的数量, 不断增加控制器数量直至达到可靠性约束。研究结果发现: 所需的控制器数量与网络中度为 1 的节点的数量呈正相关; 在 Topology Zoo^[56]的 75% 的拓扑中, 使用 8 个控制器即可满足 5 个 9 的可靠性约束, 每个交换机也只需连接 2 个控制器。

Sallahi 等人^[32]旨在设计全新的 SDN 控制网络, 且控制路径使用带外的连接方式。他们不仅考虑控制器的数量、位置以及控制器与交换机之间的连接, 还考虑控制器的类型(具有不同的端口数、处理速度和价格)、控制器与交换机以及控制器间链路类型(具有不同的带宽和价格), 并限制了控制器的放置位置(但不限定于交换机位置)。最终目标是 minimized SDN 控制网络的部署成本, 包括控制器的安装成本、控制器与交换机间链路成本以及控制器间的链路成本, 同时满足控制器能力约束和流路径设置时延约束, 流路径设置时延包括传输时延、传播时延和控制器处理时延。此外还需要满足一些其他约束条件, 例如, 控制器所使用的链路数不能大于控制器的端口数, 控制器与交换机之间的链路要满足交换机的带宽需求。他们将控制器放置模型建模为线性规划模型, 并使用 CPLEX 优化器^[60]进行求解。仿真结果显示: 当控制器放置位置数量固定时, 成本随着交换机数量的增加呈线性增加; 当交换机数量固定时, 成本随着控制器放置位置数量的增加而略微下降, 因为有更多的可能选择成本更低的放置位置。

Sallahi 等人^[34]还提出网络扩张模型, 即在已有的 SDN 网络中, 随着用户或业务需求的增长, 需要

在数据平面新增一组交换机时, 如何重新设计控制网络(例如新增控制器或调整交换机与控制器间的分配), 在满足控制器与交换机间时延约束和控制器能力约束的条件下, 最小化网络扩张成本。网络扩张成本包括控制器的安装和移除成本、控制器与交换机间链路的安装或移除成本以及控制器之间链路的安装或移除成本。他们基于之前研究^[32]提出的数学模型, 加以修改以考虑网络扩张场景, 并采用 CPLEX^[60]求解该线性规划问题。实验结果显示, 向不同规模的初始网络新增相同数量的交换机时, 初始规模较小的网络需要新增更多的控制器并安装和移除更多的链路, 从而导致更高的网络扩张成本。

Ros 等人^[65-66]提出的控制器放置方案在满足可靠性约束的条件下最小化控制器成本, 会导致控制器与交换机间时延过长, 适用于时延不敏感的网络场景。Rath 等人^[64]和 Sallahi 等人^[32,34]在控制器放置时考虑控制器能力和时延约束, 并考虑了动态放置场景, 通过调整控制器的数量和交换机与控制器间的分配以实现成本最优。Sallahi 等人^[32,34]的方案适用于使用带外连接的控制网络的设计, 且由于计算复杂度的原因只适用于小规模网络场景。

3.3.2 最小化管理成本

最小化管理成本的控制器放置方案如表 7 所示。

Obadia 等人^[44]在控制器放置时最小化静态管理成本, 同时满足控制器与交换机间时延约束。控制器之间使用全连接的方式进行连接, 在控制网络中的生成树上进行数据共享。他们提出一个贪心算法进行求解, 大致逻辑如下: 首先在每个控制器的可能放置位置放置一个控制器, 交换机连接到最近的控制器; 每次迭代时尝试关闭其中一个控制器, 使交换机与控制器间时延满足约束; 计算当前管理成本, 若小于之前的管理成本, 则进行下一轮迭代, 直至成本最低。实验结果显示: 若统计信息收集成本在管理成本中占比较大, 则控制器与交换机间的距离对管理成本具有重要影响, 控制器需要靠近交换机放置, 导致所需控制器的数量增加; 若流路径设置成本占比较大, 控制器与交换机间的距离需减小以减小管理成本, 同样也会导致控制器数量的增加; 若控制器同步成本占比较大, 则控制器的数量需要减小以最小化管理成本; 所提算法与最优解之间的误差小于 3%。

Bari 等人^[33]提出动态控制器放置的管理框架, 根据网络中流路径请求率的变化动态地调整交换机到控制器的分配, 最终目标是在满足控制器能力约束和控制器与交换机间时延约束的条件下, 最小化

表 7 最小化管理成本的控制器放置方案

Table 7 Controller placement schemes of minimizing the management cost

作者	成本与能耗	时延	控制器负载	可靠性	算法
Obadia 等人 ^[44]	最小化静态管理成本, 控制器间使用最小跳数路径相连	控制器与交换机间时延约束, 交换机连接到距其时延最短的控制器	-	-	贪心算法 (误差小于 3%)
Bari 等人 ^[33]	最小化动态管理成本	控制器与交换机间时延约束, 交换机与控制器以及控制器之间通过最短路径连接	控制器能力约束	-	基于背包问题的贪心算法, 模拟退火算法 (近优解)
Yao 等人 ^[67]	最小化流路径设置请求成本	交换机与控制器通过最短路径连接	控制器能力约束	-	聚类算法
	最小化交换机到控制器的重分配成本	交换机与控制器通过最短路径连接	控制器能力约束	-	穷举法 (最优解)
Ksentini 等人 ^[68]	最小化控制器间通信成本, 控制器与交换机间通信成本约束	交换机与控制器以及控制器之间通过最短路径连接	控制器间负载最大差值约束	-	线性规划 (最优解)
	最小化控制器与交换机间通信成本, 控制器间通信成本约束	交换机与控制器以及控制器之间通过最短路径连接	控制器间负载最大差值约束	-	线性规划 (最优解)

动态管理成本。若某个控制器连接的交换机都被分配到其他控制器, 则该控制器置为闲置状态, 闲置状态的控制器只监听特定端口以等待交换机的连接。他们将动态控制器放置问题建模为整数线性规划问题, 并提出两个启发式算法进行求解: (1) 基于背包问题的贪心算法: 每次迭代时选择到所有未分配的交换机距离之和最短的控制器, 然后使用贪心背包算法将交换机分配给该控制器; (2) 模拟退火算法: 针对之前交换机到控制器的分配, 首先调整不满足当前流量变化的分配, 然后使用模拟退火算法进一步优化分配。实验结果显示: 网络中的流量变化时, 两个启发式算法均可保证流路径设置时延不会产生较大波动; 模拟退火算法相比基于背包问题的贪心算法可以产生更加平稳和更低的流路径设置时延以及更小的动态管理成本, 且使用的控制器数量也更少, 但却产生更长的收敛时间。

