

软件定义网络中基于负载均衡的多控制器部署算法

史久根 郝伟* 贾坤荣 徐颖

(合肥工业大学计算机与信息学院 合肥 230009)

摘要: 随着软件定义网络规模扩大, 控制层与数据层解耦带来了诸如控制器部署等新问题。该文提出基于负载均衡的多控制器部署算法(Multi-Controller Deployment Algorithm Based on Load Balance, MCDALB)。算法首先根据网络拓扑结构及其负载情况, 确定控制器数量 K ; 然后根据控制器容量限制, 提出一种近似比为 2 的多控制器负载均衡算法, 将网络划分成 K 个控制区域; 最后根据区域内所有交换机到控制器距离总和最小原则, 在控制区域部署控制器。为了验证算法的性能, 选取实际网络拓扑进行实验。实验结果表明, 与 AL, WL 算法相比, 该算法在满足控制器负载近似比为 2 的同时, 网络最大延时差距不超过 0.65 ms。

关键词: 软件定义网络; 控制器部署; 负载均衡

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2018)02-0455-07

DOI: 10.11999/JEIT170464

Multi-controller Deployment Algorithm Based on Load Balance in Software Defined Network

SHI Jiugen ZHU Wei JIA Kunying XU Ying

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

Abstract: With the expansion of Software Defined Network (SDN), the decoupling of control layer and data layer brings new problems such as controller deployment. In this paper, a Multi-Controller Deployment Algorithm Based on Load Balance (MCDALB) in SDN is proposed. Firstly, the number, K , of controllers is determined based on network topology and its load. Secondly, according to the limitation of controller capacity, a multi-controller load balance algorithm with approximate ratio of 2 is proposed, which divides the network into K control regions. Lastly, the position of the controller in each region is selected, according to the minimum sum of all switch-to-controller distances in the region. In order to verify the performance of the proposed algorithm, the actual network topologies are applied. As to compare with the AL and WL algorithms, simulation results show that the proposed algorithm not only balances the controllers' load, with an approximation ratio of 2, but also meets the maximum gap of network delay not more than 0.65 ms.

Key words: Software Defined Network (SDN); Controller deployment; Load balance

1 引言

软件定义网络(SDN)^[1]作为一种新型的网络创新架构, 其核心技术 OpenFlow 通过将网络控制层与数据层解耦, 实现了网络流量的灵活控制, 为核心网络及其应用的创建提供了良好的平台。控制层通过逻辑上集中的控制器对数据层实现管理与操作。然而, 随着 SDN 网络的扩大, 控制层与数据层解耦带来了一些新的问题, 比如控制器部署问题^[2]。

控制器部署问题即给定一个 SDN 拓扑网络, 如

何寻找 K 个最佳控制器位置。在小型的局域网中, 一般部署一个控制器即可满足各种需求, 但随着网络规模扩大, 现有的单控制器方案无法满足大型的广域网络部署需求。为了解决单控制器性能不足问题, 国内外研究者相继提出一系列的方案^[3-5], 文献[3]改进控制器软件, 提高控制器性能, 进而提出了 HyperFlow 方案^[4], 用于解决控制层的扩展性和网络恢复性问题; Koponen 等人^[5]提出 Onix 方案, 致力于解决控制平面的结构扩展问题。上述方案未注意到寻找控制器节点位置并进行控制器部署, 具有一定的局限性。Heller 等人^[2]首次提出控制器部署问题, 即在给定的网络拓扑中, 需要多少个控制器, 以及这些控制器应该部署在什么位置, 并指出该问题是一个 NP 难问题, 但未给出具体的解决算法。

收稿日期: 2017-05-16; 改回日期: 2017-07-17; 网络出版: 2017-08-29

*通信作者: 郝伟 20112543@mail.hfut.edu.cn

基金项目: 国家重大科学仪器设备开发专项(2013YQ030595)

Foundation Item: The National Major Scientific Instruments Development Project (2013YQ030595)

在真实的网络环境中, 需要根据实际需求, 综合考虑影响控制器部署问题的各种因素, 并在这些因素间做出权衡^[6]。Hock 等人^[7]致力于在控制器部署问题中加入多种可恢复性因素, 便于获得在不同的部署条件下, 各种互为矛盾的优化目标之间的权衡^[8], 但无法将其应用于有延时限制的大规模广域网中。对于网络延时问题^[2], Sahoo 等人^[9]考虑控制器间的延时, 并提出基于模拟退火算法的解决方案, 但没有注意到控制器负载问题。对于网络的负载问题, Rath 等人^[10]基于非零和游戏理论提出动态放置策略, 但没有给出网络中控制器的初始位置。

