

基于逆向拍卖的 SDN 多控制器负载均衡研究

陈 虹¹
CHEN Hong

摘 要

随着网络规模爆炸式的增长, 软件定义网络 (software defined network, SDN) 控制平面面临严重负载不均衡问题, 严重影响 SDN 网络性能。基于此, 文章提出了一种结合逆向拍卖机制与模拟退火算法的多控制器负载均衡算法 (SMLBRA), 通过动态资源分配优化网络性能。仿真结果表明, SMLBRA 在响应时延和迁移效率上显著优于传统方法。

关键词

SDN; 负载均衡; 逆向拍卖; 模拟退火

doi: 10.3969/j.issn.1672-9528.2025.06.023

0 引言

软件定义网络^[1] (software defined network, SDN) 通过解耦控制与数据平面提升网络灵活性, 极大促进了网络流量的可靠性。但大规模流量场景下, 单一控制器易因过载引发响应延迟和资源浪费。同时现有负载均衡方案难以适应动态流量变化, 且缺乏全局优化视角。

SDN 多控制器网络架构中, 多控制器负载均衡问题本质上是各个控制器之间的资源分配问题。潘志安等人^[2]提出了一种动态检测控制器负载的方法, 结合改进蚁群算法选择最佳目标控制器, 智能调整交换机迁移路径, 减少任务分配时间。管吉喆等人^[3]提出了一种基于交换机迁移的动态防御方法, 通过实时监测控制器负载, 优先迁移攻击路径上的关键交换机至低负载控制器, 实验降低了控制器过载次数和迁移开销。刘向举等人^[4]提出一种基于过程优化的动态负载均衡策略, 通过优化迁移交换机 3 个阶段减少了迁移开销和时延, 提高了网络性能。黄泰铭等人^[5]提出了一种动态负载均衡策略, 通过动态调整数据采集频率和优化迁移算法, 使各控制器负载趋于均衡, 降低网络资源消耗。

本文引入经济学中的逆向拍卖机制^[6], 将过载控制器作为买方、轻载控制器作为卖方, 通过竞价策略实现资源高效匹配, 并结合模拟退火算法优化迁移路径, 最终构建轻量化、自适应的负载均衡模型。通过模拟退火算法来求出过载控制器与轻载控制器之间的最优对应关系, 根据交换机迁移概率选择适当的交换机进行迁移使控制平面达到负载均衡, 仿真对本文提出的算法进行验证其优越性。

1 基于逆向拍卖的 SDN 多控制器模型

本文提出的模型将逆向拍卖思想运用到 SDN 多控制器

模型而建立, 基于逆向拍卖的 SDN 多控制器模型图如图 1 所示, 主要由 3 部分组成: 负载监控模块、负载整合模块、负载决策模块。

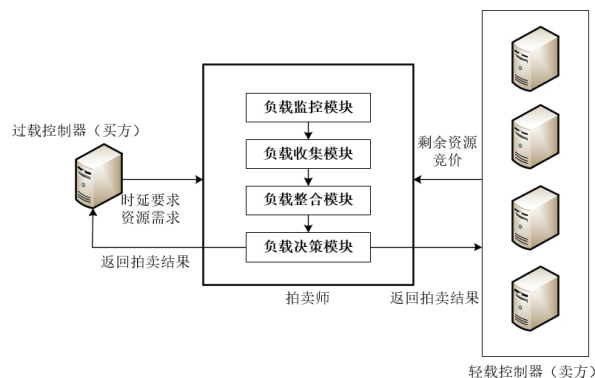


图 1 基于逆向拍卖的 SDN 多控制器模型

1.1 负载监控模块

负载监控模块负责周期性地收集负载信息, 通过监测控制器对 Packet-in 消息的最大处理速率, 实现对控制器的过载判断。第 i 个控制器在时间 t 所接收到的流请求数是与其相连的所有交换机的流请求数之和, 用 $L(t)_i$ 表示, 其中各个交换机之间的流请求相互独立, 用公式表示为:

$$L(t)_i = \sum_{j=1}^n t \lambda(t)_{ij} \quad (1)$$

负载监控模块实时收集各个控制器的负载信息后, 分别向过载控制器和轻载控制器发送确认消息, 控制器同时向负载监控模块反馈负载状态。每个控制器都具有一个负载阈值 T , 用于表示各个控制器每秒钟最多接收到 T 个 Packet-in 信息, 当某一时刻控制器收集到的负载信息 $L(t)_i$ 超出负载阈值 T (本文定义控制器负载阈值为控制器最大承受负载的 85%) 时, 将其确定为过载控制器。同时负载收集模块将其收集到的过