Yao 等人^[67]在控制器放置时考虑交换机的权重, 目标是最小化流路径设置请求成本。交换机的权重由交换机的流请求率表示, 流路径设置请求成本表示为交换机的权重和交换机与控制器之间最短距离的乘积。在控制器放置时, 首先使用聚类算法^[69]将网络划分为 k 个子网, k 由控制器的能力和总负载决定, 然后在每个子网内选择流路径设置请求成本最低的节点放置控制器。控制器放置后, 由于网络中流量的变化可能会导致控制器发生超载, 此时需要将过载控制器管理的部分交换机迁移到相邻控制器以缓解过载问题并实现控制器间的负载均衡。迁移时选择流请求率较大的边界交换机, 目标是在保证新分配的控制器不发生超载的情况下, 最小化迁移的交换机到新分配的控制器的流路径设置请求成本, 即最

小化交换机到控制器的重分配成本。

Ksentini 等人^[68]在控制器放置时考虑两个优化目标: (1) 最小化控制器与交换机间的通信成本, 该成本与控制器到交换机间占用链路的总带宽和交换机的流路径请求率有关; (2) 最小化控制器间的通信成本, 即最小化控制器间占用链路的总带宽。他们提出两个单目标的解决方案: 方案一对目标(2)进行优化, 对目标(1)设置控制器与交换机间通信成本的最大接受值进行约束; 方案二对目标(1)进行优化并设置对目标(2)的约束值。两个方案均设置控制器间负载最大差值约束来保证控制器间的负载均衡, 并使用线性规划模型进行求解。仿真结果显示: 控制器与交换机间通信成本约束值的增大会导致控制器与交换机间通信成本的增加, 并且还会导致控制器与交换机之间以及控制器之间总的通信成本增加; 同样, 控制器之间通信成本约束值的增大也会导致控制器间的通信成本以及总的通信成本的增加。

Obadia 等人^[44]提出的最小化静态管理成本的控制器放置方案适用于流量波动不大的网络场景。Bari 等人^[33]和 Yao 等人^[67]提出的动态控制器放置方案适用于网络中流量变化较大的场景。Bari 等人^[33]提出的两个放置算法中, 虽然模拟退火算法的结果更优, 但是运行时间也更长, 因此模拟退火算法适用于需要近似最优结果的场景, 而基于背包问题的启发式算法更适合需要快速得出解决方案的场景。Obadia 等人^[44]和 Ksentini 等人^[68]的研究结果表明总成本中不同成本的权重的选择以及约束值的设置对最终的放置结果会产生重要影响, 因此需要针对特定的场景来合理确定相关参数的设置。

3.3.3 最小化能量消耗

最小化能量消耗的控制器放置方案如表 8 所示。

表 8 最小化能量消耗的控制器放置方案

Table 8 Controller placement schemes of minimizing the energy consumption

作者	成本与能耗	时延	控制器负载	可靠性	算法
Huque 等人 ^[45]	最小化控制器的能耗	控制器与交换机间时延约束, 控制路径使用最短路径	控制器能力约束	-	聚类算法
Ruiz-Rivera 等人 ^[47]	最小化控制器与交换机间控制链路的能耗	控制器与交换机间时延约束	控制器能力约束	-	启发式算法 (误差为 20%)
Hu 等人 ^[46]	最小化控制器与交换机间控制链路的能耗	控制器与交换机间时延约束	控制器能力约束	-	启发式算法 (误差不超过 4%)

Huque 等人^[45]提出放置一组控制器模块来管理整个网络, 每个控制器模块包含一定数量的控制器。根据交换机流路径请求率的实时变化, 动态调整控制器模块中激活的控制器数目, 在保证控制器不发生过载的情况下最小化控制器模块中控制器的数量, 从而最小化控制器的能耗。在控制器模块放置时, 首先将网络划分为多个子网, 使每个子网内任意两个交换机间的时延满足控制器与交换机间时延约束。然后在每个子网内选择使控制器与交换机间最大时延最小的位置放置控制器模块, 该位置可以是地理平面内的任意位置。仿真结果显示, 在网络的非高峰时期, 所提方案可以通过关闭部分控制器来降低控制器能耗; 相比稀疏型网络, 在密集型网络中关闭的控制器比例更高。

Ruiz-Rivera 等人^[47]对已有的控制器放置方案进行优化。在网络的非高峰时期, 尽可能关闭交换机与控制器间的链路, 从而最小化控制链路的能耗。假设所有链路具有相同的能耗, 优化目标可简化为最小化交换机与控制器间使用的链路数。关闭链路前, 控制器与交换机间受影响的通信路径需要进行调整。为了避免调整后过长的通信时延, 他们设置控制器与交换机间最大时延约束进行限制, 并且还设置控制器能力约束避免出现过载, 设置链路最大利用率约束防止发生拥塞。他们将该问题建模为 0-1 整数规划问题, 提出一个名为 GreCo 的启发式算法进行快速求解。实验结果表明, 在网络非高峰时期通过关闭链路的方式可以节省高达 55% 的能耗, 所提启发式算法需要使用比最优解多 20% 的链路。

受 Ruiz-Rivera 等人^[47]的启发, Hu 等人^[46]从能耗的角度来考虑控制器的初始放置, 将具有能耗意识的控制器放置问题建模为 0-1 整数规划问题, 目标是最小化控制器与交换机间控制链路的能量消耗, 同时满足于控制器与交换机间时延约束和控制器能力约束。他们基于 GreCo 算法^[47]提出一个启发式算法进行求解。仿真结果表明, 为了节省链路的能耗, 需

要减少控制路径中使用的链路数量, 会导致控制器与交换机间时延增大, 但满足最大时延约束; 相比最优解, 所提启发式算法使用的额外链路数不会超过 4%。

Huque 等人^[45]提出控制器模块的概念, 并将控制器模块放置于地理平面的任意位置, 适用于地理不受限的放置场景。Ruiz-Rivera 等人^[47]和 Hu 等人^[46]旨在通过减少控制器与交换机间控制链路的数量从而减小链路的能耗, 但由于采用带内连接方式, 关闭控制链路的方法会影响到数据平面流的转发。

3.4 多目标优化

多目标优化的控制器放置方案分为带权重和 Pareto 最优两类, 从时延、控制器负载、可靠性、成本与能耗以及算法等方面对不同的放置方案进行分析。

3.4.1 带权重的多目标优化

带权重的多目标优化的控制器放置方案如表 9 所示。

Ksentini 等人^[68]在控制器放置时考虑两个优化目标: (1)最小化控制器与交换机之间的通信成本; (2)最小化控制器之间的通信成本。这两个优化目标之间是竞争关系, 使用 Nash 讨价还价博弈在两个优化目标之间进行权衡, 使总的通信成本最小, 同时使控制器之间满足负载最大差值约束, 建模为线性规划模型。仿真结果显示, 相比优化单个目标并对另一个目标设置约束的方案, 所提的多目标解决方案可以在两个优化目标之间实现更好的权衡, 并取得最低的总通信成本。

