

SDN 中基于过程优化的交换机竞争迁移算法

胡涛¹, 张建辉¹, 孔维功¹, 杨森¹, 曹路佳²

(1. 国家数字交换系统工程技术研究中心, 河南 郑州 450002; 2. 解放军信息工程大学网络空间安全学院, 河南 郑州 450001)

摘要: 针对分布式软件定义网络中交换机迁移对象选取僵化和迁移冲突问题, 提出一种基于过程优化的交换机竞争迁移 (SCM, switch competing migration) 算法。将原有的交换机迁移方案优化为迁移对象选取和交换机动态部署 2 个过程。首先, 综合权衡网络中时延、流量和故障代价, 灵活地选取迁移对象; 然后, 根据设定的存活期、迁移期和控制器负载动态门限值, 交换机以竞争迭代的方式动态部署到目标控制器。仿真结果表明, 与现有的算法相比, 迁移对象的选取更加合理, 避免交换机迁移冲突, 控制器负载均衡率平均提升了 25.6%。

关键词: 软件定义网络; 多控制器; 负载均衡; 过程优化

中图分类号: TP393

文献标识码: A

Switch competing migration algorithm based on process optimization in SDN

HU Tao¹, ZHANG Jian-hui¹, KONG Wei-gong¹, YANG Sen¹, CAO Lu-jia²

(1. National Digital Switching System Engineering R&D Center, Zhengzhou 450002, China;

2. School of Cyberspace Security, The PLA Information Engineering University, Zhengzhou 450001, China)

Abstract: In the distributed software defined network, switch selects rigid migration object and has coordinating migration problems. The prime switch migration scheme into two processes included selecting migration object and deploying switches dynamically was optimized, a switch competing migration algorithm was proposed based on process optimization. Firstly, it synthesized delay, traffic and failure cost, and selected migration object. Then, according to survival time, migration time and dynamic load threshold of controller, switches were migrated into object controller through iterative competition. Results show that compared with the existed migration algorithms the proposed algorithm selects more reasonable migration objects, avoiding migration conflict between switches, and controller load balancing rate has been promoted to 25.6% averagely.

Key words: software defined network, multiple controller, load balancing, process optimization

1 引言

基于 OpenFlow^[1] 的软件定义网络 (SDN, software defined network) 是近些年来新兴的网络体系架构, 具有集中式管理, 易于操作和可编程的特点, 受到人们的广泛关注。随着网络规模的不断扩

大和流量需求的迅速增长, 单个控制器已经无法满足现有的网络需求, 因此, 研究者相继提出具有多控制器的分布式 SDN 架构^[2] (如 HyperFlow^[3]、Kandoo^[4]、Onix^[5])。虽然多控制器的部署有效地改善了网络可靠性和可扩展性, 但其负载均衡性能受到极大挑战, 如何在分布式软件定义网络中均衡控

收稿日期: 2016-11-28; 修回日期: 2017-03-31

基金项目: 国家自然科学基金创新研究群体基金资助项目 (No.61521003); 国家科技支撑计划基金资助项目 (No.2014BAH30B01); 国家自然科学基金资助项目 (No.61372121, No.61572519); 国家高技术研究发展计划 (“863” 计划) 基金资助项目 (No.2015AA016102, No.2013AA013505); 河南省科技攻关计划基金资助项目 (No.162102210034)

Foundation Items: The National Natural Science Foundation for Creative Research Groups of China (No.61521003), The National Key Technology R&D Program of China (No.2014BAH30B01), The National Natural Science Foundation of China (No.61372121, No.61572519), The National High Technology Research and Development Program of China (863 Program) (No.2015AA016102, No.2013AA013505), The Key Scientific and Technological Project of Henan Province (No.162102210034)

制器负载成为近些年研究的热点问题。

现有的控制器负载均衡方案一般采取迁移交换机的方式，通过迁出过载控制器域内部分交换机到轻载控制器，完成控制器负载的动态转移。目前，交换机迁移算法主要从以下 2 个方面进行研究。1) 考虑网络代价。Heller 等^[6]首次提出建立交换机迁移模型，考虑网络中时延因素，对控制器负载进行动态调整。但该模型只进行了理论性分析，对整个迁移过程未做细节上的描述。张联镇等^[7]从设备角度对时延度量进一步细化，提出在交换机迁移过程中考虑交换机到控制器时延，控制器间时延和交换机间时延。Hock 等^[8]综合了时延和流量度量，在交换机迁移过程中进行多代价权衡，但对于多交换机迁移冲突问题未做讨论。Eugen 等^[9]提出了多控制器部署优化方案，重点考虑控制器负载均衡和控制器间时延，权衡了多个互相矛盾的优化目标，但是对于迁移交换机的数量和位置未做要求。Adlen 等^[10]提出使用博弈论解决 SDN 控制器最优部署，在时延、通信开销和控制器负载均衡这 3 个目标进行博弈考虑，找到均衡度量，优化控制器部署数量和位置。2) 考虑迁移策略。文献^[11]提出利用率最低迁移策略，在交换机动态部署过程中，通过共享全局网络拓扑信息，将具有最低容量利用率的控制器的交换机迁移对象。Cheng 等^[12]研究了一种可扩展机制，通过迁移交换机到轻载控制器来均衡控制平面，并提出分布式交换机迁移决策，在有限的控制资源内最大化网络利用率，服务更多的请求。文献^[13]在谱聚类的基础上，通过为 *k*-means 方法增加均衡部署的目标函数，提出在时延和容量限制下 SDN 网络多控制器负载均衡算法，虽然能够有效地降低交换机向控制器查询的平均时延，但在大规模网络拓扑中，算法的运算复杂度急剧上升。

综合国内外研究现状，现有的交换机迁移算法虽然能够动态地调整控制器负载，但在迁移对象的选取方面考虑因素较为单一，迁移策略僵化，对于不同条件的拓扑适应能力差。大部分迁移方案只注重单个迁移目标，未曾考虑多交换机迁移协调和状态一致性问题，从而导致交换机迁移完成后，网络中某一度量突出，控制器整体性能并未得到改善。

