

# SDN 中基于效能优化的交换机动态迁移策略

姚 蓝, 胡 涛, 伊 鹏, 胡宇翔, 兰巨龙, 李子勇

(国家数字交换系统工程技术研究中心, 河南郑州 450002)

**摘 要:** 交换机迁移作为一种弹性控制方式对于软件定义网络(SDN)中多控制器负载均衡具有重要作用. 针对现有迁移方案存在迁移效率低下和高迁移代价问题, 提出基于效能优化的交换机动态迁移策略. 首先, 分析控制器负载组成并构建负载差异矩阵, 设置触发因子进行负载不均衡判定. 然后确定迁移对象, 建立迁移效能模型, 同时考虑负载均衡率和迁移代价来确定迁移交换机和迁入控制器. 最后, 通过设定迁移三元组完成迁移映射, 实现高效交换机迁移. 仿真结果表明, 该策略可以有效降低控制器响应时间, 减小迁移代价并提高控制器吞吐量, 同时将负载均衡率保持在较高水平, 具有良好的拓扑适应性.

**关键词:** 软件定义网络; 交换机迁移; 多控制器; 网络优化

**中图分类号:** TP393      **文献标识码:** A      **文章编号:** 0372-2112 (2019)07-1482-08

**电子学报 URL:** <http://www.ejournal.org.cn>      **DOI:** 10.3969/j.issn.0372-2112.2019.07.011

## Switch Dynamic Migration Strategy Based on Efficiency Optimization in SDN

YAO Lan, HU Tao, YI Peng, HU Yu-xiang, LAN Ju-long, LI Zi-yong

(National Digital Switching System Engineering & Technology Research Center, Zhengzhou, Henan 450002, China)

**Abstract:** As an elastic control approach, switch migration plays an important role in multi-controller load balancing in software defined networking (SDN). In order to solve the problems of low migration efficiency and high migration cost in the existing schemes, this paper proposes Switch Dynamic Migration (SDM) Strategy based on efficiency optimization. We firstly analyze the composition of controller loads in detail, set up the load difference matrix to know the distribution of loads, and introduce a trigger factor, which is used to judge load balancing on controllers through comparing the overload threshold. Then, we formulate the migration objects, and introduce the migration efficiency model, which considers load balance rate and migration cost simultaneously to select the migrating switches and immigrating controllers. Finally, we set migration triplet to complete the migration mapping and achieve the efficient switch migration. Simulation results show this mechanism could lower controller response time effectively, reduce migration cost and improve controller throughputs. Besides, it keeps high load balancing rate and has good topology adaptability.

**Key words:** software defined networking; switch migration; multi-controller; network optimization

### 1 引言

软件定义网络 (Software Defined Networking, SDN) 作为一种新型网络架构, 实现了数据平面和控制平面的完全解耦<sup>[1]</sup>. 随着网络规模的不断扩大和业务需求的迅速增长<sup>[2]</sup>, 集中式单控制器难以适应 SDN 的大规模应用. 因此研究人员相继提出逻辑上集中, 物理上分布的多控制器部署架构<sup>[3-5]</sup>来提升控制平面的可扩展性和可靠性. Heller<sup>[7]</sup>从控制器部署角度考虑, 基于平均时延和最大时延优化控制器位置, 并引入部署实例分

析负载分布状况. Hock<sup>[8]</sup>设计了基于帕累托的最优控制器布局方式 (Pareto-based Optimal Controller, POCO), 在控制器性能、失效容忍、负载均衡等方面做出折衷. Bari<sup>[9]</sup>提出建立动态控制器规划架构, 该架构能够根据负载情况调整处于激活状态的控制器数量, 同时最小化流建立时间. Fu<sup>[10]</sup>提出多控制器休眠模型, 在轻流量状态下允许备份控制器进入休眠状态以节省网络开销.

继 Open Flow 1.4 协议<sup>[11]</sup>提出之后, 文献[12]设计了 ElastiCon 架构, 并提出了一种新颖的交换机迁移协议, 依

收稿日期: 2017-12-05; 修回日期: 2018-05-30; 责任编辑: 李勇锋

基金项目: 国家网络空间安全专项课题 (No. 2017YFB0803204); 国家 863 高技术研究发展计划 (No. 2015AA016102); 国家自然科学基金创新研究群体科学基金 (No. 61521003); 国家自然科学基金 (No. 61802429, No. 61872382)

据距离最近原则(Distance Closest Migration, DCM),在控制器间实施交换机迁移以实现 OpenFlow 标准的控制器负载转移. 文献[13]提出基于集群控制器的交换机迁移机制,整个网络划分为多域,每个域设置相应的控制器集群,通过交换机迁移实现控制器负载在多个集群间的动态分配. 文献[14]提出一种迁移优化策略,通过设定子域迁移度和交换机有效期,实现了多交换机的协调迁移,但没有考虑迁移代价对整个网络性能的影响. 文献[15]将交换机迁移规划为网络资源利用率最大问题,设计了一个分布式跳数算法,在迁移交换机时主要考虑控制器的容量因素,将交换机迁移至剩余处理容量最大的控制器. Guo<sup>[16]</sup>提出基于负载方差同步的负载均衡方案,这显著降低了控制器的同步开销,实现了无环路转发和良好的负载均衡性能. Chen<sup>[17]</sup>设计了基于博弈的交换机迁移方案,但博弈域的形成和博弈过程造成额外网络开销较大.

在实际网络环境中,流量在时间和空间的分布上具有突发性和不均匀性<sup>[6]</sup>,造成不同子域内控制器需要处理的流请求数量差异较大. 交换机迁移作为一种弹性控制方案可以有效解决网络中控制器负载分布不均衡问题. 通过迁出过载控制器管理的交换机到轻载控制器,实现控制器负载的动态调整. 为了避免交换机迁移造成网络性能不稳定,迁移方案必须经过精心设计. 然而,在交换机迁移问题研究过程中,大多数方案的迁移效率较为低下,迁移完成后控制器负载均衡性能提升不够显著,并且在迁移过程中容易产生大量的额外网络代价,反而加重了控制器负担.