综上, 本文综合考虑广域网中控制器的负载能力和延时问题, 提出一种基于负载均衡的多控制器部署算法(MCDALB)。首先根据已有的网络拓扑, 构建一棵优化的最小生成树, 并根据网络负载情况, 确定控制器数量 K 。其次, 提出一种基于负载均衡的最小生成树近似划分方法, 将最小生成树划分为 K 棵子树, 得到 K 个控制域。然后, 基于网络中距离和最小原则, 在每个控制域中部署一个控制器。最后, 仿真结果表明, 该方案在满足控制层负载均衡的同时, 能够很好的实现延时控制。

本文后续章节安排如下, 第2节, 给出问题描述与系统模型, 第3节介绍控制器部署算法, 第4节给出仿真实验, 最后给出总结与展望。

2 问题描述与系统模型

在这一节, 本文介绍 SDN 控制域划分, 并讨论将要优化的控制器部署指标。

2.1 问题描述

SDN 中, 对控制器管理的交换机而言, 控制器扮演着信息收集和操作者的角色。就这点而言, 控制器与交换机之间的反应时间影响着 SDN 网络的性能。进一步说, 控制器的反应时间由控制器的负载容量以及控制器与交换机间的距离共同决定, 控制器的负载容量决定了数据流处理时间的快慢, 控制器与交换机间的传播延时与其距离成正比。

(1) 负载均衡: 负载均衡是控制器性能一个重要指标。文献[4]考虑控制器性能问题, 并发现影响控制器服务能力的限制因素——足够大的延时和控制器过载将导致控制器无法进行实时控制。因而, 通过将大的 SDN 广域网络划分成若干个小而且均衡的网络, 控制器的服务质量将得以提高, 同时也可以抑制控制器间的广播风暴。

假设控制器的最大负载能力为 L_{cap_i} , 某一控制器的负载容量为 L_{cap_i} , 当控制器不过载时, 控制器处理请求的排队等待时间 T_{q_i} 相同。如何在给定的拓

扑结构中, 划分控制器网络, 使得每个区域内的控制器负载能力满足 $L_{\text{cap}_i} < L_{\text{cap}_i}$, 即给定一个网络拓扑, 安放多少个控制器, 以及确定控制器位置, Heller 等人^[2]指出该问题是一个 NP 难问题, 本文将给出一个近似比为 2 解决方案。

(2) 传播延时: 影响控制器服务性能的另一个因素是网络延时^[11,12], 在 SDN 网络中, 延时包括 4 个方面: 传播延时、处理延时、排队延时、发送延时。对广域网而言, 在控制器不过载的前提下, 传播延时大于其他 3 种延时, 由于其他 3 种延时的影响很小, 可以忽略。传播延时影响控制器对网络事件处理的性能。因而, 本文主要考虑在控制器满足负载能力的情况下, 最小化传播平均延时。

2.2 系统模型

本文用图 $G(V, E)$ 表示控制器网络, V 表示网络中的点集合, E 表示网络中的边集合。需要在网络中部署 K 个控制器, 即将图 G 划分成 K 个子图, 每个子图 $D_i (i = 1, 2, \dots, K)$ 表示一个控制域。可以对该子图作以下定义。

定义 1 如果将图 G 划分成 K 个子图, 每个子图 $D_i (i = 1, 2, \dots, K)$ 可以表示为 $D_i(V_i, E_i)$ 。每个子图中的节点集等价于将图 $G(V, E)$ 划分为 K 个不相交的子图 $D_1, D_2, \dots, D_K$, 即

$$V = \bigcup_{i=1}^K D_i, D_i \cap D_j = \emptyset, i \neq j, i, j = 1, 2, \dots, K \quad (1)$$

其中, K 表示网络中控制域的数量, D_i 表示第 i 个区域内节点集合, 该区域内的节点数目为 $|D_i|$, 表示为

$$D_i = \{v_{i_1}, v_{i_2}, \dots, v_{i_j}\}, i = 1, 2, \dots, K, j = |D_i| \quad (2)$$

划分完网络, 得到 K 个不相交的子网络后, 下一步要做的事是在根据优化目标, 在子网络中选择最佳位置 $C_i (i = 1, 2, \dots, K)$, 部署控制器, 进而, 本文给出部署方案定义。