针对上述问题，本文对原有的交换机迁移过程实施优化，分为迁移对象选取和交换机动态部署 2 个阶段，提出 SCM 算法。在迁移对象选取阶段，设计了 SCM-1 算法。通过周期性收集网络中拓扑

信息，包括跳数、交换机请求速率和控制器处理容量，确定迁出域和待迁移交换机；对于迁入域的选择，应用文献^[14]中遗传算法的思想，综合时延、控制流量、故障代价等参数，设定优化目标，进行多代价近似最优求解，确定迁入域。在交换机动态部署阶段，设计了 SCM-2 算法。对迁移对象设定迁移参数，包括存活期、迁移期和控制器负载动态门限值，在算法的每一次迭代中动态变化存活期，并和迁移期进行对比，通过竞争排序的方式优先迁移符合迁移条件的交换机，逐步缩小交换机竞争范围，不断迭代直至所有待迁移交换机成功连接到目标控制器。

2 模型构建

2.1 问题描述

分布式 SDN 场景如图 1 所示，整个网络被划分为 3 个子域，分别是子域 1、子域 2 和子域 3，分别由控制器 C1、C2 和 C3 进行控制。由于网络流量的不均衡分布导致控制器 C1 发生过载，现有的解决思路是通过迁移过载域内交换机来均衡控制器负载。

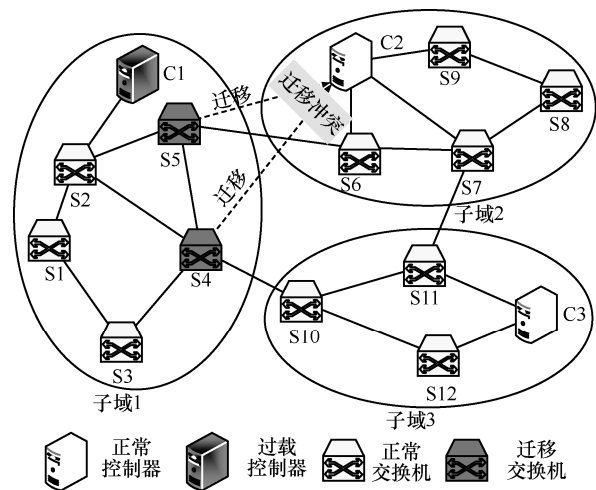


图 1 分布式 SDN 中交换机迁移

在图 1 的 3 个子域中，子域 1 作为迁出域，包含 5 个交换机和 1 个过载控制器，同时已经确定 S4 和 S5 作为迁移交换机。子域 2 和子域 3 分别包含 4 个和 3 个交换机。在迁入域选取方面，现有的方案拘泥于迁移距离最短策略，忽略了其他网络代价对迁移过程的影响（例如，在控制器负载方面，C2 的负载高于 C3），僵化地选取控制器 C2 所在子域 2 作为迁入域（S4 和 S5 距离 C2 最近），同时，将它

们的主控制器由 C1 转换为 C2。

尽管多交换机并行迁移可以快速地降低过载控制器的负载,但同时也引发了一个新的问题——交换机迁移冲突。当 S4 和 S5 同时选择 C2 作为迁入控制器时,考虑到处理容量限制和负载均衡,C2 只能接收一个迁移交换机,此时,S4 和 S5 形成竞争迁移关系,当它们同时迁入 C2,则在 2 个交换机之间产生迁移冲突。如果 C2 在迁移冲突背景下选择继续接受 S4 和 S5,将会导致 C2 成为新的过载控制器,需要再次迁移 S4 或 S5,造成二次迁移,加重了网络负担。

因此,本文所要解决的问题是:1) 在 SDN 多控制器网络中,根据当前网络状况,优化迁移对象的选取;2) 设定交换机动态部署方案,协调迁移过程,避免迁移冲突,实现控制器负载的均衡分布。

2.2 网络模型

根据图论的相关知识,对 SDN 多控制器实施建模。整个网络拓扑用无向图 $G=(V,E)$ 表示,其中, V 表示网络中节点集合, E 表示节点间链路集合。在网络中共包含 M 个控制器和 N 个交换机,有 $|V|=M+N$ 。整个网络被划分为多个子域即 $G=\{G_1,G_2,\dots,G_M\}$,且任意 2 个子域的交集为空。SDN 中控制器集合为 $C=\{c_1,c_2,\dots,c_M\}$,交换机集合为 $S=\{s_1,s_2,\dots,s_N\}$ 。 d_{ij} 表示设备 i 和 j 之间的跳数。交换机的流请求速率为 λ_i ,控制器的处理容量为 η_j 。交换机和控制器之间的连接关系为 y_{ij} ,如式(1)所示。网络中所有设备的连接关系构成连接矩阵 $\Omega=[y_{ij}]_{N \times M}$ 。

$$y_{ij} = \begin{cases} 1, & \text{第 } i \text{ 个交换机连接到第 } j \text{ 个控制器} \\ 0, & \text{其他} \end{cases} \quad (1)$$

为解决迁移对象选取僵化问题,基于 OpenFlow 机制对网络中相关参数进行定义。

定义 1 流请求传输时延。交换机 s_i 到控制器 c_j 的流请求传输时延定义为 D_{ij}^{req} ,如式(2)所示。由于交换机到控制器的请求信息可能途径域中其他设备,设交换机请求矩阵为 $X=[x_{ij}]_{N \times M}$,其中, x_{ij} 表示交换机 s_i 与控制器 c_j 进行通信时产生的平均路径请求次数。

$$D_{ij}^{\text{req}} = \sum_{i=1}^N \sum_{j=1}^M y_{ij} d_{ij} x_{ij} \lambda_i \quad (2)$$