针对上述问题,本文提出了一种基于效能优化的交换机动态迁移(Efficiency Optimization Switch Dynamic Migration, EOSDM)策略. 本文主要贡献和创新工作总结如下:

(1)对控制器负载不均衡现象进行描述,结合现有交换机迁移方案,分析发现网络中存在交换机迁移效率低下问题,主要包括负载均衡性能提升有限和高交换机迁移代价.

(2)提出一种基于效能优化的交换机动态迁移策略(EOSDM). 通过分析控制器负载构成,构建负载差异矩阵并计算触发因子判定负载不均衡,进而引入迁移效能模型,优化迁移交换机和迁入控制器选取,最终确定控制器间的交换机迁移映射,实现高效交换机迁移.

(3)为 EOSDM 策略设计相应的两阶段实施算法,同时在仿真实验中,与现有的代表性迁移策略进行比较,综合多种评价指标,基于真实拓扑场景开展实验研究.

## 2 分析与建模

本节首先对现有交换机迁移方案中存在的问题进行描述;然后介绍 EOSDM 策略,定义和计算相关参数.

### 2.1 问题描述

如图 1 所示,整个网络共部署 3 台控制器,交换机和控制器之间的连接关系如表 1 所示,网络被划分为 3 个子域,交换机 S1 流量突发导致控制器 C1 成为过载控制器(总负载大于控制器 C1 处理能力).

传统交换机迁移方案将流请求速率最高的交换机由过载控制器所在子域迁移到具有最低控制器负载的子域,即将 S1 由 Domain1 迁移到 Domain3,迁移完成后的负载分布如表 2 所示. 可以看出,尽管通过迁移 S1 消除了过载控制器,但迁移完成后的控制器负载状况呈现严重的不均衡状态,C3 的负载等于 C1 和 C2 的负载之和.

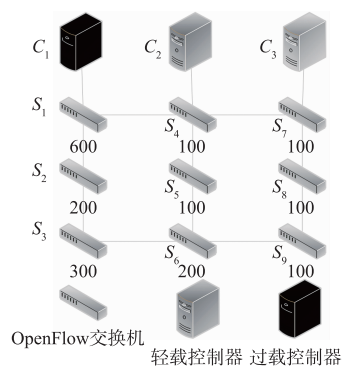


图1 具有三个控制器的多域SDN网络

针对现有交换机迁移方案中存在的问题,本文从提高控制器负载均衡率和降低迁移代价的角度出发,提出基于效能优化思想的交换机动态迁移方案. 具体思路如下:通过将具有低请求速率的 S2 和 S3 分别迁移到 C2 和 C3,可以有效降低控制器 C1 的负载,避免产生高额的网络代价,同时也能保证网络的控制器负载处于均衡分布状态,如表 3 所示.

表 1 原始网络状态下交换机和控制器之间的连接关系

域	Domain1			Domain2			Domain3		
控制器	C1			C2			C3		
交换机	S1	S2	S3	S4	S5	S6	S7	S8	S9
流请求 (packets/ms)	600	200	300	100	100	200	100	100	100
控制器负载状况	600 + 200 + 300 = 1100 > 1000			100 + 100 + 200 = 400 < 1000			100 + 100 + 100 = 300 < 1000		

表 2 传统交换机迁移方案后交换机和控制器之间的连接关系

域	Domain1		Domain2			Domain3			
控制器	C1		C2			C3			
交换机	S2	S3	S4	S5	S6	S1	S7	S8	S9
流请求 (packets/ms)	200	300	100	100	200	600	100	100	100
控制器负载状况	200 + 300 = 500 < 1000		100 + 100 + 200 = 400 < 1000			600 + 100 + 100 + 100 = 900 < 1000			

表 3 基于效能优化方案后交换机和控制器之间的连接关系

域	Domain1	Domain2				Domain3			
控制器	C1	C2				C3			
交换机	S1	S2	S4	S5	S6	S3	S7	S8	S9
流请求 (packets/ms)	600	200	100	100	200	300	100	100	100
控制器负载状况	600 < 1000	200 + 100 + 100 + 200 = 600 < 1000				300 + 100 + 100 + 100 = 600 < 1000			

## 2.2 EOSDM 策略

基于效能优化的交换机动态迁移(EOSDM)主要分为三个阶段进行实施. 阶段 1, 收集网络中控制器负载信息, 测量负载差异并设置触发因子以判定控制器负载是否均衡. 阶段 2, 构建迁移效能模型, 权衡负载均衡率和迁移代价, 用于指导可能实施的交换机迁移. 阶段 3, 设定迁移方案并规划迁移操作.

### 2.2.1 控制器负载与迁移触发

SDN 网络拓扑用无向图  $G = (V, E)$  表示,  $V$  和  $E$  分别表示网络中的节点集合和链路集合.  $G$  包含  $M$  个控制器  $C = \{c_1, c_2, \dots, c_M\}$  和  $N$  个交换机  $S = \{s_1, s_2, \dots, s_N\}$ , 其中  $|V| = M + N$ . 假设控制器可以优化地部署在网络拓扑中<sup>[8]</sup>, 一个控制器管理多个交换机形成 SDN 子域, 控制器  $c_m$  的处理容量为  $\eta_m$ ,  $U_{c_m} \in S$  定义为控制器  $c_m$  管理的交换机集合. 网络设备之间的跳数定义  $d$ , 设交换机  $s_i$  和控制器  $c_m$  的连接关系为  $x_{im}$ , 如式(1)所示.

$$x_{im} = \begin{cases} 1, & s_i \text{ is connected to } c_m \\ 0, & \text{others} \end{cases} \quad (1)$$

在 SDN 网络, 控制器负载主要包含数据交互, 路由规则和状态同步三部分内容.