定义 2 在划分好的 K 个子网络中, 控制器部署方案为

$$C = (C_1, C_2, \dots, C_K) \quad (3)$$

即在每一个子网络中, 以最小传播延时为目标, 选择一个最佳的控制器部署位置, 最终得到整个网络的控制器部署方案。

上述两个定义中, 定义 1 对应于网络的负载均衡问题, 定义 2 对应于网络延时问题, 为进一步描述问题, 本文分别给出两个定义的数学模型。

在网络 $G(V, E)$ 中, 网络中存在 $|V|$ 个交换机, 每个交换机 $V_j (j = 1, 2, \dots, |V|)$ 的负载请求表示为

W_{V_j} ，那么第 i 个网络 $D_i(V_i, E_i)$ 中控制器的负载容量 L_{capt_i} 可以表示为

$$L_{\text{capt}_i} = \sum_{V_j \in D_i} W_{V_j}, j = 1, 2, \dots, |V| \quad (4)$$

其中，子网络中的负载 L_{capt_i} 不能大于控制器 C_i 自身的最大负载能力 L_{cap_i} ，即满足：

$$\lambda L_{\text{cap}_i} > L_{\text{capt}_i}, i = 1, 2, \dots, K, 0 \leq \lambda \leq 1 \quad (5)$$

在保证每个子网络中控制器不过载的情况下，本文以最小传播延时为目标，确定最佳的控制器部署位置。网络的传播延时与控制器和交换机间的地理距离有关，对于网络中的任意两个节点 V_i, V_j , $i \neq j, i = 1, 2, \dots, |V|, j = 1, 2, \dots, |V|$ ，分别由式(6)、式(7)得到两个节点间的距离以及传播时间^[13]：

$$\xi = \sin^2(\alpha/2) + \cos(\text{lat}_i) \cos(\text{lat}_j) \sin^2(\beta/2) \quad (6)$$

$$t_{ij} = \frac{2 \arcsin \sqrt{\xi \times R}}{v_c} \quad (7)$$

$$\alpha = |\text{lat}_i - \text{lat}_j| \quad (8)$$

$$\beta = |\text{lon}_i - \text{lon}_j| \quad (9)$$

$$R = 6378.137 \text{ km} \quad (10)$$

$$v_c = 3 \times 10^8 \text{ m/s} \quad (11)$$

其中，节点 i, j 的经纬坐标表示为 $V_i(\text{lat}_i, \text{lon}_i)$, $V_j(\text{lat}_j, \text{lon}_j)$ 。式(8)表示两点间的纬度坐标之差，式(9)表示两点之间的经度坐标之差，式(10)表示地球半径，式(11)表示网络信号传播速度。网络中，节点 i 到该网络中其他节点的最短时间总和为：

$$T_i = \sum_{j=1, j \neq i}^{|D_i|} t_{ij} \quad (12)$$

进而，本文需要在每个子网络中，选取到某一位置 i 部署控制器 C_i 使得该位置到该区域内其他节点的平均距离最短，即满足：

$$i = \min(T_i), i = 1, 2, \dots, |D_i| \quad (13)$$

重复在选取 K 个子网中选择最优部署位置，最终得到部署方案 $C = (C_1, C_2, \dots, C_K)$ 。进而，整体网络的平均延时时间为

$$\bar{T} = \frac{1}{K} \sum_{i=1}^K T_i \quad (14)$$

下一节中，本文将基于上述定义及模型，提出基于最小生成树的子网络负载均衡算法，得到 K 个子网络，然后，根据最小延时，在每个控制区域内部署控制器，最后得到本文所需要的控制器部署方案。

3 控制器部署算法 MCDALB

在这一节中，将详细描述控制器部署算法

(MCDALB)。该算法以网络负载均衡为目标，并最小化网络延时。网络负载均衡，即各控制器节点的负载能力近乎相同，或者说最小化负载能力最大的控制器，满足表达式(15)：

$$\min \max(L_{\text{capt}_i}), i = 1, 2, \dots, K \quad (15)$$

在 SDN 网络中，交换机可能与多个控制器有物理连接，然而该交换机只受某一特定的控制器管理。首先，根据最短距离优先条件^[14]，从复杂的网络图模型中生成一颗最小生成树。然后，根据各交换机的请求量，将树划分成 K 棵的负载近似相等子树，每棵树代表一个控制器管理的子网络。为此本文提出一个近似比为 2 的负载均衡解决方案。最后，根据原有网络的拓扑结构，在每个控制器网络中安放控制器，得到整个网络的部署方案。