定义 2 控制器间同步时延。网络中多控制器在进行状态同步时,假设控制器同步周期为 T ,则控制器 c_p 到控制器 c_q 的同步时延定义为 D_{pq}^{syn} ,计算式为

$$D_{pq}^{\text{syn}} = \sum_{p=1}^M \sum_{q=1}^M d_{pq} T \quad (3)$$

定义 3 控制流量。假设每一个交换机请求需要被发送给至少一个控制器,则控制器 c_j 的控制流量为 L_j ,计算式为

$$L_j = \sum_{i=1}^M \sum_{i=1}^N d_{ij} y_{ij} \lambda_i \quad (4)$$

定义 4 控制器故障代价。当控制器未连接任何交换机时,则考虑该控制器处于故障状态^[15]。假设网络中出现链路失效的可能性都为 l ,设控制器 c_j 的故障代价为 B_j ,计算式为

$$B_j = \sum_{i=1}^N \prod_{j=1}^M (1-l^{d_{ij}}) y_{ij} \quad (5)$$

综合上述网络开销和代价,设交换机迁入域的优化目标为 $\min E$,目标函数为

$$E = \mu_1 D_{ij}^{\text{req}} + \mu_2 D_{pq}^{\text{syn}} + \mu_3 L_j + \mu_4 B_j \quad (6)$$

约束条件为

$$\forall i, j \ y_{ij} \in \{0,1\} \quad (7)$$

$$\sum_{i=1}^N y_{ij} = 1 \quad (8)$$

$$\sum_{i=1}^N \sum_{j=1}^M x_{ij} \lambda_i \leq \eta_j \quad (9)$$

其中, μ_1 、 μ_2 、 μ_3 和 μ_4 分别为流请求传输时延、控制器同步时延、控制流量和控制器故障代价的对应权值。式(7)表示每一个交换机都能连接到控制器,确保网络运行的可靠性。式(8)表示每一个交换机只能选择一个控制器作为主控制器。式(9)保证子域内交换机的请求速率之和不超过控制器的处理容量限制。

3 算法设计

本节介绍 SCM 算法 (SCM-1 和 SCM-2)。算法分为 2 个阶段,在阶段 1,SCM-1 完成迁移对象选取;在阶段 2,SCM-2 完成选取后交换机的动态部署。

3.1 SCM-1

在阶段 1, SCM-1 依据当前网络状况, 灵活地选取待迁移交换机和迁入、迁出域。待迁移交换机和迁出域可以根据收集的网络信息直接确定, 选取控制器处理容量最小子域作为迁出域 (第 2) 行)。在迁出域中选取请求速率大于速率平均值的交换机构成待迁移交换机集合 (第 3) 行), 初步避免无原则的过量交换机迁移问题。在迁入域的选取过程中, 必须综合网络中时延、流量和控制器故障代价等参数, 进行多代价优化处理 (第 6) 行)。基于遗传算法, 通过对现有的解空间实施选择、交叉、变异等遗传操作, 不断迭代, 逼近优化目标 $\min E$ (第 14) 行), 最终得到可接受的交换机迁入域。SCM-1 算法伪代码如算法 1 所示。

算法 1 SCM-1

输入 SDN 网络: $G = \{G_1, G_2, \dots, G_M\}$

交换机集合: S

交换机请求速率: λ_i

控制器处理容量: η_j

优化目标: $\min E$

输出 迁出域: G_h

迁入域: G_l

待迁移交换机集合: S_M

- 1) 控制器处理容量排序, 找出其中最小值 $\eta_h = \min\{\eta_1, \eta_2, \eta_3, \dots, \eta_M\}$
- 2) 选择第 h 个子域作为迁出域 G_h
- 3) 在迁出域 G_h 中选取 $\lambda_i > \bar{\lambda}$ 的交换机构成待迁移交换机集合 S_M
- 4) G_h 的相邻子域集合 $\Gamma_h = \{G_{h-1}, G_{h-2}, G_{h-3}, \dots, G_{h-r}\} (r < M)$
- 5) while ($\Gamma_h \neq \emptyset$)
- 6) E 中权值 μ 进行 0, 1 编码, 生成初始解空间
- 7) 计算 G_{h-r} 的适应度 E_r , 得到 E_{avg}
- 8) Select $G_{h-m} (E_m \leq E_{\text{avg}}), G_{h-n} (E_n > E_{\text{avg}}), \forall m, n \in \{1, 2, 3, \dots\}$
- 9) Crossover 两点交叉, 染色体结构互换
- 10) $G'_{h-m} \leftarrow G_{h-m}, G'_{h-n} \leftarrow G_{h-n}$
- 11) Mutation 随机选取逆转点, 染色体逆向变异
- 12) $G^*_{h-m} \leftarrow G'_{h-m}, G^*_{h-n} \leftarrow G'_{h-n}$
- 13) 计算新个体的适应度 $\{E_1, E_2, \dots\}$
- 14) if $E_l = \min\{E_1, E_2, \dots\}$

15) then 解码得到迁入域 G_l

16) end if

17) end while

由于迁出域 $G(h)$ 的相邻子域个数有限, 因此, SCM-1 的复杂度与遗传算法有关, 最小为 $O(r)$ 。在执行遗传操作前, 首先, 确定迁出域位置, 然后, 获得迁出域的相邻子域数量为 r 个, 在每轮数据处理时, 对邻域集合 Γ_h 进行遗传操作, 只需进行简单的乘、加线性运算, 无需大量复杂运算, 最终解码得到迁入域, 具有实时性。

3.2 SCM-2

在阶段 2, 算法 SCM-2 通过协调交换机迁移过程, 实现交换机的动态部署。集合 S_M 中所有待迁移交换机构成竞争迁移关系, 为了避免多交换机迁移引发的迁移冲突问题, 在阶段 2 设定相应的迁移参数 (存活期、迁移期和控制器负载动态门限值), 采用周期迭代和竞争排序的方式选取交换机迁移至 G_l 域。SCM-2 的核心思想是: 在算法的每一次迭代中动态变化交换机存活期, 并和交换机迁移期进行对比, 在待迁移交换机集合中依据迁移条件对交换机实施竞争排序, 优先迁移满足迁移条件且具有 $\max(\lambda_i d_{ij})$ 的交换机, 逐步缩小 S_M 范围, 最终实现多交换机之间的协调迁移, 避免交换机迁移冲突。