**定义 1** 数据交互代价. 为了实现集中式控制, 子域控制器需要和交换机进行周期性信息交互, 收集各交换机的跳数和流量信息. 设控制器  $c_m$  的数据交互代价为  $\Gamma_{\text{data}}^{c_m}$ , 如式(2)所示, 其中  $v_{\text{SW}}$  是轮循一个交换机的平均速率, 它取决于每个交换机所连接的链路数量.

$$\Gamma_{\text{data}}^{c_m} = \sum_{i \in U_{c_m}} d_{im} \cdot v_{\text{SW}} \cdot x_{im} \quad (2)$$

**定义 2** 路由规则代价. 当交换机收到新流量时, 它会向控制器发送 PACKET-IN 消息<sup>[1]</sup>, 并要求控制器计算流量路径, 下发路由表项, 由此在控制器中产生路由规则代价  $\Gamma_{\text{route}}^{c_m}$ , 如式(3)所示, 其中  $f_{\text{PACKET}}$  为交换机发

送的 PACKET-IN 数据包平均大小,  $\lambda_k$  为交换机  $s_k$  的流请求速率.

$$\Gamma_{\text{route}}^{c_m} = f_{\text{PACKET}} \cdot \sum_{s_i \in S} \sum_{c_n \in C} \lambda_k \cdot d_{km} \cdot x_{km} \quad (3)$$

**定义 3** 状态同步代价. 在分布式 SDN 网络中, 控制器间发送同步消息来维护全局网络视图, 产生状态同步代价  $\Gamma_{\text{state}}^{c_m}$ , 如式(4)所示. 其中  $\mu_{\text{syn}}$  是控制器间共享的网络数据, 与  $f_{\text{PACKET}}$  有关, 但  $\mu_{\text{syn}} < f_{\text{PACKET}}$ , 因为控制器不会共享子域所有信息.

$$\Gamma_{\text{state}}^{c_m} = \mu_{\text{syn}} \cdot \sum_{c_m, c_n \in C} x_{mn} \cdot d_{mn} \quad (4)$$

因此, 网络中控制器负载是上述三类代价的线性聚合, 控制器  $c_m$  的负载  $\Gamma_{c_m}$  计算如式(5)所示.

$$\Gamma_{c_m} = \Gamma_{\text{data}}^{c_m} + \Gamma_{\text{route}}^{c_m} + \Gamma_{\text{state}}^{c_m} \quad (5)$$

控制器  $c_m$  和控制器  $c_n$  之间的负载差异为:

$$q_{c_m, c_n} = \frac{\Gamma_{c_m}}{\Gamma_{c_n}} \quad (6)$$

基于上述设定, 得到控制器的负载差异矩阵  $\mathbf{Q}_{M \times M}$ , 如式(7)所示.

$$\mathbf{Q}_{M \times M} = \begin{Bmatrix} 1 & q_{c_1, c_2} & q_{c_1, c_3} & \cdots & q_{c_1, c_{M-1}} & q_{c_1, c_M} \\ q_{c_2, c_1} & 1 & q_{c_2, c_3} & \cdots & q_{c_2, c_{M-1}} & q_{c_2, c_M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{c_{M-1}, c_1} & q_{c_{M-1}, c_2} & q_{c_{M-1}, c_3} & \cdots & 1 & q_{c_{M-1}, c_M} \\ q_{c_M, c_1} & q_{c_M, c_2} & q_{c_M, c_3} & \cdots & q_{c_M, c_{M-1}} & 1 \end{Bmatrix} \quad (7)$$

对于给定的负载差异阈值  $\sigma$ , 交换机迁移触发条件如式(8)所示,  $\varphi_{mn}$  为触发因子.

$$\exists c_m, c_n \in C, \varphi_{mn} = (q_{c_m, c_n} - q_{c_n, c_m}) > \sigma \quad (8)$$

从式(8)可以看出, 如果触发因子  $\varphi_{mn}$  超过预设阈值  $\sigma$ , 即认为网络中出现控制器负载不均衡, 同时触发交换机迁移. 因此负载差异阈值对于调节网络中控制器负载分布具有重要意义.

负载差异阈值  $\sigma$  的取值和控制器的最大负载和最

小负载都有关系. 因此, 负载差异阈值的计算如式(9)所示, 其中  $\max \mathbf{Q}_{M \times M}$  为矩阵  $\mathbf{Q}_{M \times M}$  中元素最大值,  $\min \mathbf{Q}_{M \times M}$  为矩阵  $\mathbf{Q}_{M \times M}$  中元素最小值.

$$\sigma = \frac{\max \mathbf{Q}_{M \times M} - \min \mathbf{Q}_{M \times M}}{\max \mathbf{Q}_{M \times M}} \quad (9)$$

为了有效提高迁移效率, 减少迁移复杂度, 基于负载差异矩阵  $\mathbf{Q}_{M \times M}$  确定迁移对象. 将向外迁移交换机的控制器定义为迁出控制器, 接收交换机的控制器定义为迁入控制器. 因此, 通过对  $\mathbf{Q}_{M \times M}$  进行检测, 得到迁出控制器集合  $C_{OM}$  和迁入控制器集合  $C_{IM}$ . 如果  $\varphi_{mn} > \sigma$ , 则控制器  $c_m$  和  $c_n$  将被分别添加到集合  $C_{OM}$  和  $C_{IM}$  中.

### 2.2.2 迁移效能模型

假设所有控制器通过状态交互已获取全局网络视图, 当交换机迁移发生时, 网络  $G$  中控制器负载均衡性能提升, 但同时也会产生额外代价.