Tanha 等人^[70]在控制器放置时为了应对控制器故障为交换机分配多个备用控制器, 备用控制器的不同数量表示不同的弹性等级, 目标是在满足控制器能力约束下, 最小化控制器的部署成本以及最小化流路径设置请求成本。假设控制器的部署成本与控制器放置位置的属性相关, 放置位置的节点度越大, 部署成本越低, 所以在控制器放置时倾向于将

表 9 带权重的多目标优化的控制器放置方案

Table 9 Weighted multi-objective optimized controller placement schemes

作者	时延	控制器负载	可靠性	成本与能耗	算法
Ksentini 等人 ^[68]	交换机与控制器以及控制器之间通过最短路径连接	控制器间负载最大差值约束	-	最小化控制器与交换机间通信成本, 最小化控制器间通信成本	线性规划(最优解)
Tanha 等人 ^[70]	交换机到控制器通过最短路径连接	控制器能力约束	可靠性约束	最小化控制部署成本, 最小化流路径设置请求成本	线性规划(最优解)
Zhang 等人 ^[37]	最小化控制器与交换机间最大时延(包括控制器处理时延、交换机处理时延和控制器与交换机间传播时延)	最小化控制器负载率方差	最小化控制器与交换机间控制路径故障概率	-	自适应细菌觅食优化算法(近优解)

控制器放置在节点度大的位置, 使控制器与交换机间具有较高的连通性。流路径设置请求成本包括交换机到主控制器和所有备用控制器的流路径设置请求成本, 与交换机的流请求率和交换机与控制器间的时延有关。他们将以上两个成本的加权和作为最终的优化目标, 并使用线性规划方法进行求解, 根据交换机的流请求率是否相同以及控制器的能力是否一致, 考虑三种不同的网络场景。仿真结果显示, 所需的控制器数量不仅与弹性等级有关, 并且还依赖于网络拓扑和网络场景; 在大多数网络场景中, 交换机与控制器间的时延随着弹性等级的增加而增大; 随着弹性等级的增加, 控制器间的负载均衡情况在某些拓扑中会不断恶化, 而在其余拓扑中则会不断改善, 再次体现了放置结果对网络拓扑的依赖。

Zhang 等人^[37]在控制器放置时考虑三个优化目标: (1)最大化可靠性, 即最小化控制路径故障概率; (2)最大化控制器间负载均衡状况, 即最小化控制器负载率方差; (3)最小化控制器与交换机间最大时延, 包括控制器处理时延、交换机处理时延以及控制器与交换机间传播时延。在交换机到控制器的分配时考虑将交换机同时连接到多个控制器并将流路径请求分配到多个控制器。在建模时为多个目标分配权重从而将多目标优化问题转化为单目标优化问题, 使用自适应细菌觅食优化算法^[71]进行求解。仿真结果显示, 所提方案可以获得近优解, 在优化目标之间取得权衡, 在提高网络可靠性的同时, 实现控制器间的负载均衡并减少控制器与交换机间时延。

Ksentini 等人^[68]的研究结果表明设置权重的权衡方式相比只优化单个目标而对其他目标设置约束的方式可以取得更优的放置结果。Tanha 等人^[70]和 Zhang 等人^[37]提出的方案均适用于可靠的控制器放置, Zhang 等人^[37]的方案考虑了更多的优化目标并且算法更为高效, 适合大规模网络场景的放置。

3.4.2 Pareto 最优的多目标优化

Pareto 最优的多目标优化的控制器放置方案如表 10 所示。

Heller 等人^[22]不仅分别研究了最小化控制器与交换机间平均时延和最小化控制器与交换机间最大时延的控制器放置, 还研究了同时考虑平均时延和最大时延时的 Pareto 最优解, 分析了两个指标之间的关系。实验结果表明, 最小化平均时延和最小化最大时延这两个优化目标在现实网络拓扑中很难同时达到最优。

Hock 等人^[27]权衡最小化无故障时控制器与交换机间最大时延以及最小化控制器故障时控制器与交换机间最大时延, 通过穷举法找出所有 Pareto 最优放置, 由运营商选择满足其需求的放置方案。此外, 他们还考虑最小化控制器与交换机间最大时延和最小化控制器间负载最大差值, 使用 Topology Zoo^[56]的拓扑进行大量仿真, 展示这两个指标的 Pareto 最优放置, 结果表明: 在 20%的拓扑中, 存在这两个指标同时最优的放置, 而在 80%的拓扑中, 一个指标的最优会显著恶化另一个指标。之后, 他们又加入控制器故障时控制器与交换机间最大时延和控制器故障时控制器间负载最大差值两个指标, 展示了无故障时的 Pareto 最优放置对控制器故障时的时延和负载均衡都会产生负面影响。最后, 他们研究最小化控制器与交换机间最大时延和最小化控制器间最大时延的 Pareto 最优放置, 结果表明这两个优化目标不可能同时达到最优, 基于这两个指标对控制器放置位置特征的影响, 对两个指标的不同侧重将影响控制器放置位置在整个网络中的分散程度。

Lange 等人^[24]使用 Czyżak 等人^[72]提出的 Pareto 模拟退火算法来解决多目标优化的控制器放置问题。Pareto 模拟退火算法是一个通用、多目标的启发式算法, 可以将任意目标加入到评估中, 且不限制