首先, 对 SCM-2 中出现的存活期、迁移期和控制器负载动态门限值进行定义与计算。

定义 5 交换机存活期 $Y(s_i)$ 是指交换机处于待迁移状态的时间, 与交换机到控制器的距离 d_{ij} 和流请求传输时延 D_{ij}^{req} 有关, 计算式为

$$Y(s_i) = d_{ij} D_{ij}^{\text{req}} \quad (10)$$

定义 6 交换机迁移期 $T(s_i)$ 是指交换机 s_i 完成迁移任务所需时间, 通过对流请求传输时延和控制器同步时延进行求和可以得到, 计算式为

$$T(s_i) = D_{ij}^{\text{req}} + D_{hi}^{\text{syn}} \quad (11)$$

定义 7 控制器负载动态门限值 L_j^{thres} , 用于判断控制器 c_j 是否处于过载状态的警戒阈值, 如式 (12) 所示。通过设置 L_j^{thres} 可以防止迁移造成子域交换机数量过多或过少。其中, L_{max} 、 L_{min} 和 L_{avg} 分别为控制器的最大、最小和平均控制流量负载。

$$L_j^{\text{thres}} = \min\left[\frac{1}{2}(L_{\text{max}} - L_{\text{min}}) \frac{L_j}{L_{\text{avg}}}, L_{\text{max}}\right] \quad (12)$$

因此, 交换机成功迁移需要满足以下 2 个基本条件。

- 1) 待迁移交换机的存活期大于迁移期。
- 2) 迁移完成后, 控制器负载不能超过门限值 L_j^{thres} 。

SCM-2 算法的流程是: 在 SCM-1 算法基础上, 计算集合 S_M 中待迁移交换机的存活期和迁移期 (第 2) 行)。初始时刻, 所有交换机的存活期都大于迁移期 (第 3) 行), 在待迁移交换机集合 S_M 中选取具有 $\max(\lambda_i d_{ij})$ 的交换机实施首次迁移。在接下来的迭代过程中, 每成功迁移一个交换机, 集合 S_M 中待迁移交换机数量减小, 同时剩余交换机的存活期都要减去相对周期 $\left[1 + \ln\left(\frac{m-i}{m}\right)\right]T$ (i 为成功迁移的交换机个数), 并判断是否满足迁移条件 (第 9) 行), 优先选取符合迁移条件且具有 $\max(\lambda_i d_{ij})$ 的交换机, 若待迁移集合为空或迁入域控制器负载超出控制器负载动态门限值 (第 12) 行), 则算法跳出, 最终输出新的多域网络拓扑, 实现控制器负载的均衡分布。SCM-2 算法伪代码如算法 2 所示。

算法 2 SCM-2

输入 迁出域: G_h

迁入域: G_l

待迁移交换机集合: S_M

输出 新的 SDN 多域网络拓扑

- 1) S_M 中待迁移交换机编号为 $S_M = \{s_1^M, s_2^M, \dots, s_m^M, \dots\}$, $|S_M| = K$
- 2) 计算 S_M 中所有待迁移交换机存活期 $Y(s_m^M)$ 和迁移期 $T(s_m^M)$
- 3) Initial $\forall s_m^M \in S_M, Y(s_m^M) \geq T(s_m^M)$
- 4) $\exists s_m^M, \lambda_m d_{ml} = \max\{\lambda_1 d_{1l}, \lambda_2 d_{2l}, \dots\}$
- 5) 交换机 s_m^M 进入迁移状态
- 6) $G_h = G_h \setminus \{s_m^M\}$, $G_l = G_l \cup \{s_m^M\}$, $S_M = S_M \setminus \{s_m^M\}$
- 7) $L_l^* = L_l + d_{ml} \gamma_{ml} \lambda_m$
- 8) $Y(s_n^M)^* = Y(s_n^M) - T\left[1 + \ln\left(\frac{m-i}{m}\right)\right]$
- 9) if $\exists s_n^M \in S^M, Y(s_n^M)^* < T(s_n^M)$
- 10) 若交换机 s_n^M 不满足迁移条件, 则重置 $Y(s_n^M)^*$

返回第 4) 行

11) end if

12) if $S_M = \emptyset$ or $L_l^* \geq L_l^{\text{thres}}$

13) then 结束迁移过程

14) else 返回至第 6) 行

15) end if

根据算法流程得到 SCM-2 的复杂度为 $O(K)$, 需要对 S_M 中 K 个待迁移交换机进行迭代计算, 且计算方式均为简单线性运算, 收敛度高, 具有实时性和有效性。

4 仿真分析

4.1 仿真环境

关于仿真中实验环境和参数设定做出以下说明。

1) 控制器选择和实验平台搭建。本文应用基于分布式的 OpenDaylight 控制器^[16], OpenDaylight 控制器使用 Java 语言编写, 运行于 JVM。同时为了减少控制器的传输路径开销, 与文献[11]相比, 将控制器与交换机的数量比设定为 1:5。本文算法均采用 Java 语言在 OpenDaylight 应用层进行实现。硬件设施为 3 台具有相同配置的服务器, Intel Core i7, CPU 为 2.8 GHz, RAM 为 8 GB。控制器和 Mininet 在 3 台服务器上基于 VMware 平台以虚拟机的方式实现。2 台设备分别运行普通 OpenDaylight 控制器和装载 SCM 算法的 OpenDaylight 控制器, 1 台设备运行 Mininet 用于创建多控制器网络拓扑。3 台设备均连接到硬件交换机上 (型号 H3C S5500)。

2) 网络拓扑。为了使实验更具代表性, 实验中拓扑模型均取自 OS3E^[17]和 Topology zoo^[18]。首先在 OS3E 网络中完成算法验证, 然后在 Topology zoo 中选取部分拓扑重新实验, 验证算法的拓扑适应性。

3) 实验参数设定。设定交换机请求速率处于动态范围内, 且服从 Poisson 分布, 平均请求速率为 0.5 MB/s。同时为了体现各子域中控制器状态的差异性, 控制器处理容量为 10 MB~15 MB, 按照控制器的剩余处理容量将其划分为重载和轻载状态^[12], 如果控制器剩余处理容量低于 10%, 则认为该控制器处于重载状态, 否则将其视为轻载控制器。假设网络中所有控制器不可能同时处于高负载状态。借鉴文献[9]中权值设置, 本文考虑时延和控制流量是主要作用因素, 同时兼顾故障代价, 设定流请求传输时延、控制器同步时延、控制流量和控制流量故障