3.1 最小生成树

在一个控制器网络 $G(V, E)$ 中，存在 $|V|$ 个交换机节点和 $|E|$ 条边，并且有

$$|E| > |V| \quad (16)$$

式(16)表示网络中的一个交换机在物理上与多个控制器连接。根据本文的优化目标，在满足网络负载均衡的同时，最小化网络传播延时。本文依据控制器与控制器间的最小距离，结合深度优先和最小生成树算法，构建一颗带有负载(交换机的负载用权值表示)的最小生成树 $T(V, E', W)$ 。其中， V 表示生成树的节点， E' 表示生成树中的边，且 $E' \subseteq E$ ， W 表示生成树中节点及其子节点权值和，表示为

$$W_i < w_{h_i}, w_i >, i = 1, 2, \dots, |V| \quad (17)$$

其中， w_{h_i} 表示节点 i 及其子节点的权值和， w_i 表示节点 i 自身的权值。

构建完生成树后，依据式(17)，划分生成树，并部署控制器。本文先从一个实例来描述算法的实施方案。

例如，对一个网络(参考 OS3E)，如图 1 所示：网络中有 10 个交换机节点，13 条边，点与点间的距离形式化表示为 1, 2, ..., 13 每个节点自身的权值为 1。现在需要在该网络中部署 2 个控制器，按照本文的算法，首先要构建最小生成树，如图 2 所示。

接着根据子树划分算法，将生成树划分成两个子树，即将节点 8 和节点 9 间的连接断开后，可以使得两棵子树的负载能力相同，但在网络中节点 9 距离节点 8 更近，根据最小延时原则，将节点 9 分配到节点 8 所在的子树中，即断开节点 9 和节点 10 直接连接，进而得到两棵子树，相应的节点集合为 $\{1, 2, 3, 10\}$ 与 $\{4, 5, 6, 7, 8, 9\}$ ，这两个集合为两个控制器区域。最后，依据交换机到控制器间的最短距离和最小，确定每一个控制区域内控制器的位置，

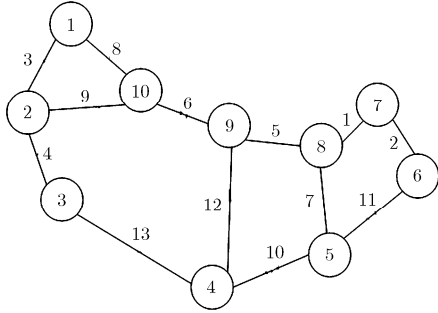


图1 网络拓扑

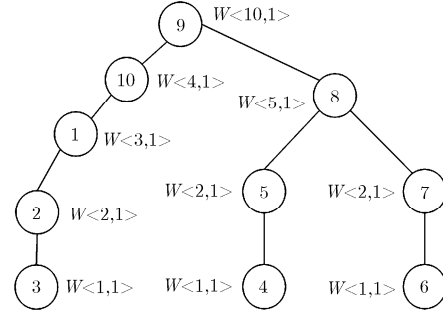


图2 最小生成树

得到最终的部署方案,即选择节点2和节点8部署控制器。

3.2 子树划分

构建好最小生成树后,本文需要根据节点的权值信息 W 将生成树划分成 K 个子树,每棵子树对应一个控制区域,并在每个控制区域内部署一个控制器。控制器数量 K 由网络的总负载与单个控制器的负载 L_{cap_i} 共同决定。理想情况下,在控制器网络中,各个控制器的负载能力应相同或者差别很小。然而控制器部署问题是一个 NP 难问题,为此本文基于最小生成树,提出一种近似算法(MCDALB)来划分网络,进而部署控制器,具体步骤为:

(1)初始化:由3.1节,以及交换机间距离计算表达式,输入网络拓扑模型 $G(V, E)$ 获得最小生成树模型 $T(V, E', W)$; 根据生成树根节点的 W 值,得到网络的总负载,确定需要部署的控制器数量为 K ; 置迭代轮次 $k = 0$;

(2)确定递归迭代次数: $n = \lceil \lg K \rceil$, 即经过 n 次递归迭代即可划分出 K 个控制器区域;