**定义 4** 迁移代价. 当交换机  $s_k$  从控制器  $c_m$  迁移到控制器  $c_n$  时, 产生的迁移代价  $L_{s_k}^{c_m, c_n}$  如式(10)和式(11)所示, 其中  $L_{s_k}^c$  是将交换机  $s_k$  迁移到控制器  $c_n$  所产生的迁移请求代价,  $L_{c_m, c_n}$  是控制器负载变化代价.

$$L_{s_k}^{c_m, c_n} = L_{s_k}^c + L_{c_m, c_n} \quad (10)$$

$$\begin{cases} L_{s_k}^c = f_{\text{PACKET}} \cdot (\text{mind}_{kn}) \cdot \sum_{s_i \in U_c} x_{kn} \\ L_{c_m, c_n} = |\lambda_{s_k} \cdot d_{kn} - \lambda_{s_k} \cdot d_{km}| \end{cases} \quad (11)$$

**定义 5** 负载均衡率. 负载均衡率是对网络中控制器负载分布状况的直观体现, 如式(12)所示. 其中式(12)的后半部分是方差计算, 因此需要借助常数  $\alpha$  进行线性变换, 确保负载均衡率呈现: 数值越大, 网络中控制器负载分布越均衡.  $\bar{\Gamma}$  为网络中  $M$  个控制器的平均负载. 在迁移交换机  $s_k$  之前, 网络  $G$  的负载均衡率为:

$$\Phi = \alpha - \frac{1}{M} \sum_{m=1}^M (\Gamma_{c_m} - \bar{\Gamma})^2 \quad (12)$$

在实施交换机迁移后, 负载均衡率更新为:

$$\Phi^* = \alpha - \frac{1}{M} \sum_{m=1, m \neq n}^M [(\Gamma_{c_m}^* - \bar{\Gamma}^*)^2 + (\Gamma_{c_n}^* - \bar{\Gamma}^*)^2] \quad (13)$$

在式(13)中,  $\Gamma_{c_m}^* = \Gamma_{c_m} - \lambda_{s_k} \cdot d_{km}$  和  $\Gamma_{c_n}^* = \Gamma_{c_n} + \lambda_{s_k} \cdot d_{kn}$  分别是  $\Gamma_{c_m}$  和  $\Gamma_{c_n}$  的更新.  $\bar{\Gamma}^*$  是平均负载的更新.

**定义 6** 迁移效能. 将交换机  $s_k$  从控制器  $c_m$  迁移到控制器  $c_n$  的迁移效能  $\zeta_{s_k, c_n}$  定义为负载均衡率变化与迁移代价的比值,  $\zeta_{s_k, c_n}$  值越高, 迁移完成后网络性能越好.

$$\zeta_{s_k, c_n} = \frac{|\Phi^* - \Phi|}{L_{s_k}^{c_m, c_n}} \quad (14)$$

因此, 交换机迁移效能模型可以描述为:

$$\max_{s_k, c_n} \zeta_{s_k, c_n} \quad (15)$$

$$\text{s. t. } \forall i, j \in V, x_{ij} = \{0, 1\} \quad (16)$$

$$\forall s_k \in S, \sum_{c_m \in C} x_{km} = 1 \quad (17)$$

$$\exists c_m \in C, \Gamma_{c_m} \leq \eta_m \quad (18)$$

式(15)是迁移效能模型的目标函数; 式(16)对网络中所有设备的连接关系进行限定; 式(17)表示每个交换机只能连接一个 master 控制器; 式(18)表示网络中不会出现所有控制器均过载现象.

### 2.2.3 迁移方案设定

**定义 7** 迁移动作. 在网络状态确定的条件下, 迁移动作定义为一系列的三元组  $\langle c_m, s_k, c_n \rangle$ , 其中  $c_m \in C_{OM}$  是迁出控制器 (即过载控制器),  $s_k$  是由  $c_m$  管理的迁移交换机,  $c_n \in C_{IM}$  是迁入控制器. 当迁出控制器  $c_m$  确定后 (依据触发因子  $\varphi_{mn}$ ), 迁移方案的焦点集中于从  $c_m$  管理的交换机集合  $U_{c_m}$  中选取  $s_k$ , 以及从  $C_{IM}$  中选出  $c_n$ .

#### (1) 交换机选取

$$p_{s_k} = 1 - \frac{|\bar{\Gamma} - (\Gamma_{c_m} \cdot |U_{c_m}| - \lambda_k)| \cdot e^{(\max d_{s_k})}}{\Gamma_{c_m} \cdot e^{\sum_{s_i \in U_{c_m}} (\max d_{s_i})}} \quad (19)$$

迁出控制器  $c_m$  选取交换机  $s_k$  基于如下考虑. 首先,  $c_m$  更愿意选择一个具有低迁移代价的交换机来消除过载, 期望  $s_k$  能够在  $\Gamma_{c_m}$  和  $\bar{\Gamma}$  之间产生较小的负载差异. 其次,  $c_m$  从时延角度考虑, 优先迁移距它最远的交换机. 因此, 交换机  $s_k$  选取概率如式(19)所示, 其中  $\Gamma_{c_m} \cdot |U_{c_m}| - \lambda_k$  是当  $s_k$  迁出后  $c_m$  的负载.  $p_{s_k}$  越大, 则交换机  $s_k$  被控制器  $c_m$  从子域中迁出的可能性越高.

#### (2) 迁入控制器选取

当迁移交换机  $s_k$  连接至新控制器时, 首先检测该连接是否会导致新控制器过载, 如果  $s_k$  的加入造成控制器处理容量超出, 则将该控制器从迁入控制器集合  $C_{IM}$  中进行删除. 同时在选择迁入控制器时必须考虑两方面因素: 1) 该控制器是否具有较大的剩余处理容量  $\max(\eta_{c_n} - \Gamma_{c_n} - \lambda_{s_k})$  ( $\eta_{c_n}$  为控制器  $c_n$  的总处理容量,  $\Gamma_{c_n}$  为控制器  $c_n$  的负载,  $\lambda_{s_k}$  为迁移交换机的流速率), 即当迁移交换机连接到该控制器时不会造成该控制器成为新的过载控制器; 2) 该控制器必须具有较高的迁移效能  $\max \zeta_{s_k, c_n}$ , 即迁移交换机连接到该控制器时会带来高迁移效能, 保证负载均衡.