代价的权值比为 0.3:0.3:0.3:0.1。

4.2 仿真实验及分析

为了验证本文设计的交换机竞争迁移 (SCM) 算法性能, 将 SCM 算法同文献[8]中交换机弹性迁移 (SEM, switch elastic migration) 算法、文献[11]中利用率最低迁移策略 (UMS, under-utilised migration strategy) 在相同的实验场景下进行仿真对比。SEM 算法以线性规划的方式在时间和资源消耗之间进行抉择, 选取交换机实施迁移。而 UMS 算法从迁移策略方面考虑, 将交换机迁移至容量利用率最低的控制器实现负载均衡。

实验 1 本实验基于 OS3E 网络, 当网络中控制器出现过载状况时, 对比了 2 种算法在迁移对象选取方面的性能。OS3E 网络有 34 个节点, 42 条链路, 利用 k -中心 AP 聚类算法^[19]对网络中节点进行聚类划分, 分割为 5 个子域, 并部署相应的控制器 (C1~C5)。网络在初始时刻处于负载均衡状态, 然后向网络中注入大量流量, 使控制器 C1 过载, 如图 2(a)中虚线部分所示。在此基础上, 分别运行 3 种迁移算法, 结果如图 2(b)~图 2(d)所示。通过拓扑对比可以看出, SCM 得到的子域节点部署结果更加均衡且节点分布集中。SEM 虽然能够减轻 C1 负载, 但造成 C1 和 C2 域内节点数量严重失衡。UMS 迁

移交换机至容量利用率最低的控制器 C3, 使 C3 域内节点分布较为分散, 交换机和控制器产生较多通信开销。

为了量化算法在迁移对象选取方面的性能, 统计图 2 中算法运行时间、控制器通信开销和流表建立时延, 结果分别如图 3~图 5 所示。

通过直观的数据对比可以看出, 使用 SEM 选取迁移对象时, 首先需要对比延、流量、失联性等代价进行线性规划和关联性分析, 然后再优化求解, 算法运行时间达到了 5.5 s。同时 SEM 迁移过载域中 3 个邻近交换机到控制器 C2 的子域, 产生较高的通信开销, 但由于选取的交换机邻近 C2, 因此, 流表建立时延处于较低水平。UMS 的策略是将交换机迁移到容量使用率最低的控制器 C3, 算法的运行时间最短 (2.7 s), 且省去了不必要的控制器通信开销, 所带来的负面影响是流表建立时延达到了 65 ms。相较于以上 2 种算法, SCM 的主要优势在于对迁移对象进行分类, 设计了不同的选取规则。根据控制器剩余处理容量和交换机请求速率分别确定迁出域和待迁移交换机; 借助遗传算法, 综合多种代价确定迁入域。因此, 尽管 SCM 算法对优化目标进行迭代探索, 牺牲一定的算法运行时间, 较 UMS 算法有小幅提升, 但通过对迁移对象进行分类