表 10 Pareto 最优的多目标优化的控制器放置方案

Table 10 Pareto-optimal multi-objective optimized controller placement schemes

作者	时延	控制器负载	可靠性	成本与能耗	算法
Heller 等人 ^[22]	最小化控制器与交换机间平均时延, 最小化控制器与交换机间最大时延	-	-	-	穷举法 (最优解)
	最小化无故障时控制器与交换机间最大时延, 最小化控制器故障时控制器与交换机间最大时延	-	-	-	穷举法 (最优解)
Hock 等人 ^[27]	最小化控制器与交换机间最大时延	最小化控制器间负载最大差值	-	-	穷举法 (最优解)
	最小化无故障时控制器与交换机间最大时延, 最小化控制器故障时控制器与交换机间最大时延	最小化无故障时控制器间负载最大差值, 最小化控制器故障时控制器间负载最大差值	-	-	穷举法 (最优解)
	最小化控制器与交换机间最大时延, 最小化控制器间最大时延	-	-	-	穷举法 (最优解)
Lange 等人 ^[24]	最小化控制器与交换机间平均时延, 最小化控制器与交换机间最大时延, 最小化控制器间平均时延, 最小化控制器间最大时延	最小化控制器间负载最大差值	-	-	Pareto 模拟退火算法(平均误差为 2%)
Lange 等人 ^[25]	最小化控制器与交换机间平均时延	最小化控制器间负载最大差值	-	-	聚类算法 (近优解)
Jalili 等人 ^[74]	最小化控制器间最大时延	最小化控制器间负载最大差值	-	-	遗传算法 (近优解)
Ahmadi 等人 ^[75]	最小化控制器间最大时延	最小化控制器间负载最大差值	-	-	遗传算法 (近优解)
	最小化控制器与交换机间最大时延, 最小化控制器间最大时延	-	-	-	遗传算法 (近优解)
Ahmadi 等人 ^[76]	最小化控制器与交换机间最大时延	最小化控制器间负载最大差值	-	-	遗传算法 (近优解)
	最小化控制器间最大时延	最小化控制器间负载最大差值	-	-	遗传算法 (近优解)
	最小化控制器与交换机间平均时延, 最小化控制器与交换机间最大时延, 最小化控制器间平均时延	控制器能力约束	-	-	聚类算法

优化目标的数量。在仿真中, 对控制器与交换机间平均时延、控制器与交换机间最大时延、控制器间平均时延、控制器间最大时延以及控制器间负载最大差值等五个指标进行优化, 评估 Pareto 模拟退火算法的运行时间以及与最优解之间的误差。结果显示, 对于穷举搜索算法需要几十分钟才能得出结果的问题实例, Pareto 模拟退火算法在几十秒内即可得出结果, 且与最优解之间的平均误差为 2%。

Lange 等人^[25]还研究专用的算法对控制器放置问题的适用性, 考虑控制器与交换机间平均时延和控制器间负载最大差值两个优化指标。他们基于 k-medoids 聚类算法^[73], 提出 Pareto 能力约束的 k-medoids 算法, 通过设置控制器能力约束来限制每个控制器分配的交换机个数, 控制器能力约束隐式地影响了控制器间负载最大差值, 通过不断增加控制器能力约束值, 最优化控制器与交换机间平均时

延, 从而得出时延与控制器负载均衡在控制器放置时的不同权衡。仿真结果表明, 专用的启发式算法(即 Pareto 能力约束的 k-medoids 算法)的结果准确性优于通用方法(即 Pareto 模拟退火算法^[72]), 但当考虑更多的优化目标时(例如最小化控制器与交换机间最大时延、最小化控制器间平均时延和最小化控制器间最大时延), 由于通用算法可以搜索更多的空间, 因此可以获取更优解。

Jalili 等人^[74]使用 NSGA-II 算法^[78]来解决多目标的控制器放置问题, 目标包括最小化控制器间最大时延和最小化控制器间负载最大差值。NSGA-II 算法是一种有效的多目标遗传算法, 用于寻找 Pareto 最优解, 它采用快速非支配排序算法降低计算复杂度, 采用拥挤度和拥挤度比较算子保留种群多样性, 并使用精英机制提高优化结果。在 Internet2 OS3E 拓扑^[55]上的仿真结果验证了所提算法可获得近似

Pareto 边界的解。

Ahmadi 等人^[75]基于 NSGA-II 算法提出一种混合的 NSGA-II 算法来解决控制器放置问题, 最小化控制器间最大时延并最小化控制器间负载最大差值。通过应用两次三重锦标赛选择策略, 在混合交叉过程中获得两个父代种群, 在每次锦标赛选择中, 从种群中选择三个随机的方案, 并通过拥挤度比较算子返回最优的方案, 父代确定后, 通过基于路径重连策略^[79]的交叉产生子代种群, 这种修改允许生成更具多样性的子代种群。通过仿真发现, 所提算法在运行过程中不断提高结果准确性的同时, 展示了结果的多样性, 避免陷入局部最优或聚集在少数几个最优解当中, 最终取得至少一半的 Pareto 最优解。

Ahmadi 等人^[76]基于两个有效的遗传算法 NSGA-II^[78]和 NSGA-III^[80]提出一种称为多起点的混合非支配排序遗传算法来解决控制器放置问题。该算法使用贪心的启发式产生高质量的初始种群, 还加入了强化机制、本地搜索机制以及分散机制, 所有机制在多起点的步骤中进行协作, 构成一个有效的进化算法。实验评估了算法在控制器与交换机间最大时延、控制器间最大时延和控制器间负载最大差值等优化指标之间的权衡。结果显示所提算法可以使用较少的计算时间和内存资源生成靠近 Pareto 边界并具有良好的分布的近优解。

Liao 等人^[77]提出基于密度的控制器放置方案, 首先使用基于密度的聚类算法将网络划分为 k 个子网, 使每个子网内的交换机具有紧密的连接, 而子网之间的交换机具有相对较少的连接。之后, 根据不同的优化目标, 通过穷举法在每个子网内找出一个最优的放置或一组 Pareto 最优放置, 从而实现全网的近似最优放置。实验评估了所提方案在控制器与交换机间平均时延、控制器与交换机间最大时延和控制器间平均时延等优化指标的 Pareto 最优解, 并与 Zhang 等人^[41]提出的基于最小割的放置方案和 Lange 等人^[24]提出的 Pareto 模拟退火算法进行比较。实验结果显示: 相比 Pareto 模拟退火算法, 所提方案与基于最小割的方案均可以取得更小的控制器与交换机间时延, 且所提方案的结果优于基于最小割的方案; Pareto 模拟退火算法可以取得更小的控制器间时延, 这是因为所提方案与基于最小割的方案都是在每个子网内放置一个控制器, 控制器间的距离受到了子网划分的影响; 所提方案在链路故障时还可取得较高的可靠性, 所有放置方案在一个规模较小的网络拓扑中均产生了非常低的可靠性, 说明网络的拓扑结构对网络的可靠性也有重要影响。他们还

提出了考虑控制器能力约束的基于密度的聚类算法, 相比能力约束的 k -center 算法^[29], 该算法放置更少的控制器使每个控制器不发生过载。

Pareto 最优的多目标优化方案可以清楚地展示优化目标之间的权衡。穷举法^[22,27]遍历所有搜索空间可以获得最优解, 适用于小规模网络的放置。通用的多目标算法(例如 Pareto 模拟退火算法和 NSGA-II 遗传算法)可以有效地解决控制器放置问题, 但针对控制器放置问题的专用算法(例如聚类算法^[25,77])通常可以取得更优解。随着优化目标数量的增多, 通用算法具有更高的准确性。因此, 专用算法更适合解决优化目标数量较少的控制器放置问题, 而通用算法适用于优化目标数量较多的控制器放置。