1. 山西工程技术学院大数据与智能工程系 山西阳泉 045000

载信息发送到负载整合模块。

1.2 负载整合模块

负载整合模块将需求信息形成一个二元组 $R=(\text{delay}, \text{Rre})$, 其中, delay 表示过载控制器时延需求; Rre 表示资源需求, 资源即为控制器接收交换机 Packet-in 消息的能力。同时将形成的二元组 $R=(\text{delay}, \text{Rre})$ 发送给轻载控制器, 当轻载控制器向拍卖师发出竞价信息。同时负载整合模块根据控制器的负载信息确定具有接收 Packet-in 消息的能力的控制器为轻载控制器, 并将所有轻载控制器的负载能力进行从大到小的排序。根据过载控制器的时延需求以及资源需求, 选出满足条件的轻载控制器作为卖方集合。

1.3 负载决策模块

本文主要考虑的资源特性有: 剩余资源、开始价格、最低价格。卖方将资源特性以三元组的形式整合, 即 $\text{Bid}_i=(\text{Reu}, \text{sprice}, \text{lprice})$, 其中 Reu 表示轻载控制器的剩余资源; sprice 表示资源的单位价格; lprice 表示资源的最低价格。目标控制器选择流程如图 2 所示。

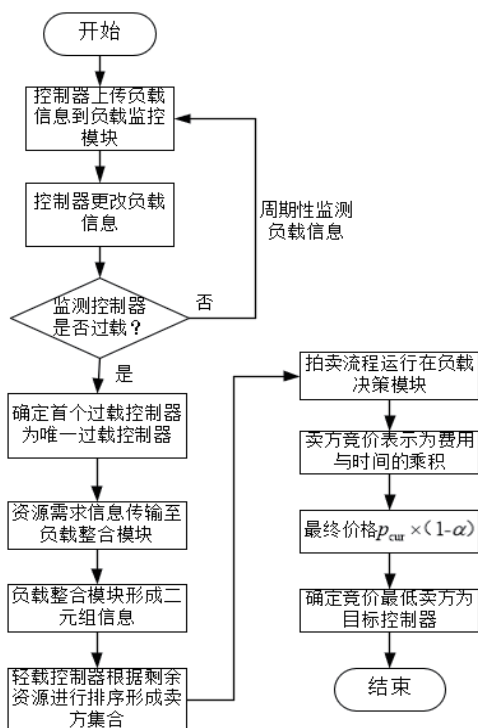


图 2 目标控制器选择流程图

负载决策模块完成具体的拍卖流程, 选出目标控制器:

(1) 买方将需求信息提交, 包括时延需求以及过载资源需求。

(2) 拍卖师接收到买方需求信息后, 并以元组的形式发送到轻载控制器进行拍卖。

(3) 卖方进行竞标资源, 其中的资源信息包括: 轻载

控制器剩余资源, 资源单位价格, 资源最低价格。

(4) 拍卖师根据卖方提供的资源信息, 计算出卖方完成资源分配所需要的最小费用以及完成时间。将费用与完成时间的乘积作为评价标准, 乘积最低的卖方为本轮的赢家。

(5) 拍卖师将此轮拍卖的结果分别发送到买方与卖方。对于赢家, 其资源价格不变。对于输家, 将其资源价格按照一定程度调整降低, 使其最终价格为 $p_{\text{cur}} \times (1-\alpha)$, 其中 p_{cur} 为本轮价格, α 为降低率。直到没有卖家提出更低竞价完成拍卖, 竞价最低的轻载控制器为目标控制器。

2 SMLBRA 算法实现

2.1 模型构建

建立 SDN 网络拓扑图 $G=(V, E)$ 作为实验模型, 其中 V 代表 SDN 网络中所有的网络节点, 即由控制器与交换机所构成的网络节点; E 代表 SDN 网络中所有网络节点所连接的边, 即由控制器与交换机之间的连接构成网络拓扑的边。假设网络拓扑中控制器个数为 M , 交换机个数为 N 。所以控制器集合可以表示为: $C=\{c_1, s_2, \dots, c_M\}$, $c_i \in V$, $i=1, 2, \dots, M$, 交换机集合可以表示为: $S=\{s_1, s_2, \dots, s_N\}$, $s_j \in V$, $j=1, 2, \dots, N$ 。将 d_{ij} 定义为控制器 c_i 与交换机 s_j 的最短路径距离, 用于表示控制器与交换机之间的通信方式。 x_{ij} 表示控制器 c_i 是否管理交换机 s_j , 是为 1, 否则为 0。