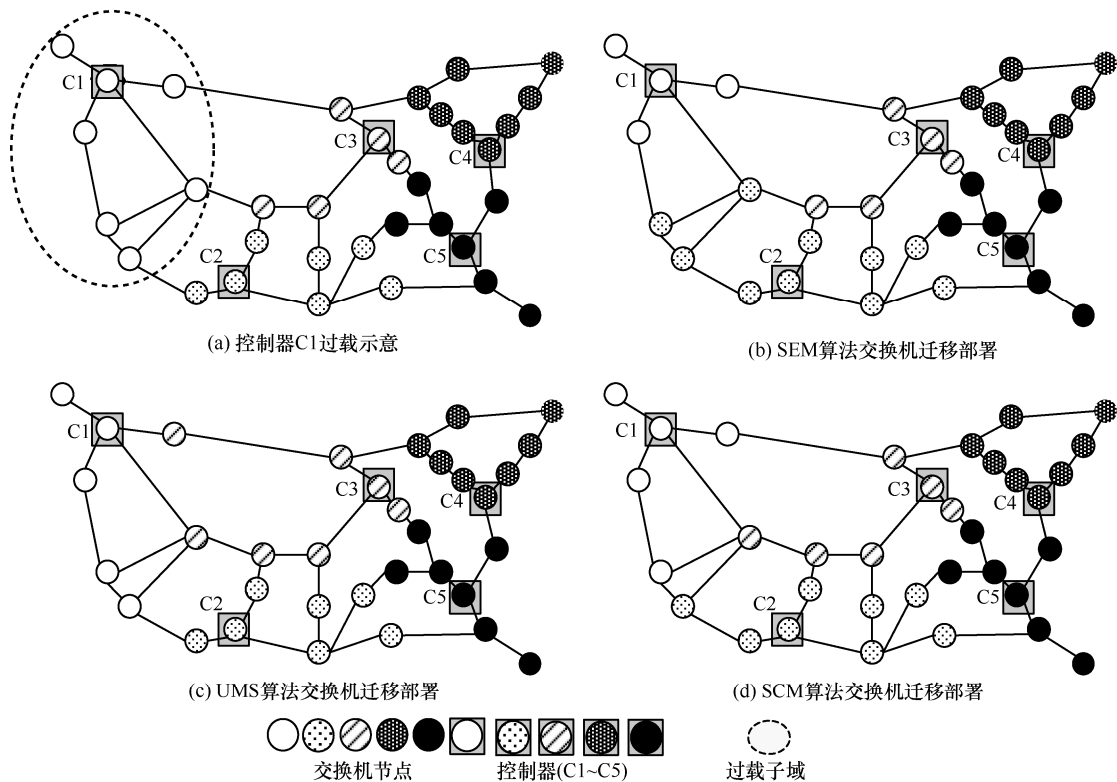


图 2 分布式 SDN 子域网络拓扑

求解, 保证算法运行的可靠性, 同时也使控制器通信开销和流表建立时延都处于 3 类算法中最低值, 比 UMS 算法分别下降了 19.4%和 24.7%。

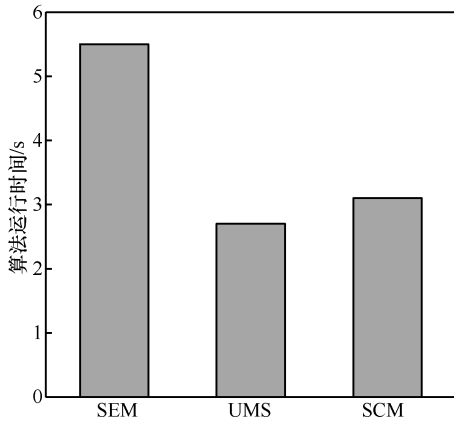


图 3 算法运行时间对比

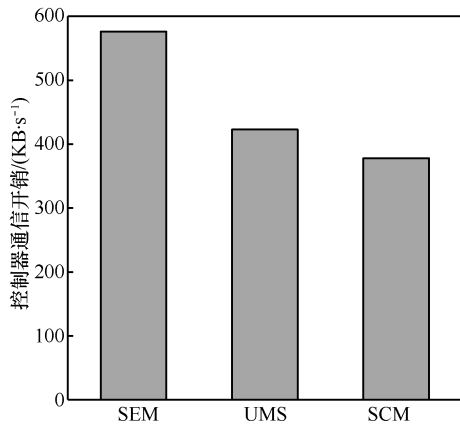


图 4 交换机迁移过程的通信开销

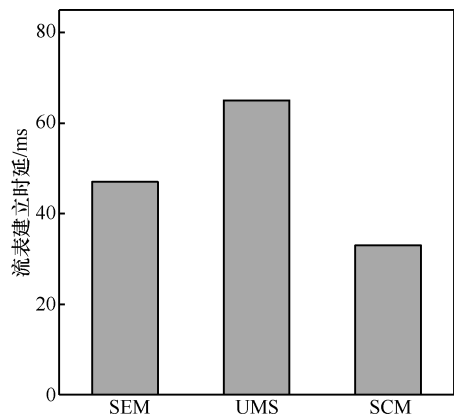


图 5 流表建立时延

实验 2 在迁移对象确定的基础上, 研究交换机动态部署过程中 3 类算法对时延和控制器资源利用率的影响, 同样基于 OS3E 网络, 仿真结果如图 6~图 8 所示。

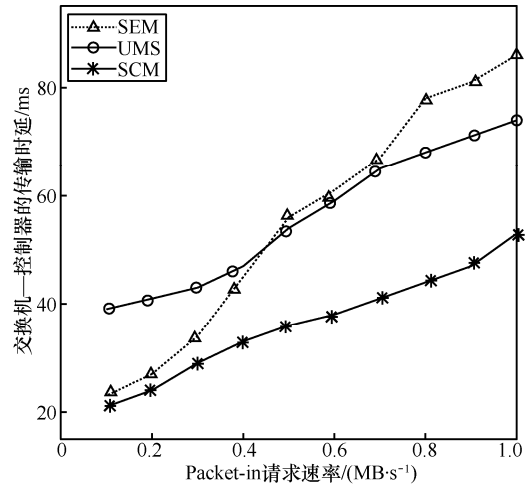


图 6 传输时延

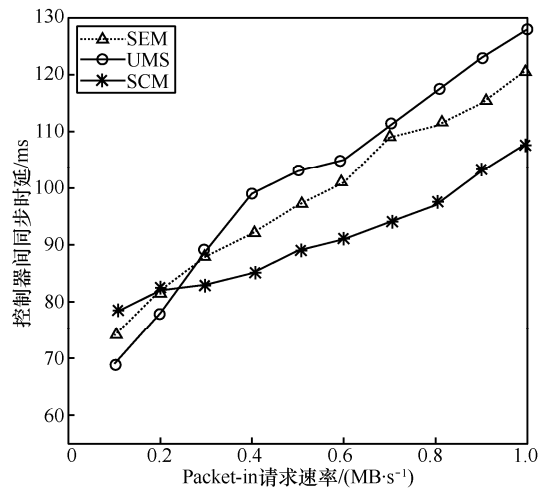


图 7 控制器间同步时延

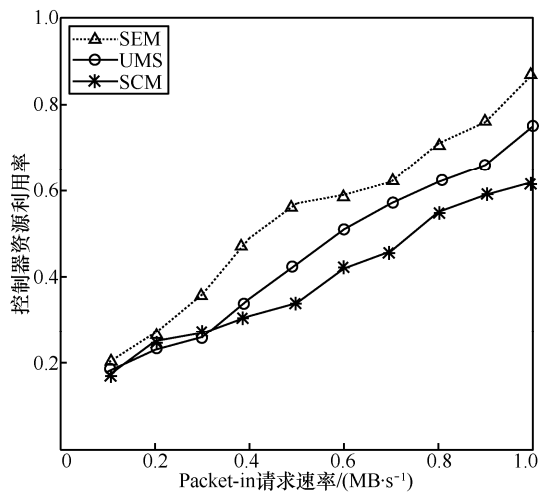


图 8 控制器资源利用率

在交换机迁移过程中, 时延随 Packet-in 请求速率变化趋势如图 6 和图 7 所示。随着 Packet-in 请求速率的上升, SEM 算法将多种非线性因素转

换为线性规划，需要收集各类消息，设备间频繁地进行信息交互，时延上升趋势最为明显，传输时延增加了 63 ms，控制器间同步时延增加了 48 ms。UMS 算法从控制器处理容量角度出发，虽然有效缓解了数据传输过程中流拥塞程度，传输时延增长幅度较低，但由于控制器间距离较大，导致控制器同步时延增长 60 ms。SCM 算法在迁移时设定交换机存活期和有效期，通过周期性竞争比较，选取交换机进行有序迁移，传输时延和控制器同步时延的增长规模都处于较低值，分别为 31 ms 和 27 ms。

在控制器资源利用率方面，进行归一化处理，结果如图 8 所示。相比于 SEM 算法和 UMS 算法，SCM 算法设定控制器负载动态门限值，实现控制器处理资源的合理利用，在 Packet-in 请求速率较高时，通过竞争排序避免了交换机迁移冲突，仍能保证控制器的处理性能。

实验 3 在 Topology zoo 选取部分网络拓扑对 SCM 算法的拓扑适应性以及控制器负载均衡性能进行验证。选取的拓扑信息如表 3 所示。

| 拓扑名称 | 节点数 | 链路数 |
|-----------|-----|-----|
| OS3E | 34 | 42 |
| Germany50 | 50 | 88 |
| Columbus | 70 | 85 |
| TW | 96 | 118 |