3.5 小结

由于在地理分布广泛的 WAN 中搭建专用的控制链路是成本不划算的, 大多数方案使用带内连接的方式构建控制路径。分布式控制平面大多使用扁平模型, 且控制器之间使用全网状互连的方式进行连接。时延作为最重要的指标之一成为很多方案的优化指标或考虑因素, 且主要考虑传播时延。基本所有的控制器放置方案都使用最短路径作为控制路径以减小时延。由于控制器的处理能力有限, 使用控制器能力约束对控制器的负载进行限制。此外, 控制器的负载也会影响控制器的性能以及可靠性, 因此控制器间负载的分布情况也被加以考虑。随后, 可靠性、成本与能耗等多种指标加入到控制器放置问题的考虑因素, 控制器放置问题成为了多目标的优化问题。已有研究根据不同的优化目标和不同的问题实例规模, 使用不同的权衡方式以及不同的算法来解决控制器放置问题。设置约束的权衡方式适用于优化单个目标的放置场景, 其他指标约束值的设定需要相关的网络知识, 可以根据运营商的需求预先指定, 例如, 运营商可能希望部署的 SDN 控制网络能够在满足控制器与交换机间特定的时延约束下最大化网络的可靠性。当需要同时优化多个目标时, 由于同类指标可以较为合理地分配权重, 因此设置权重的权衡方式更加适用于优化多个同类指标的放置场景, 例如同时优化多种管理成本。Pareto 最优的权衡方式可以展示多个优化目标之间的权衡关系, 使运营商可以根据需求从中选择合适的放置方案, 更加适用于优化非同类指标的放置场景。

4 研究展望

已有研究工作主要从性能、可靠性、成本与能耗等方面解决 SDN 控制器放置问题。然而, SDN 控

制器放置仍然存在很多问题有待进一步研究, 简要介绍如下:

(1) 控制器数量的确定

控制器数量的不同会导致不同的控制器放置位置以及交换机与控制器之间的分配, 因此合适的控制器数量对控制器的放置非常重要。文献[22,24,27,41,49,61]提前设置控制器数量, 然后寻找该控制器数量下的最优放置位置和分配。然而, 根据经验设置的控制器数量并不一定产生最优的放置结果。文献[29]采用逐渐增加控制器数量的方式寻找满足控制器能力约束的最小控制器数量。文献[77]选择使时延收益递减的控制器数作为放置的控制器数。文献[28]基于控制器与交换机间时延约束将网络分成 k 个子网并在每个子网放置一个控制器。文献[26]通过控制器间时延约束确定控制器的数量。这些方案在确定控制器数量时只考虑单一的优化指标并且通常比较耗时。控制器的数量不仅决定了控制器的部署成本与能耗, 还对时延和可靠性等方面产生重要影响。因此, 控制器数量的确定需要综合考虑多个优化指标并寻找快速、准确的算法。

(2) 控制器负载对控制器性能和可靠性的影响

受处理器、内存和带宽等资源的限制, 控制器具有有限的处理能力, 只能管理有限数量的交换机。控制器的负载也会影响控制器的处理能力, 并最终影响控制器的处理时延。文献[81]提出了一个控制器处理时延与控制器能力和负载之间关系的模型, 由排队模型和路由计算的时间复杂度两部分组成。但是该模型没有考虑控制器的并行处理能力以及不同类型控制器的处理逻辑。此外, 控制器的负载还会影响控制器的可靠性, 可靠性会随着负载的升高而降低。目前还没有对控制器负载与其可靠性之间关系的定量研究。控制器的负载、性能和可靠性是控制器放置时需要考虑的重要指标, 它们之间的相互影响有待进一步研究。

(3) 控制器同步方案

控制器之间需要经常通信以同步网络状态, 已有文献提出多种状态同步方案, 例如, 使用网络信息库(NIB)的同步方案^[13]、基于定期同步的方案^[82]、基于负载方差的同步方案^[83]和基于 Paxos 的快速同步方法^[84]。不同方案具有不同的同步效率并产生不同的同步开销。因此, 在控制器放置时, 当考虑控制器间时延或管理成本等相关指标时, 需要指定控制器放置场景中使用的控制器同步方案, 以便得出准确的放置结果, 或者设计更优的同步方案以减小控制器间同步时延和管理成本。

(4) 控制路径的分配

控制器放置时, 交换机通常分配到距离最近的控制器, 并使用最短路径与控制器进行通信。这种分配方式可以最大程度减小控制器与交换机间的时延, 但是对某些指标可能会产生负面影响, 例如分配到距离最近的控制器可能会导致控制器间负载不均衡甚至出现控制器过载, 某些最短路径可能具有较大的故障概率或产生较高的成本。因此需要综合考虑多个目标恰当地进行控制器与交换机间的分配并选择合适的控制路径。此外, 很少有文献在控制器放置时考虑控制路径中链路的负载状况。过高的链路负载会导致链路传输性能下降和可靠性降低。在选择控制路径时需要考虑链路的利用率, 使控制流量均衡地分布在整個控制网络中。

(5) 交换机迁移的触发条件及迁移选择

由于网络中流量的变化, 需要动态地调整交换机与控制器之间的分配(即交换机迁移)以实现控制器之间的负载均衡, 进而提高控制平面的性能。交换机迁移通常发生在以下三个场景中: (1)当网络中的总流量过大甚至超过所有控制器的能力时, 需要加入新的控制器并将一些交换机迁移到该控制器以分担部分负载; (2)当网络中的总流量过低时, 需要将部分控制器关闭或置为睡眠状态并将这些控制器管理的交换机迁移到其他控制器, 从而减小控制器能耗和控制器间通信成本; (3)当网络中总流量没有超过控制器能力但却出现控制器负载不均衡时, 需要将负载过大的控制器管理的部分交换机迁移到其他控制器以实现控制器间负载均衡。在上述三个场景中, 负载的“过大”和“过低”目前还没有明确的定量表示, 通常是在控制器过载时才进行交换机的迁移。因此需要合理地考虑交换机迁移的触发条件。当执行迁移操作时, 也需要合理地选择迁移交换机和迁移目的控制器, 不仅要在控制器负载均衡、控制路径时延、控制路径可靠性和管理成本等方面进行权衡, 还需要考虑迁移操作的成本。

(6) 高效的算法

随着网络规模的扩大和网络状态的不断变化, 网络管理员需要实时调整控制器的数量、位置以及交换机到控制器的分配, 以实现网络高效、可靠和经济地运行。然而, 控制器放置问题通常是 NP 难问题, 寻找最优的放置方案是耗时的。此外, 控制器放置问题通常是多目标的联合优化问题, 且目标之间具有竞争关系, 增加了问题的复杂性。因此, 在建模时需要综合考虑多个优化目标并设计更高效的算法快速、准确地解决控制器放置问题。