在迁移效能模型指导下, 通过设定并集  $\cup$  联立上述两方面因素, 如式(20)所示, 控制器  $c_n$  被选为迁入控制器.

$$c_n = \arg \max_{c_n \in C_{IM}} \{ \max(\eta_{c_n} - \Gamma_{c_n} - \lambda_{s_k}) \cup \zeta_{s_k, c_n} \} \quad (20)$$

## 3 算法设计

在迁移效能模型的基础上, 对迁移方案中三元组  $\langle c_m, s_k, c_n \rangle$  执行迁移操作. 由于交换机迁移是 NP-Hard<sup>[18]</sup> 问题, 因此 EOSDM 基于贪婪决策<sup>[19]</sup> 进行实施. EOSDM 实施包含两个步骤: EOSDM-I 完成控制器负载检测,

EOSDM-II 实现交换机协调迁移。

### 3.1 EOSDM-I: 控制器负载检测

控制器和子域内交换机进行信息交互,并计算控制器  $c_m$  聚合负载  $\Gamma_{c_m}$  和负载差异  $q_{c_m,c_n}$  (行 3),得到负载差异矩阵  $\mathbf{Q}_{M \times M}$  (行 6). 然后在  $\mathbf{Q}_{M \times M}$  中对于不同控制器计算触发因子  $\varphi_{mn}$ ,与负载差异阈值  $\sigma$  进行比较 (行 9),如果  $\varphi_{mn}$  超过该阈值,则控制器  $c_m$  和  $c_n$  将会被分别添加到迁出控制器集合  $C_{OM}$  和迁入控制器集合  $C_{IM}$  中 (行 10). 该过程一直执行,直至  $\mathbf{Q}_{M \times M}$  中元素全部轮询完毕 (行 15).

#### 算法 1 EOSDM-I: 控制器负载检测

Step 1 EOSDM-I  
 输入: 网络拓扑  $G = (V, E)$   
 输出: 迁出控制器集合  $C_{OM}$   
       迁入控制器集合  $C_{IM}$

- 1 while ( $C \neq \varnothing$ )
- 2 计算  $\Gamma_{data}^{c_m}, \Gamma_{route}^{c_m}, \Gamma_{state}^{c_m}$
- 3 得到  $\Gamma_{c_m}$  和  $q_{c_m,c_n}$
- 4  $C = C - \{c_m\}$
- 5 endwhile
- 6 构建矩阵  $\mathbf{Q}_{M \times M}$
- 7 while 0.7 ( $\mathbf{Q}_{M \times M} \neq \varnothing$ )
- 8 从  $\mathbf{Q}_{M \times M}$  选择  $q_{c_m,c_n}$  并计算  $\varphi_{mn}$
- 9 if ( $\varphi_{mn} > \sigma$ )
- 10 增加  $c_m$  到  $C_{OM}, c_n$  到  $C_{IM}$
- 11 else  $C_{OM} = \varnothing$  和  $C_{IM} = \varnothing$
- 12 endif
- 13  $\mathbf{Q}_{M \times M} = \mathbf{Q}_{M \times M} - \{q_{c_m,c_n}, q_{c_n,c_m}\}$
- 14 endwhile
- 15 输出  $C_{OM}$  和  $C_{IM}$

### 3.2 EOSDM-II: 交换机协调迁移

基于 EOSDM-I 得到的控制器集合,交换机迁移方案被规划为一系列的迁移动作. 首先,计算迁移代价和负载均衡率,进而得到迁移效能 (行 2 ~ 行 4). 依据贪婪决策,在  $C_{OM}$  中,对于每一个控制器生成概率选择器 (行 6),迁移具有最大选取概率的交换机  $s_k$  (行 7); 在  $C_{IM}$  中,基于迁移效能模型,选择具有  $\max(\eta_{c_n} - \Gamma_{c_n} - \lambda_{s_k}) \cup \max \zeta_{s_k,c_n}$  的控制器作为迁入控制器 (行 10). 添加  $\langle c_m, s_k, c_n \rangle$  到迁移行为集合  $P$  (行 14),执行迁移动作,并更新控制器状态 (行 16 ~ 行 17),直至  $\varphi_{mn} < \sigma$ ,最终输出新的拓扑连接关系 (行 19).

#### 算法 2 EOSDM-II 交换机协调迁移

Step2 EOSDM-II  
 输入: 网络拓扑  $G = (V, E)$   
       迁出控制器集合  $C_{OM}$   
       迁入控制器集合  $C_{IM}$

输出: 控制器管理的交换机集合  $U$

      连接关系  $x_{ij}^*$

- 1 初始化迁移动作集合  $P = \{\}$
- 2 计算  $L_{s_k}^{c_m}$  和  $L_{c_m,c_n}^{c_m}$ , 得到  $L_{s_k}^{c_m,c_n}$
- 3 基于控制器负载  $\Gamma_{c_m}$ , 计算  $\Phi$  和  $\Phi^*$
- 4 得到迁移效能  $\zeta_{s_k,c_n}$
- 5 while ( $C_{OM} \neq \varnothing$ )
- 6 计算由  $c_m$  管理的交换机  $p_{s_k}$
- 7 选择具有  $\max p_{s_k}$  的交换机作为迁移交换机
- 8  $C_{OM} = C_{OM} - c_m$
- 9 while ( $C_{IM} \neq \varnothing$ )
- 10 选择  $\max(\eta_{c_n} - \Gamma_{c_n} - \lambda_{s_k}) \cup \max \zeta_{s_k,c_n}$  的控制器
- 11  $C_{IM} = C_{IM} - c_n$
- 12 endwhile
- 13 endwhile
- 14 添加  $\langle c_m, s_k, c_n \rangle$  到  $P$
- 15 if  $\exists \varphi_{mn} > \sigma$
- 16 迁移  $s_k$  从  $c_m$  到  $c_n$
- 17 Update states of  $c_m$  and  $c_n$
- 18 endif
- 19 Output  $U$  and  $x_{ij}^*$  in  $G$