不同的网络拓扑中，保证控制器数量不超过网络节点总数的 25%，同时设置各节点上交换机的请求速率大于 1 MB/s，此时网络中部分控制器出现过载现象，对比 SEM、UMS、SCM 和 NSCM（不使用 SCM）这 4 种算法对控制器负载均衡性能的影响，通过计算控制器相对负载标准差得到控制器负载均衡率。实验结果如图 9 所示，其中，横坐标 $\frac{M}{N}$ 是网络中控制器个数与交换机个数的比例。

根据仿真结果，SCM 算法在不同的网络拓扑中均具有较好的控制器负载均衡性能。分析不同算法的控制器负载均衡率随 $\frac{M}{N}$ 的变化情况。当 $\frac{M}{N} < 5\%$

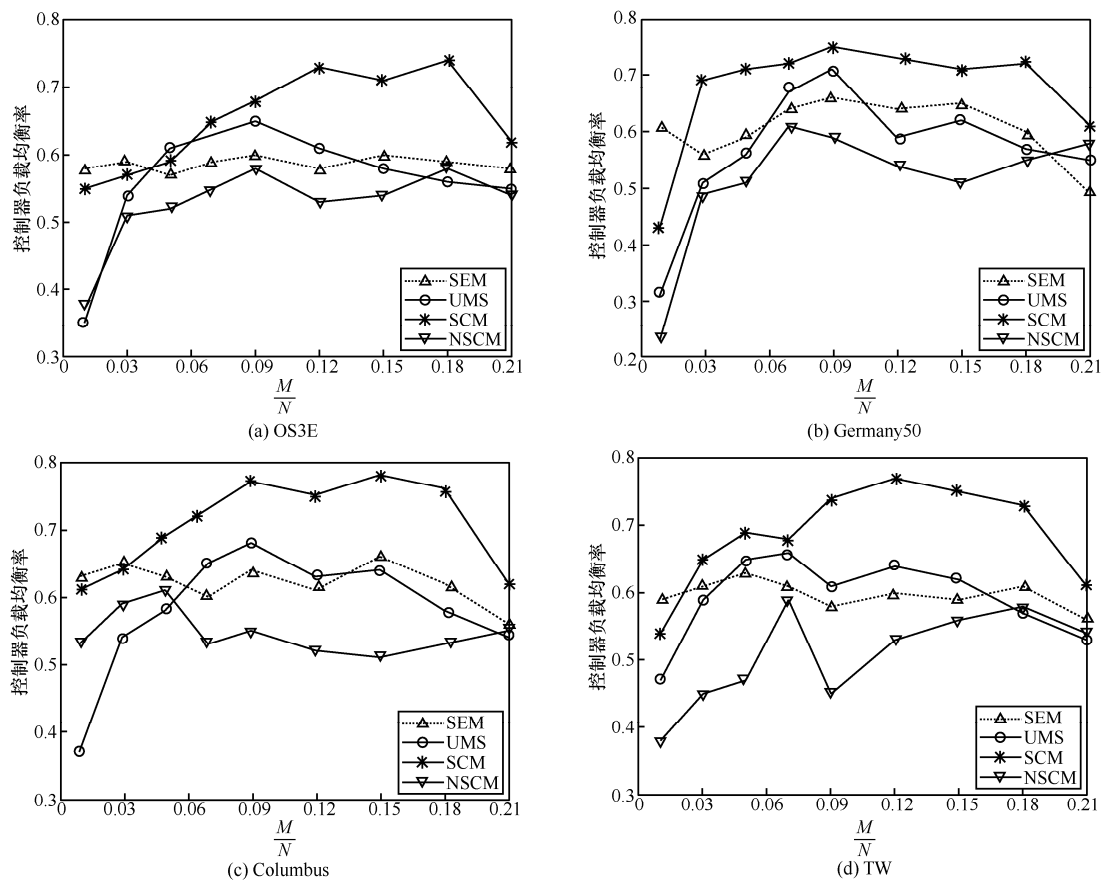


图 9 不同拓扑条件下算法性能比较

时,少量控制器难以应对交换机的高请求速率,控制器处于过载状态,此时,4种算法的控制器负载均衡率都处于较低值,并且UMS和NSCM的控制器过载状况最为严重,这是因为当控制器均具有高容量利用率时,UMS算法失效,而NSCM未采取任何负载迁移措施。

随着 $\frac{M}{N}$ 的比值增大(5%~20%),网络中控制器数量逐渐增多,新增控制器可以分担过载控制器的负载,因此4种算法的控制器负载均衡率均有不同程度提升。相比较而言,NSCM的控制器负载均衡率最低,SEM和UMS次之,SCM最高。这是因为NSCM只是单纯地依靠增加新控制器来调整网络负载分布,是一种静态调节方式,负载调节范围有限且容易产生局部优化。SEM综合多种开销代价,通过线性规划方式选取交换机实施迁移,尽管控制器负载均衡率较为稳定,但该方式的优化粒度较粗,且算法运行时间长(如图3所示),负载调节具有滞后性,控制器负载均衡率未得到较大提升。UMS仅根据控制器容量确定交换机迁入域,当多个交换机同时迁入该域时未考虑迁移冲突,由此导致UMS的控制器负载均衡率处于较大波动状态。SCM基于遗传算法对迁入、迁出域和待迁移交换机进行细粒度优化,灵活地选取迁移目标,同时设定存活期、迁移期和动态负载门限值,避免了交换机迁移冲突。虽然SCM的2阶段设计牺牲了算法的运行时间,但控制器负载均衡率得到明显提升。相比其他3种算法,SCM的控制器负载均衡率平均提升了25.6%。

当 $\frac{M}{N} > 20\%$ 时^[12],网络中出现控制器冗余,闲置控制器的出现增加了控制器负载均衡难度,此时已经不能完全依靠迁移交换机来实现控制器负载均衡,SCM和其他3种算法之间的负载均衡率差异逐渐缩小,必须采取提高控制器性能的方式进行解决。

5 结束语

本文针对SDN多控制器负载均衡过程中,交换机迁移对象选取僵化和多交换机迁移冲突问题,提出了一种基于过程优化的交换机竞争迁移(SCM)算法,分为迁移对象选取和交换机动态部署2个阶段。通过多代价综合分析和迁移参数设定,有效提高交换机迁移效率(时延降低、通信开销减小)和控制器负载均衡率。相比于现有