表 11 不同类型网络的控制器放置问题研究

Table 11 Survey of controller placement problem for different types of networks

作者	网络类型	放置特点	放置目标	放置约束	算法
Wang 等人 ^[81]	数据中心网络	控制器处理时延是主要时延	最小化带权重的控制器响应时间和流路径设置请求成本	控制器能力约束	联合博弈(平均误差为 0.35%)
Abdel-Rahman 等人 ^[85]	无线网络	需要考虑网络接入时延, 与选择的信道模型有关	最小化控制器数量	平均响应时延约束, 最大响应时延约束	整数线性规划(最优解)
Qin 等人 ^[86]	边缘网络	无线连接, 某些节点由于能力有限不能放置控制器	最小化带权重的控制器与节点之间的管理成本和控制器同步成本	控制器能力约束	贪心算法(近优解)
Papa 等人 ^[87]	近地轨道网络	地理位置、无线链路和流量不断变化	最小化平均流路径设置时延	-	整数线性规划(最优解)

(7) 不同类型网络的控制器放置

大多数已有研究关注的都是 WAN 中的控制器放置问题。近年来, 随着 SDN 应用场景的扩展和多控制器需求的增加, 研究人员对数据中心网络和无线网络中的控制器放置问题也展开了研究, 代表性成果如表 11 所示。在数据中心网络中, 控制器放置于接入层交换机连接的服务器上, 可采用带外的连接方式, 主要考虑控制器处理时延。在无线网中, 控制器与被控制元素之间通常通过无线连接进行通信, 需要指定使用的无线信道模型以确定通信方式并得出网络接入时延。不同类型的网络具有不同的网络特征、特定的优化目标和相应的指标计算公式, 在控制器放置时需要考虑这些差异以构建合适的模型, 并使用合适的算法进行求解。

5 总结

控制器的放置对使用分布式控制平面的 SDN 的实际部署至关重要, 有效的放置方案会降低控制器与交换机以及控制器间的通信时延、提高控制网络的可靠性并降低成本与能耗。本文首先详细介绍了控制器放置问题的优化指标, 包括时延、控制器负载、可靠性和成本与能耗, 然后根据优化目标和优化目标之间的权衡方式将已有研究提出的解决方案分为以性能为主要优化目标、以可靠性为主要优化目标、以成本与能耗为主要优化目标以及多目标优化等四类控制器放置方案, 并对这些方案进行详细地对比与分析。最后, 总结了分布式 SDN 控制器放置未来需要解决的问题。

参考文献

- [1] Kim H, Benson T, Akella A, et al. The evolution of network configuration: a tale of two campuses[C]. *The 2011 ACM SIGCOMM conference on Internet measurement conference*, 2011: 499-514.
- [2] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow[J]. *ACM SIGCOMM Computer Communication Review*, 2008, 38(2): 69-74.
- [3] Gude N, Koponen T, Pettit J, et al. Nox[J]. *ACM SIGCOMM Computer Communication Review*, 2008, 38(3): 105-110.
- [4] Floodlight Project[EB/OL]. <http://www.projectfloodlight.org/>.
- [5] Ryu SDN Framework[EB/OL]. <https://osrg.github.io/ryu/>.
- [6] Erickson D. The Beacon Openflow Controller[C]. *The second ACM SIGCOMM workshop on Hot topics in software defined networking - HotSDN '13*, 2013: 13-18.
- [7] Tootoonchian A, Gorbunov S, Ganjali Y, et al. On Controller Performance in Software-defined Networks[C]. *Hot-ICE'12: The 2nd USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*. 2012: 10.
- [8] Michel O, Keller E. SDN in wide-area networks: A survey[C]. *2017 Fourth International Conference on Software Defined Systems (SDS)*, 2017: 37-42.
- [9] Yeganeh S H, Tootoonchian A, Ganjali Y. On Scalability of Software-defined Networking[J]. *IEEE Communications Magazine*, 2013, 51(2): 136-141.
- [10] Kandula S, Sengupta S, Greenberg A, et al. The Nature of Data Center Traffic: Measurements & Analysis[C]. *The 9th ACM SIGCOMM conference on Internet measurement conference - IMC '09*, 2009: 202-208.
- [11] Curtis A R, Mogul J C, Tourrilhes J, et al. DevoFlow: scaling flow management for high-performance networks[C]. *ACM SIGCOMM Computer Communication Review*, 2011: 254-265.
- [12] Yu M L, Rexford J, Freedman M J, et al. Scalable Flow-based Networking with DIFANE[J]. *ACM SIGCOMM Computer Communication Review*, 2010, 40(4): 351-362.
- [13] Koponen T, Casado M, Gude N, et al. Onix: A Distributed Control Platform for Large-scale Production Networks[C]. *OSDI'10: The 9th USENIX Conference on Operating Systems Design and Implementation*. 2010: 351-364.
- [14] Tootoonchian A, Ganjali Y. HyperFlow: A Distributed Control Plane for OpenFlow[C]. *INM/WREN'10: The 2010 Internet Network Management Conference on Research on Enterprise Net-*