## 4 性能评估

为了说明本文所提出的 EOSDM 策略性能,本节仿真实验层面将 EOSDM 策略与静态连接 (Static Connection, SC) 策略,文献 [12] 中距离就近迁移 (Distance Closest Migration, DCM) 策略和文献 [15] 中分布式逐跳 (Distributed Hopping Algorithm, DHA) 策略进行比较,验证 EOSDM 策略的有效性和可行性. 各种策略性能描述如表 4 所示.

表 4 策略比较

策略	性能描述
SC	交换机和控制器固定连接
DCM	将交换机迁移至距离最近控制器
DHA	选取剩余容量最大的控制器为迁移对象
EOSDM	基于效能优化的交换机动态迁移策略

### 4.1 仿真环境搭建

#### (1) 实验平台和物理设备介绍

本文选用 OpenDaylight<sup>[20]</sup> 作为实验控制器,同时选取 Mininet<sup>[21]</sup> 作为轻量级测试平台. 实验环境包含 7 台具有相同实验配置的服务器 (Intel Core i7 3.5GHz 4GB RAM). 6 台服务器运行 OpenDaylight 控制器,编号分别为 1 ~ 6, 剩余 1 台服务器运行 Mininet. 所有服务器均通过 H3C S1016 交换机进行连接.

#### (2) 拓扑选择

为了使实验更具说服力,所有实验均基于权威网络拓扑. 首先在 Internet2 OS3E<sup>[22]</sup> 验证 EOSDM 策略的有效性,然后在 Topology Zoo<sup>[23]</sup> 中选取部分拓扑重新实

验,验证 EOSDM 策略的拓扑适应性.依据文献[9]的通信开销最小原则,在拓扑中预先优化控制器放置.

### (3) 实验参数设定

类比文献[13]所示的流量特征,使用流量生成器来模拟网络中流量分布状况,平均流产生速率为 300KB/s. 轮循一个交换机的平均速率  $v_{sw}$  等于 25KB/s,  $f_{packet}$  等于 20Byte,  $\mu_{syn}$  等于 10Byte. 经过多次仿真统计,设定负载差异阈值  $\sigma = 0.5$  即能满足实验需求. 对负载均衡率进行归一化处理,设定  $\alpha = 1$ .

## 4.2 仿真结果分析

**实验 1 控制器响应时间** 本实验基于 OS3E 网络拓扑,每次仿真时间 120s,初始时刻所有控制器负载基本相同,在第 20s,通过调整 PACKET\_IN 发送速率,使得 3 号控制器发生过载. 实验结果如图 2 所示,图中虚直线上数字标记表示 DCM, DHA 和 EOSDM 三种策略的交换机迁移执行时间,分别为 32s, 39s 和 23s.

可以看出,当控制器发生过载时,由于 SC 中交换机和控制器静态连接,控制器平均响应时间达到了 75ms. DCM 将交换机迁移至最近的控制器,邻近迁移很容易产生新的负载不均衡问题,导致迁移期间的控制器平均响应时间最长(64ms),并且在迁移完成后,和初始状态相比,DCM 的平均响应时间仍处于较高水平. DHA 迁移交换机至剩余处理容量最大的控制器,虽然增加了迁移执行时间,但在迁移期间,控制器平均响应时间增加较少(59ms),待迁移完成之后,平均响应时间略高于初始状态. EOSDM 合理的迁移策略设定可以有效缩短迁移执行时间,控制器平均响应时间也处于 4 种策略中最低值(51ms),至少降低了 15.7%.

图 3 描述了控制器响应时间的累积分布函数,图中  $\alpha$  表示过载控制器数量. 与其他三种策略相比,当网络中有多个过载控制器时,EOSDM 仍然具有明显的性能优势.

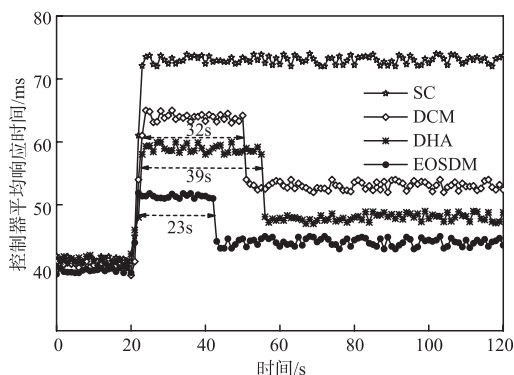


图2 控制器平均响应时间

**实验 2 迁移代价与控制器吞吐量** 同样基于 OS3E 网络拓扑,本实验对比了当 3 号控制器发生过载时不同迁移策略所产生的迁移代价,以及迁移完成后

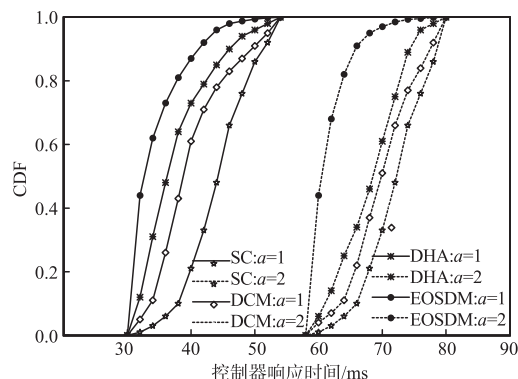


图3 控制器响应时间的CDF

控制器吞吐量状况. 实验结果分别如图 4 和图 5 所示. 结合图 4 和图 5 实验结果,在迁移代价方面,DHA 最高,EOSDM 略高于 DCM;在控制器吞吐量方面,EOSDM 最高,DHA 次之,DCM 最低. 尽管 DCM 的就近迁移策略可以有效减小迁移代价,但局部选择很容易造成邻居控制器负载过高. DHA 从全局角度出发寻找剩余处理容量最大的控制器,提高了控制器吞吐量. 然而单方面性能增长所引发的后果是在网络中产生高迁移代价. EOSDM 通过建立迁移效能模型,在负载均衡率和迁移代价之间进行有效权衡,既保证了迁移完成后 EOSDM 具有三种策略中最高控制器吞吐量,同时也使得迁移代价较低.