的交换机迁移算法,本文算法的优点体现在:1)选取迁移对象时,SCM算法考虑传输时延、同步时延和控制流量等因素,并引入故障场景,借助启发式算法进行多代价综合判定,改善了迁移对象选取僵化问题;2)动态部署交换机时,设定存活期、迁移期和控制器负载动态门限值,以竞争迭代的方式避免交换机迁移冲突。实验结果表明,SCM算法较好地优化了交换机迁移过程,虽然在部分度量(如算法运行时间)方面略有欠缺,但对于传输时延、同步时延和控制器资源利用率改善明显,迁移过程中通信开销下降19.4%,且算法对于不同的网络拓扑具有普适性,控制器负载均衡率平均提升了25.6%。

在交换机迁移过程中,针对设备故障,本文只选取了控制器进行建模分析,未考虑交换机失效,且算法运行时间不具有明显的竞争优势,因此,在下一步工作中,将综合考虑多设备失效状况,改进现有的启发式算法。

参考文献:

- [1] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks[C]//ACM SIGCOMM Computer Communication Review. 2008, 38(2): 69-74.
- [2] 左青云, 陈鸣, 赵广松, 等. 基于OpenFlow的SDN技术研究[J]. 软件学报, 2013, 24(5): 1078-1097.
- [3] ZUO Q Y, CHEN M, ZHAO G S, et al. Research on OpenFlow-based SDN technologies[J]. Journal of Software, 2013, 24(5):1078-1097.
- [4] TOOTOONCHIAN A, GANJALI Y. HyperFlow: a distributed control plane for OpenFlow[C]//The Internet Network Management Conference on Research on Enterprise Networking. 2010: 3-3.
- [5] HASSAS Y S, GANJALI Y. Kandoo: a framework for efficient and scalable offloading of control applications[C]//The 1st Workshop on Hot Topics in Software Defined Networks. 2012: 19-24.
- [6] KOPONEN T, CASADO M, et al. Onix: a distributed control platform for large-scale production networks[C]//USENIX Conference on Computer Systems Design and Implementation. 2010:351-364.
- [7] HELLER B, SHERWOOD R, MCKEOWN N. The controller placement problem[C]//Proc of HotSDN. 2012: 7-12.
- [8] 张联镇, 黄传河, 曾毓珑, 等. SDN中基于时延优化的多控制器部署方案[J]. 计算机工程与设计, 2015, 05:1121-1125.
- [9] ZHANG L Z, HUANG C H, ZENG M L, et al. Multi-controller deployment scheme based on delay optimization in SDN[J]. Computer Engineering and Design, 2015, 05:1121-1125.
- [10] HOCK D, GEBERT S, HARTMANN M H, et al. POCO-framework for Pareto-optimal resilient controller placement in SDN-based core network[C]//The 2014 IEEE Network Operations and Management

Symposium. 2014:1-2

- [9] EUGEN B, RADU B, SERBAN G O, et al. On multi controller placement optimization in software defined networking based WANs[C]//The 14th International Conference on Networks. 2015: 261-266.
- [10] ADLEN K, MILOUD B, TARIK T. On using bargaining game for optimal placement of SDN controllers[C]//2016 IEEE International Conference on Communications (ICC). 2016: 1-6.
- [11] YAO G, BI J, LI Y, et al. On the capacitated controller placement problem in software defined networks[J]. IEEE Communications Letters, 2014,18(8): 1339-1342.
- [12] CHENG G Z, CHEN H C, WANG Z M, et al. DHA: distributed decisions on the switch migration toward a scalable SDN control plane[C]//2015 IFIP Networking Conference (IFIP Networking). 2015: 1-9.
- [13] 覃匡宇, 黄传河, 王才华, 等. SDN 网络中受时延和容量限制的多控制器均衡部署[J]. 通信学报, 2016, 37(11): 90-103.
- QIN K Y, HUANG C H, WANG C H, et al. Balanced multiple controllers placement with latency and capacity bound in software-defined network[J]. Journal on Communications, 2016, 37(11): 90-103.
- [14] LING S H, LEUNG F H F. Real-coded genetic algorithm with average-bound crossover and wavelet mutation for network parameters learning[C]//IEEE International Joint Conference on Neural Networks. 2005: 1325-1330.
- [15] HU Y N, WANG W D, GONG X Q, et al. On reliability-optimized controller placement for software-defined networks[J]. China Communications, 2014, 11(2): 38-54.
- [16] MAYORAL A, VILALTA R, MUNOZ R, et al. Experimental validation of automatic lightpath establishment integrating OpenDayLight SDN controller and Active Stateful PCE within the ADRENALINE testbed[C]//The 16th International Conference on Transparent Optical Networks. 2014:1-4.
- [17] LIU B, MANNEMELA P. Internet2 end-to-end performance tuning for distributed computing applications[C]//IEEE International Conference on Industrial Technology. 2005: 587-592.
- [18] 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.
- [19] FERY B J, DUECK D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814): 972-976.

作者简介:



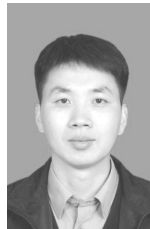
胡涛 (1993-), 男, 陕西武功人, 国家数字交换系统工程技术研究中心硕士生, 主要研究方向为宽带信息网、软件定义网络。



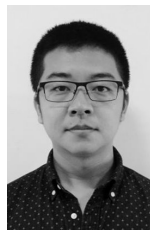
张建辉 (1977-), 男, 河南平顶山人, 国家数字交换系统工程技术研究中心副研究员, 主要研究方向为宽带信息网、网络安全。



孔维功 (1980-), 男, 河南封丘人, 国家数字交换系统工程技术研究中心博士生, 主要研究方向为宽带信息网。



杨森 (1985-), 男, 辽宁盖州人, 国家数字交换系统工程技术研究中心助理研究员, 主要研究方向为通信与信息网络。



曹路佳 (1983-), 男, 河北抚宁人, 解放军信息工程大学助教, 主要研究方向为网络安全。