定义网络平均时延。假设 c_i' 为过载控制器, c_{object} 为目标控制器, $d(ij)$ 为控制器 c_i 与交换机 s_j 之间的通信路径。 $D(u, v)$ 表示节点 u 与节点 v 之间的传输时延, $D(\text{object}, j)$ 表示交换机 s_j 与目标控制器 c_{object} 的传输时延, 如式 (2) 所示。 n 表示过载控制器 c_i 管理域内需要迁移的交换机个数, D_{avg} 表示过载控制器 c_i 域中所需要迁移交换机到目标控制器的平均时延, 如式 (3) 所示。

$$D(\text{object}, j) = \sum_{(u,v) \in d(ij)} D(u, v), \forall i \in c_{\text{object}}, j \in s_j \quad (2)$$

$$D_{\text{avg}} = \frac{1}{n} \sum_{j \in s_j} (D(\text{object}, j) \times x_{ij}), \forall i \in c_{\text{object}} \quad (3)$$

定义网络开销。流转发开销和负载状态同步开销共同构成网络开销。控制器处理 Packet-in 数据包带来的网络开销是构成流转发开销的主要部分, 公式表示为:

$$P_s = v_c \sum_{i=1}^N \sum_{j=1}^M \lambda(t) d_{ij} x_{ij} \quad (4)$$

式中: v_c 表示数据包到控制器的平均速率; $\lambda(t)$ 表示交换机到控制器在 t 时刻的流请求速率。

负载状态同步开销用公式表示为:

$$P_c = v_s \sum_{i,j \in N} d_{ij} x_{ij} \quad (5)$$

式中: v_s 表示控制器状态信息的平均传输速率。

所以可以将网络开销表示为:

$$P(t) = P_s + P_c \quad (6)$$

为实现 SDN 多控制器负载均衡, 将优化平均时延和网络开销作为优化目标, 使用模拟退火算法来实现交换机的迁移, 同时获得控制器与交换机的重映射, 重新达到网络负载均衡状态。将算法的目标函数与约束条件可以表示为:

$$\min F = \beta D_{\text{avg}} + (1 - \beta) P(t) \quad (7)$$

$$x_{ij} \in \{0, 1\}, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (8)$$

$$\beta \in (0, 1) \quad (9)$$

2.2 多控制器资源分配过程

多控制器资源分配过程由 SMLBRA 算法实现。SMLBRA 算法基本流程: 首先根据控制器的负载约束条件搭建自己的匹配列表。然后使用模拟退火算法将过载控制器下的交换机根据目标函数迁移到目标控制器管理域中, 从而将控制器资源重新分配, 直到 SDN 网络中控制平面达到负载均衡, 获得控制器与交换机的重映射关系。算法初始温度、降温函数以及接收函数的设置是影响算法性能优劣的主要因素。

SMLBRA 算法流程图如图 3 所示。

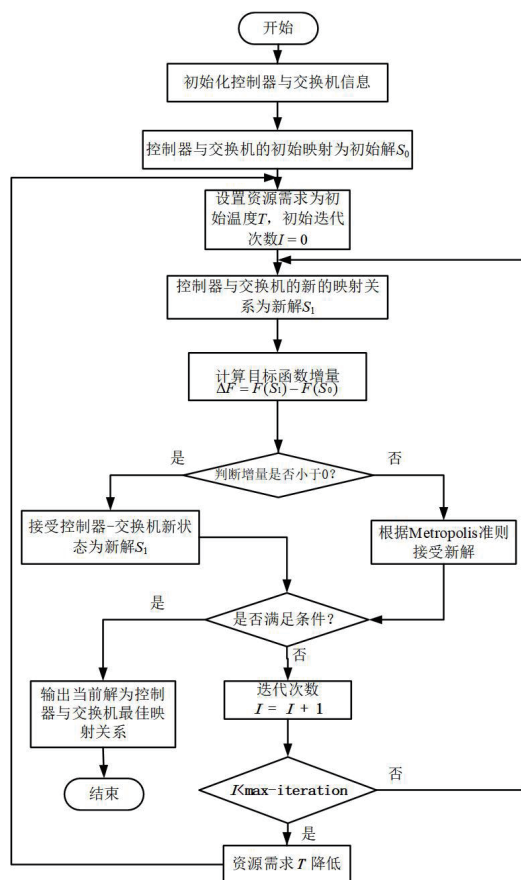


图 3 SMLBRA 算法流程图

算法将综合考虑最小化平均时延与网络开销的多控制器负载均衡问题比作模拟退火算法中的退火过程; 目标函数的性能指标比作能量; 最小化网络平均时延与网络开销为能量

最低状态; 控制器与交换机之间的初始映射关系为初始解空间; 过载控制器资源需求为初始温度。具体实现步骤如下:

(1) 初始化: 设定过载控制器资源需求 T_0 为初始温度, 当前温度为 T ; 确定初始解 S_0 为控制器与交换机初始映射集合; 每个温度下最大迭代次数为 max-iteration, 当前迭代次数为 I 。

(2) 若 $I < \text{max-iteration}$, 当前温度 T 下产生新解 S_1 : 即在初始解 S_0 的基础上选择过载控制器中的交换机迁移到目标控制器; 若 $I > \text{max-iteration}$, 转到 (6)。

(3) 计算目标函数增量: 计算当前解 S_0 状态下的目标函数值为 $F(S_0)$, 计算新解 S_1 状态下的目标函数为 $F(S_1)$, 增量 $\Delta F = F(S_1) - F(S_0)$ 。

(4) 使用 Metropolis 准则进行判断: 如果 $\Delta F > 0$, 接收 S_1 为新解; 如果 $\Delta F < 0$, 基于 Metropolis 准则, 按照概率 $p = \exp(-\Delta F/T)$ 判断, 将随机产生 (0,1) 之间的一个随机数 r , 若 $\exp(-\Delta F/T) > r$, 则接受新解 S_1 作为控制器与交换机的重映射关系。

(5) 当迭代次数到达 max-iteration 且满足预设条件, 算法结束, 输出当前解为最优解, 即控制器与交换机实现最佳重映射关系, 重新达到控制器负载均衡; 否则, $I = I + 1$ 。

(6) 降温操作: 若未达到结束条件, 当前温度 T 按照降温函数下降, 重复 (2) ~ (6), 直到满足终止条件为止。

3 仿真与实验分析

3.1 仿真环境设置

采用 Mininet^[7] 仿真平台作为实验仿真平台, 将本文设计的负载均衡算法应用在该环境中完成实验测试。实验架构如图 4 所示。

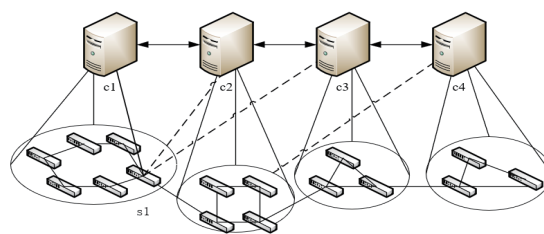


图 4 单控制器过载 SDN 网络架构

实验构建 4 个控制器: 分别是控制器 c1、c2、c3、c4, 4 个控制器共同管理 16 个交换机。如图 4 中交换机 s1 与控制器 c1、c2、c3、c4 相连, 其中 c1 为 s1 的 Master 控制器, c2、c3、c4 为 s1 的 Slave 控制器。为达到更好的仿真效果, 测试单个控制器每秒能处理 16 298 个 Packet-in 数据包。系统负载阈值设定为 85%, 即设置控制器处理容量最大值为 13 853 个 Packet-in 数据包, 实验循环测试 20 次, 得到测试平均值。

设置 3 种不同的仿真实验, 分别从控制器资源利用率、

响应时延与负载迁移完成时间方面将 SMLBRA 算法与 STATIC、ECSM 和 SMDM 算法进行对比。

表 1 仿真对比算法

算法	简介
STATIC	静态负载均衡算法
ECSM	交换机采用就近迁移操作, 交换机就近选择控制器进行迁移
SMDM	从迁出控制器集合中选择交换机迁移至迁入控制器集合中
SMLBRA	本文所提算法

3.2 响应时延对比

实验通过改变 Packet-in 消息的产生速率来测试控制器的响应时延, 通过计算每个 Packet-in 数据包的往返时间可以得到控制器响应时延。实验仿真通过改变 c1 控制域内的交换机的 Packet-in 请求速率来测试控制器响应时延, 并且对不同算法的 Packet-in 消息的响应时延进行对比, 验证 3 种算法在降低控制器响应时延方面的可行性与有效性。实验结果如图 5 所示。