- working. 2010: 3.
- [15] Berde P, Gerola M, Hart J, et al. ONOS: Towards an Open, Distributed SDN OS[C]. *The third workshop on Hot topics in software defined networking*, 2014: 1-6.
 - [16] Hassas Yeganeh S, Ganjali Y. Kandoo: A Framework for Efficient and Scalable Offloading of Control Applications[C]. *The first workshop on Hot topics in software defined networks - HotSDN '12*, 2012: 19-24.
 - [17] Jain S, Kumar A, Mandal S, et al. B4: experience with a globally-deployed software defined wan[J]. *ACM SIGCOMM Computer Communication Review*, 2013, 43(4): 3-14.
 - [18] Yap K K, Motiwala M, Rahe J, et al. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering[C]. *The Conference of the ACM Special Interest Group on Data Communication*, 2017: 432-445.
 - [19] Oktian Y E, Lee S, Lee H, et al. Distributed SDN Controller System: A Survey on Design Choice[J]. *Computer Networks*, 2017, 121: 100-111.
 - [20] Karakus M, Durrresi A. A Survey: Control Plane Scalability Issues and Approaches in Software-Defined Networking[J]. *Computer Networks*, 2017, 112: 279-293.
 - [21] Stribling J, Sovran Y, Zhang I, et al. Flexible, wide-area storage for distributed systems with WheelFS[C]. *The 6th USENIX symposium on Networked systems design and implementation*, 2009: 43-58.
 - [22] Heller B, Sherwood R, McKeown N. The controller placement problem[C]. *The first workshop on Hot topics in software defined networks*, 2012: 7-12.
 - [23] Owen S H, Daskin M S. Strategic Facility Location: A Review[J]. *European Journal of Operational Research*, 1998, 111(3): 423-447.
 - [24] Lange S, Gebert S, Zinner T, et al. Heuristic Approaches to the Controller Placement Problem in Large Scale SDN Networks[J]. *IEEE Transactions on Network and Service Management*, 2015, 12(1): 4-17.
 - [25] Lange S, Gebert S, Spoerhase J, et al. Specialized Heuristics for the Controller Placement Problem in Large Scale SDN Networks[C]. *2015 27th International Teletraffic Congress*, 2015: 210-218.
 - [26] Jiménez Y, Cervelló-Pastor C, García A J. On the Controller Placement for Designing a Distributed SDN Control Layer[C]. 2014: 1-9.
 - [27] Hock D, Hartmann M, Gebert S, et al. Pareto-optimal resilient controller placement in SDN-based core networks[C]. *The 2013 25th International Teletraffic Congress (ITC)*, 2013: 1-9.
 - [28] Wang G, Zhao Y, Huang J, et al. A K-means-based network partition algorithm for controller placement in software defined network[C]. *2016 IEEE International Conference on Communications (ICC)*, 2016: 1-6.
 - [29] Yao G, Bi J, Li Y L, et al. On the Capacitated Controller Placement Problem in Software Defined Networks[J]. *IEEE Communications Letters*, 2014, 18(8): 1339-1342.
 - [30] Guo S, Yang S, Li Q, et al. Towards Controller Placement for Robust Software-Defined Networks[C]. *IPCCC '15: The 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*. 2015: 1-8.
 - [31] Guang Y, Jun B, Luyi G. On the cascading failures of multi-controllers in Software Defined Networks[C]. *2013 21st IEEE International Conference on Network Protocols*, 2013: 1-2.
 - [32] Sallahi A, St-Hilaire M. Optimal Model for the Controller Placement Problem in Software Defined Networks[J]. *IEEE Communications Letters*, 2015, 19(1): 30-33.
 - [33] Bari M F, Roy A R, Chowdhury S R, et al. Dynamic Controller Provisioning in Software Defined Networks[C]. *The 9th International Conference on Network and Service Management*, 2013: 18-25.
 - [34] Sallahi A, St-Hilaire M. Expansion Model for the Controller Placement Problem in Software Defined Networks[J]. *IEEE Communications Letters*, 2017, 21(2): 274-277.
 - [35] Muller L F, Oliveira R R, Luizelli M C, et al. Survivor: An Enhanced Controller Placement Strategy for Improving SDN Survivability[C]. *2014 IEEE Global Communications Conference*, 2014: 1909-1915.
 - [36] Ying H, Tao L, Wenjie W, et al. On the load balanced controller placement problem in Software defined networks[C]. *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016: 2430-2434.
 - [37] Zhang B, Wang X, Ma L, et al. Optimal Controller Placement Problem in Internet-Oriented Software Defined Network[C]. *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2016: 481-488.
 - [38] Le K V, Li V O K. Modeling and Analysis of Systems with Multimode Components and Dependent Failures[J]. *IEEE Transactions on Reliability*, 1989, 38(1): 68-75.
 - [39] Hu Y, Wang W, Gong X, et al. Reliability-aware controller placement for Software-Defined Networks[C]. *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, 2013: 672-675.
 - [40] Hu Y, Wang W, Gong X, et al. On the placement of controllers in software-defined networks[J]. *The Journal of China Universities of Posts and Telecommunications*, 2012, 19: 92-171.
 - [41] Zhang Y, Beheshti N, Tatipamula M. On Resilience of Split-Architecture Networks[C]. *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, 2011: 1-6.
 - [42] Vizarrata P, Machuca C M, Kellerer W. Controller placement