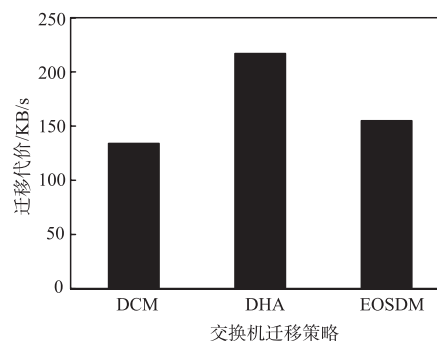


图4 不同迁移策略的迁移代价

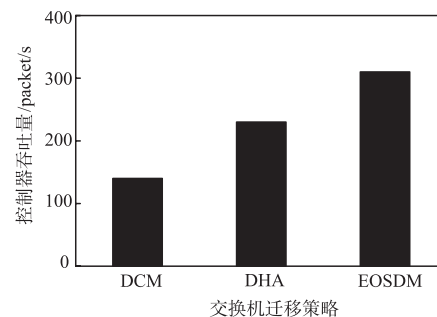


图5 不同迁移策略的控制器吞吐量

**实验 3 负载均衡率** 本实验在 Topology zoo 中选取多个网络拓扑,对比 4 种策略的负载均衡率随过载控制器数量变化情况. 拓扑信息如表 5 所示,网络拓扑

规模逐步扩大.

表 5 拓扑信息

拓扑	Abilene	BT	Germany50	Columbus
节点	11	36	50	70
链路	14	76	88	85

在实验 3 扩大控制器部署,将网络中控制器部署总数提升至 12 个,同时为了设定过载控制器数量不超过控制器总数的 80% (10 个控制器),仿真结果如图 6 所示,就整体状况而言,在不同规模的网络拓扑中,所有策略的负载均衡率都呈现下降趋势,这是因为网络

中过载控制器数量增加会使得负载均衡难度变大.随着过载控制器数量上升,SC 负载均衡率迅速下降.当邻居控制器未发生过载时,DCM 通过迁移交换机至距离最近的控制器,仍能保持较高负载均衡率,一旦邻居控制器也处于过载状态,该策略失效,其负载均衡率明显下降. DHA 考虑全局控制器容量,负载均衡率下降趋势相对平稳. EOSDM 通过效能感知迁移算法的设定也确保交换机高效迁移,其负载均衡率始终高于其他三种策略.特别地,对实验结果分析可以得出,当网络中过载控制器增多至一定程度(超过 8 个控制器),4 种策略的负载均衡率均较低水平.

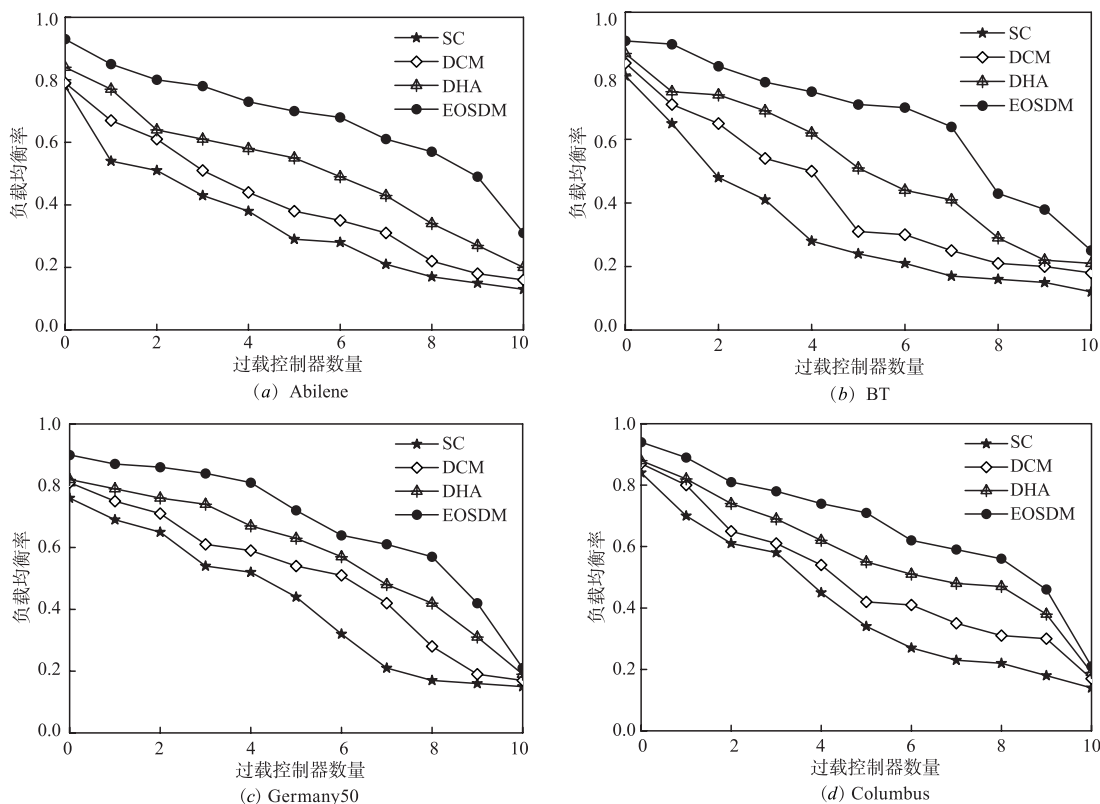


图 6 不同规模网络拓扑中负载均衡率

## 5 结论

本文针对交换机迁移过程中迁移效率低下和高迁移代价问题,提出基于效能优化的交换机动态迁移(EOSDM)策略.区别于传统解决思路,EOSDM 策略通过建立迁移效能模型指导迁移策略设定,并设计相应的算法进行实施.实验结果表明,和现有方案相比,EOSDM 策略具有较好的负载均衡优势.下一步研究工作将围绕故障场景下失效转移和路由规划两方面内容进行展开.