(3)基本划分:假设最小生成树根节点的负载为 W_r , 其中 r 表示最小生成树的根节点,则需要将最小生成树划分成两棵子树 T_1, T_2 。从根节点出发,按照先序遍历方法,遍历最小生成树进行,每遍历一个节点,获取其负载值 W_i , 判断 $W_i \leq W_r/2$ 是否成立,若成立,则停止遍历。获取该节点 V_i 的父亲节点 V_{if} , 计算节点 V_{if} 到其邻居节点的距离,若节点 V_i 到节点 V_{if} 距离最短,则节点 V_{if} 断开除 V_i 外的所有节点的边; 否则断开节点 V_i 与节点 V_{if} 之间的边,得到两棵生成树,并重新调整生成树中各节点的权值 W , 并置迭代轮次 $k = k + 1$ 。

(4)算法终止:如果 $k \geq n$, 则算法停止; 否则计算 $K - 2^k$, 如果 $(K - 2^k) \geq 2^k$, 则以 2^k 个子树作为输入, 转入步骤(3), 进行基本划分; 否则选择 $\lfloor (K - 2^k) - 2^k \rfloor$ 个 W 值最大的子树作为输入, 转入步骤(3), 进行基本划分。

根据上述划分,本文可以得到一个近似比为2的优化策略,为此,本文给出以下定理:

定理1 采用 MCDALB 划分算法,网络中控制器的负载能力 $W_a (W_{a1}, W_{a2}, \dots, W_{aK})$ 与最优解 $W (W_1, W_2, \dots, W_K)$ 的近似比为2。其中

$$W_1 \approx W_2 \approx \dots \approx W_K \quad (18)$$

并且,在SDN网络中,单个节点的负载能力远小于控制器的最小负载,即

$$W_{V_i} \ll \min W_i \quad (19)$$

证明 本文从两个方面来证明该定理。

(1)部署 2^r 个控制器:由于基本划分中,寻找 $W_i \leq W_r/2$ 可知,划分的子树中最大负载为:以 i 为根的子树负载与其父亲节点的负载之和 $W_i + W_{V_{if}}$, 又由式(19)可知:

$$W_i + W_{V_{if}} < W_i + W_i = 2 \times W_i \quad (20)$$

(2)部署 $K = 2^r + t$ 个控制器, $1 \leq t < 2^r$: 经过第 r 次迭代后,选择 t 棵负载最大的子树进行第 $r+1$ 次迭代划分,完成划分后,得到 $2^r + t$ 棵子树,则负载容量最大的控制器在第 r 次迭代后,剩余的 $2^r - t$ 棵子树中,最大的控制器负载不会超过 $W/2^r + W_{V_{if}}$, 又因为 $1 \leq t < 2^r$, 可知,当 t 越大时,该值越大,即,如果 $t = 2^r - 1$ 时,算法满足近似比2, 则对于任意的 t , $1 \leq t < 2^r$, 算法均满足该近似比。

已知,最优情况下,每个控制器的负载能力为 $W/(2^r + t)$, 第 r 次迭代后,控制器的负载为 $W/2^r + W_{V_{if}}$, 又因为单个节点负载远小于控制器的最小负载,此时网络中最小控制器的负载为 $W/2^{r+1}$, 即满足:

$$W_{V_{if}} \ll W/2^{r+1} \quad (21)$$

可以得到:

$$\frac{W}{2^{r+1} - 1} < \frac{W}{2^r} + W_{V_{if}} \approx \frac{W}{2^r} \quad (22)$$

进一步,可以得到:

$$\frac{W}{2^{r+1}-1} < \frac{2W}{2^{r+1}-1} < \frac{2W}{2^r} \quad (23)$$

定理得证。

综上两个方面，得出 MCDALB 划分算法的近似比为 2。该策略由两部分组成，如表 1 中的算法 1 和表 2 中的算法 2 所示。

3.3 确定控制器位置

最小生成树划分完成后，本文根据每个生成子树的节点，在网络拓扑结构中确定每个控制区域，并根据式(13)，式(14)确定各个控制区域内控制器的位置，最终得到控制器部署方案 $C = (C_1, C_2, \dots, C_K)$ 。

由上述过程，可以得到算法的时间复杂度为最小生成树的时间复杂度为 $O(n^2)$ ；网络划分，需要的时间为 $O(\lg K \times n)$ ；控制器位置确定，需要的时间为 $O(n^2)$ 。总的时间为 $O(n^2) + O(\lg K \times n) + O(n^2)$ ，即最终的时间复杂度为 $O(n^2)$ 。