- strategies for a resilient SDN control plane[C]. *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, 2016: 253-259.
- [43] Hu Y N, Wang W D, Gong X Y, et al. On Reliability-optimized Controller Placement for Software-Defined Networks[J]. *China Communications*, 2014, 11(2): 38-54.
- [44] Obadia M, Bouet M, Rougier J, et al. A greedy approach for minimizing SDN control overhead[C]. *The 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, 2015: 1-5.
- [45] Huque M T I U, Jourjon G, Gramoli V. Revisiting the controller placement problem[C]. *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, 2015: 450-453.
- [46] Hu Y, Luo T, Beaulieu N C, et al. The Energy-Aware Controller Placement Problem in Software Defined Networks[J]. *IEEE Communications Letters*, 2017, 21(4): 741-744.
- [47] Ruiz-Rivera A, Chin K W, Soh S. GreCo: An Energy Aware Controller Association Algorithm for Software Defined Networks[J]. *IEEE Communications Letters*, 2015, 19(4): 541-544.
- [48] Arya V, Garg N, Khandekar R, et al. Local search heuristic for k-median and facility location problems[C]. *The thirty-third annual ACM symposium on Theory of computing*, 2001: 21-29.
- [49] Wang G D, Zhao Y X, Huang J, et al. An Effective Approach to Controller Placement in Software Defined Wide Area Networks[J]. *IEEE Transactions on Network and Service Management*, 2018, 15(1): 344-355.
- [50] Alshamrani A, Guha S, Pisharody S, et al. Fault Tolerant Controller Placement in Distributed SDN Environments[C]. *2018 IEEE International Conference on Communications*, 2018: 1-7.
- [51] Killi B P R, Rao S V. Optimal Model for Failure Foresight Capacitated Controller Placement in Software-Defined Networks[J]. *IEEE Communications Letters*, 2016, 20(6): 1108-1111.
- [52] Killi B P R, Rao S V. Capacitated Next Controller Placement in Software Defined Networks[J]. *IEEE Transactions on Network and Service Management*, 2017, 14(3): 514-527.
- [53] Killi B P R, Rao S V. Link failure aware capacitated controller placement in software defined networks[C]. *2018 International Conference on Information Networking*, 2018: 292-297.
- [54] Katsavounidis I, Kuo C J, Zhen Z. A new initialization technique for generalized Lloyd iteration[J]. *IEEE Signal Processing Letters*, 1994, 1(10): 144-146.
- [55] Internet2 open science, scholarship and services exchange[EB/OL]. <http://www.internet2.edu/products-services/advanced-networking/layer-2-services>.
- [56] Knight S, Nguyen H X, Falkner N, et al. The Internet Topology Zoo[J]. *IEEE Journal on Selected Areas in Communications*, 2011, 29(9): 1765-1775.
- [57] Khuller S, Sussmann Y J. The Capacitated K-Center Problem[J]. *SIAM Journal on Discrete Mathematics*, 2000, 13(3): 403-418.
- [58] SNDlib: Library of test instance for survivable fixed telecommunication network design[EB/OL]. <http://sndlib.zib.de/home.action>.
- [59] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing[J]. *science*, 1983, 220(4598): 671-680.
- [60] CPLEX Optimizer[EB/OL]. <https://www.ibm.com/analytics/cplex-optimizer>.
- [61] Guo M, Bhattacharya P. Controller Placement for Improving Resilience of Software-Defined Networks[C]. *2013 Fourth International Conference on Networking and Distributed Computing*, 2013: 23-27.
- [62] Clauset A, Newman M E J, Moore C. Structure in very Large Networks[J]. *Physical Review E*, 2004, 70(6): 066111.
- [63] Gurobi Optimization I. Gurobi Optimizer reference manual[J]. URL: <http://www.gurobi.com>, 2014.
- [64] Rath H K, Revoori V, Nadaf S M, et al. Optimal controller placement in Software Defined Networks (SDN) using a non-zero-sum game[C]. *The IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, 2014: 1-6.
- [65] Ros F J, Ruiz P M. Five Nines of Southbound Reliability in Software-defined Networks[C]. *The third workshop on Hot topics in software defined networking*, 2014: 31-36.
- [66] Ros F J, Ruiz P M. On reliable controller placements in Software-Defined Networks[J]. *Computer Communications*, 2016, 77: 41-51.
- [67] Ksentini A, Bagaa M, Taleb T, et al. On using bargaining game for Optimal Placement of SDN controllers[C]. *2016 IEEE International Conference on Communications*, 2016: 1-6.
- [68] Buluç A, Meyerhenke H, Safro I, et al. Recent Advances in Graph Partitioning[M]. *Algorithm Engineering*. Cham: Springer International Publishing, 2016: 117-158.
- [69] Tanha M, Sajjadi D, Pan J. Enduring Node Failures through Resilient Controller Placement for Software Defined Networks[C]. *2016 IEEE Global Communications Conference*, 2016: 1-7.
- [70] Shen H, Zhu Y L. Adaptive Bacterial Foraging Optimization Algorithm Based on Social Foraging Strategy[J]. *Journal of Networks*, 2014, 9(3): 799-806.
- [71] Czyżżak P, Jaskiewicz A. Pareto Simulated Annealing—a Meta-heuristic Technique for Multiple-objective Combinatorial Optimization[J]. *Journal of Multi-Criteria Decision Analysis*, 1998, 7(1): 34-47.
- [72] Hadi A S, Kaufman L, Rousseeuw P J. Finding Groups in Data: An Introduction to Cluster Analysis[J]. *Technometrics*, 1992, 34(1): 111.
- [73] Jalili A, Ahmadi V, Keshtgari M, et al. Controller placement in software-defined WAN using multi objective genetic algorithm[C]. *2015 2nd International Conference on Knowledge-Based Engi-*

neering and Innovation, 2015: 656-662.

- [74] Ahmadi V, Jalili A, Khorramizadeh S M, et al. A hybrid NSGA-II for solving multiobjective controller placement in SDN[C]. *2015 2nd International Conference on Knowledge-Based Engineering and Innovation*, 2015: 663-669.
- [75] Ahmadi V, Khorramizadeh M. An Adaptive Heuristic for Multi-objective Controller Placement in Software-defined Networks[J]. *Computers & Electrical Engineering*, 2018, 66: 204-228.
- [76] Liao J X, Sun H F, Wang J Y, et al. Density Cluster Based Approach for Controller Placement Problem in Large-scale Software Defined Networkings[J]. *Computer Networks*, 2017, 112: 24-35.
- [77] Deb K, Pratap A, Agarwal S, et al. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [78] Glover F, Laguna M, Martí R. Fundamentals of Scatter Search and Path Relinking[J]. *Control and Cybernetics*, 2000, 29(3): 652-684.
- [79] Seada H, Deb K. U-NSGA-III: A Unified Evolutionary Optimization Procedure for Single, Multiple, and many Objectives: Proof-of-Principle Results[J]. *Evolutionary Multi-Criterion Optimization*, 2015: 34-49.
- [80] Wang T, Liu F, Guo J, et al. Dynamic SDN controller assignment in data center networks: Stable matching with transfers[C]. *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016: 1-9.
- [81] Levin D, Wundsam A, Heller B, et al. Logically centralized?: state distribution trade-offs in software defined networks[C]. *The first workshop on Hot topics in software defined networks*, 2012: 1-6.
- [82] Guo Z, Su M, Xu Y, et al. Improving the performance of load balancing in software-defined networks through load variance-based synchronization[J]. *Computer Networks*, 2014, 68: 95-109.
- [83] Ho C, Wang K, Hsu Y. A fast consensus algorithm for multiple controllers in software-defined networks[C]. *2016 18th International Conference on Advanced Communication Technology*, 2016: 1.
- [84] Abdel-Rahman M J, Mazied E A, Mackenzie A, et al. On Stochastic Controller Placement in Software-Defined Wireless Networks[C]. *2017 IEEE Wireless Communications and Networking Conference*, 2017: 1-6.
- [85] Qin Q, Poularakis K, Iosifidis G, et al. SDN Controller Placement at the Edge: Optimizing Delay and Overheads[C]. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018: 684-692.
- [86] Papa A, Cola T D, Vizarreta P, et al. Dynamic SDN Controller Placement in a LEO Constellation Satellite Network[C]. *2018 IEEE Global Communications Conference*, 2018: 206-212.



蔡宁 于 2014 年在四川大学通信工程专业获得学士学位。现在中国科学院信息工程研究所信号与信息处理专业攻读博士学位。研究领域为软件定义网络。研究兴趣包括: 流表压缩、控制器放置。Email: caining@iie.ac.cn



韩言妮 于 2010 年在北京航空航天大学计算机专业获得博士学位。现任中国科学院信息工程研究所第五研究室副研究员。研究领域为 SDN 网络、网络安全。研究兴趣包括: 安全防御、新型网络架构。Email: hanyanni@iie.ac.cn



安伟 于 2012 年在华东理工大学控制科学与工程专业获得工学博士学位。现任中国科学院信息工程研究所第五研究室助理研究员。研究领域为网络与系统安全。研究兴趣包括: 网络优化、网络安全及防护、5G 移动通信与安全。Email: anwei@iie.ac.cn



徐震 于 2005 年在中国科学院软件研究所信息安全专业获得工学博士学位。现任中国科学院信息工程研究所第五研究室正高级工程师。研究领域为网络体系结构与安全防护。研究兴趣包括云安全、可信计算。Email: xuzhen@iie.ac.cn