### 参考文献

[1] Dave T. OpenFlow: Enabling innovation in campus net-

works[J]. ACM Sigcomm Computer Communication Review, 2008, 38(2): 69-74.

[2] Hu T, Yi P, Zhang J, Lan J. Reliable and load balance-aware multi-controller deployment in SDN[J]. China Communications, 2018, 15(11): 184-198.

[3] Hu T, Guo Z, Yi P, Baker T, Lan J. Multi-controller-based software-defined networking: A survey[J]. IEEE Access, 2018, 06: 15980-15996.

[4] Koponen T, Casado M, Gude N, et al. Onix: a distributed control platform for large-scale production networks[A]. Proceedings of Usenix Conference on Operating Systems Design and Implementation [C]. USENIX Association, 2010. 351-364.

- [5] Hu T, Yi P, Guo Z, Lan J, Zhang J. Bidirectional matching strategy for multi-controller deployment in distributed software defined networking[J]. IEEE Access, 2018, 06: 14946 – 14953.
- [6] Guo Z, Xu Y, Cello M, et al. JumpFlow: Reducing flow table usage in software-defined networks[J]. Computer Networks, 2015, 92: 300 – 315.
- [7] Sherwood R, Mckeown N. The controller placement problem[A]. Proceedings of the Workshop on Hot Topics in Software Defined Networks[C]. ACM, 2012. 7 – 12.
- [8] Hock D, Hartmann M, Gebert S. Pareto-optimal resilient controller placement in SDN-based core networks[A]. Proceedings of Teletraffic Congress [C]. IEEE, 2013. 1 – 9.
- [9] Bari M F, Roy A R, Chowdhury S R, et al. Dynamic controller provisioning in software defined networks[A]. Proceedings of International Conference on Network and Service Management[C]. IEEE, 2013. 18 – 25.
- [10] Yonghong F U, Jun B I, et al. A dormant multi-controller model for software defined networking[J]. China Communications, 2014, 11(3): 45 – 55.
- [11] OpenFlow switch specification version 1.4.0 [OL]. <https://www.opennetworking.org/>, 2018.
- [12] Dixit AA, Hao F, Mukherjee S, et al. ElastiCon: an elastic distributed sdn controller[A]. Proceedings of Acm Sigcomm Workshop on Hot Topics in Software Defined Networking[C]. IEEE, 2017. 17 – 28.
- [13] Liang C, Kawashima R, Matsuo H. Scalable and crash-tolerant load balancing based on switch migration for multiple open flow controllers[A]. Proceedings of Second International Symposium on Computing and Networking [C]. IEEE, 2014. 171 – 177.
- [14] Hu T, Yi P, Zhang J. A distributed decision mechanism for controller load balancing based on switch migration in SDN[J]. China Communications, 2018, 15(10): 129 – 142.
- [15] Hu T, Guo Z, Zhang J, Lan J. Adaptive slave controller assignment for fault-tolerant control plane in software-defined networking[A]. Proceedings of IEEE International Conference on Communications (ICC) [C]. Kansas City, 2018. 1 – 6.
- [16] 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(11): 95 – 109.
- [17] Hong chang, et al. A game-theoretic approach to elastic control in software-defined networking[J]. China Communications, 2016, 13(5): 103 – 109.
- [18] Qin X L, Zhang W B. Enabling elasticity of key-value stores in the cloud using cost-aware live data migration[J]. Journal of Software, 2013, 24(6): 1403 – 1417.
- [19] Sahoo K S, et al. Optimal controller selection in software defined network using a greedy-SA algorithm[A]. Proceedings of International Conference on Computing for Sustainable Global Development[C]. IEEE, 2016. 1 – 6.
- [20] OpenDaylight[OL]. <http://www.opendaylight.org/>, 2018.
- [21] Mininet [OL]. <http://mininet.org/>, 2018.
- [22] Internet2 Open Science, Scholarship and Services Exchange [OL]. <http://www.internet2.edu/network/ose/>, 2018.
- [23] 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.

#### 作者简介



**姚 蓝 女**. 1982 年生于河南信阳. 现为国家数字交换系统工程技术研究中心博士研究生. 研究方向为软件定义网络.  
E-mail: 8661312@qq.com



**胡 涛 (通信作者) 男**. 1993 年 8 月生于陕西武功. 现为国家数字交换系统工程技术研究中心博士研究生. 主要研究方向为新型网络体系结构、网络安全.  
E-mail: hutaondsc@163.com

**伊 鹏 男**. 1977 年生于湖北黄冈. 现为国家数字交换系统工程技术研究中心研究员. 研究方向为宽带信息网络、网络安全.

**胡宇翔 男**. 1982 年生于河南周口. 现为国家数字交换系统工程技术研究中心副研究员. 研究方向为新型网络.

**兰巨龙 男**. 1962 年出生于河北张北. 现为国家数字交换系统工程技术研究中心教授. 主要研究方向为新型网络.

**李子勇 男**. 1995 年生于安徽蚌埠. 现有国家数字交换系统工程技术研究中心硕士生. 研究方向为新型网络.