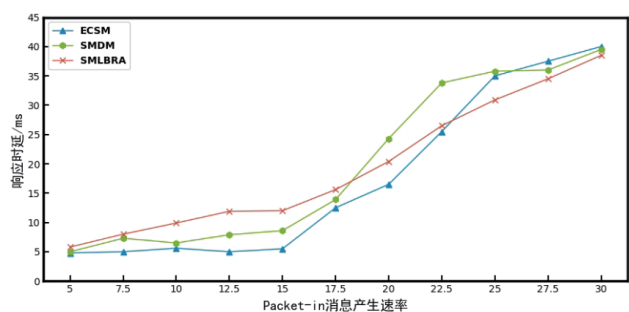


图 5 ECSM、SMDM 与本算法系统响应时延对比

当网络中 Packet-in 产生速率大于 20 Packets/ms 时, 控制器出现过载状况, 随着 Packet-in 产生速率的不断增加, 控制器过载程度不断增大, 造成控制器的响应时延急剧增加, 此时 ECSM 算法和 SMDM 算法相较于 SMLBRA 算法响应时间更长, 同时 SMLBRA 算法会避免 SMDM 算法中由于误判控制器过载数量, 导致时延增大的情况。

3.3 迁移时间对比

本次实验为单控制器过载, 未考虑在角色转换的时间间隔内出现其他控制器过载的状况。为对算法进行全面评价, 本实验通过在不同的 Packet-in 消息产生速率下, 进行 20 次实验, 计算交换机完成迁移所需时间取平均值得到实验结果。实验结果如图 6 所示, 当 Packet-in 消息的产生速率大于 27.5 Packets/ms 时, 随着需要迁移的交换机数量增多, 如果每次只迁移一个交换机, 会增加迁移交换机的数量和次数, 同时增加了网络的通信代价, SMLBRA 算法能够每次迁移多

个交换机提高算法的收敛速度, 相较于 ECSM 算法和 SMDM 算法能够时间较短地达到控制器负载均衡状态, 从而较好地实现控制平面负载均衡。

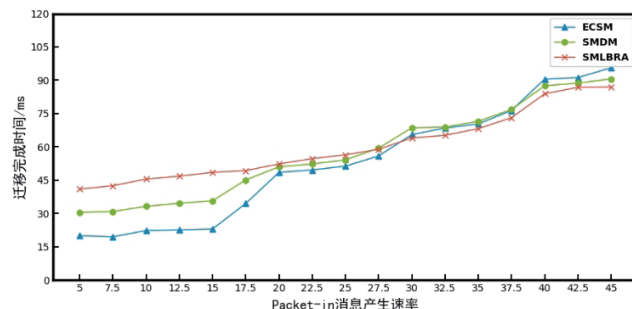


图 6 ECSM、SMDM 与本算法负载迁移完成时间对比

4 结语

针对 SDN 多控制在负载均衡过程中由于单个控制器过载会造成控制平面资源分配不均衡问题, 结合拍卖机制, 本文提出的 SMLBRA 算法, 将逆向拍卖理论与 SDN 网络架构相结合构建模型, 同时结合模拟退火算法, 达到网络最小开销成本。通过仿真实验与性能分析, 相较于 ECSM 算法和 SMDM 算法, SMLBRA 算法能够有效地降低网络响应时延, 在降低交换机迁移完成时间方面具有优势, 使得控制器负载分配更加均衡。

参考文献:

- [1] 杨耀忠, 刘宝军, 段鸿杰, 等. 基于 SDN 的数据中心流量负载均衡 [J]. 计算机应用与软件, 2022, 39(11):130-135.
- [2] 潘志安, 王小英, 王茂发. 基于改进蚁群算法的多控制器负载均衡仿真 [J]. 计算机仿真, 2023, 40(10):321-325.
- [3] 管吉喆, 程光, 周余阳. 基于交换机迁移的控制平面饱和攻击防御方法 [J]. 计算机学报, 2024, 47(12):2889-2908.
- [4] 刘向举, 赵霖, 方贤进, 等. SDN 中基于过程优化的动态负载均衡策略 [J]. 计算机工程, 2023, 49(8):137-145.
- [5] 黄泰铭, 王雪, 于波, 等. SDN 多控制器网络中基于交换机迁移的动态负载均衡策略 [J]. 计算机与数字工程, 2021, 49(7): 1389-1395.
- [6] 胡婷婷, 朱晓娟. 基于逆向拍卖模型的群智感知激励机制 [J]. 佳木斯大学学报 (自然科学版), 2020, 38(3):114-119.
- [7] 肖桂霞. Mininet 网络拓扑类型及其构造方法综述 [J]. 教育现代化, 2020, 7(50):16-18.

【作者简介】

陈虹 (1993—), 女, 山西晋中人, 硕士, 助教, 研究方向: 计算机网络、信息安全。

(收稿日期: 2025-03-03 修回日期: 2025-06-11)