4 仿真实验

为了评估模型的可行性和算法的有效性，本文以实际的网络拓扑为例，进行仿真实验。验证本文算法在不同的网络结构中，控制器的负载情况；并对比 CPP^[2]问题中给出的平均延时 AL(Average-

表 1 基本划分算法

算法 1：基本划分算法	
输入：	生成树 T_k
输出：	两棵生成子树 T_1, T_2
1:	初始化：根节点负载为 W_r ；
2:	先序遍历最小生成树 T_k ，找到节点 i ，满足 $W_i \leq W_r / 2$ ；
3:	获取该节点 V_i 的父亲节点 V_{if} ，计算 V_{if} 到其邻居节点的距离；
4:	if V_i 到 V_{if} 距离最短
5:	then V_{if} 断开除 V_i 外的所有节点的边；
6:	else
7:	断开 V_i 与 V_{if} 之间的边；
8:	end if
9:	得到两棵生成子树 T_1, T_2 ；
10:	重新调整生成树中各节点的权值 W ；
11:	return 两棵子树；

表 2 整体划分算法

算法 2：整体划分算法	
输入：	最小生成树 T ，控制器数量 K
输出：	K 棵子树
1:	初始化： $n = \lfloor \lg K \rfloor, k = 0$ ；
2:	while $n > k$ do
3:	调用基本划分算法；
4:	调整 $k = k + 1$ ；
5:	计算 $K - 2^k$ ；
6:	if $(K - 2^k) \geq 2^k$
7:	then 以 2^k 个子树作为输入，转入 3；
8:	else
9:	选择 $\lfloor (K - 2^k) - 2^k \rfloor$ 个最大的子树作为输入，
	转入 3；
10:	end if
11:	end while
12:	return K 棵子树；

case Latency) 和最坏延时 WL(Worst-case Latency)，验证算法在保证传输延时的同时，实现了负载均衡。

4.1 实验环境

本文选择的仿真平台是 matlab2010，系统配置为：win7 64 位系统，i7 处理器，8 GB 内存。实验采用的网络拓扑为 Internet2(Internet OS3E)^[2]和 Ntelos^[15]。网络信息以及 7 种情况下，各交换机的初始平均流表安装请求数如表 3 所示。已知各个节点的经纬度坐标及连接关系，利用式(6)~式(11)，确定城市间的距离及传输时间。设定个交换机的最大负载能力 L_{cap_i} 为 1800 kb/s。网络部署时，控制器的负载能力满足式(5)，考虑到网络在运行过程中，各个控制器交换机的请求会发生变化，因而，规定参数 $\lambda \leq 0.5$ 为最佳部署方案。

4.2 性能分析

图 3 表示在不同的拓扑网络中，分别部署 2,3,4,5,6,7,8 个控制器时，控制器的负载情况。纵坐标表示控制器负载的近似比(定理 1)，利用最大负载核最优负载的比值表示，其中的最大负载即单个控制器处理的最大流表安转请求数量，即利用式(4)、式(15)

表 3 网络信息及流表请求量(kb/s)

网络名	节点数	边数	情况 1	情况 2	情况 3	情况 4	情况 5	情况 6	情况 7
OS3E	34	42	50.0	60.0	70.0	90.0	120.0	180.0	180.0
Ntelos	48	61	37.5	56.3	75.0	90.0	112.5	128.6	150.0

求取的结果; 最优负载, 满足式(18), 即网络中的各个控制器的负载能力近似相等。实验环境, 如表3所示。从图3可以看出, 本文提出的方案, 在不同网络拓扑结构中, 交换机的初始请求发生变化时, 近似比小于2。根据本文的算法的划分方式, 当网络中部署 $K = 2^r$ 个控制器, 即 $K = 2, 4, 8$ 时, 其负载均衡效果最好, 比值接近1; 当部署 $K = 2^r + t$ 个控制器, 随着 t 值的增加, 近似比相应增加, 最大值分别为1.65和1.60, 满足定理1。

图4表示在 Internet2(Internet OS3E)网络中, 网络延时情况。实验环境, 如表3所示。部署2, 3, 4, 5, 6, 7, 8个控制器时, 利用式(13)与式(14), 计算MCDALB算法的延时, 并比较网络的平均延时AL和最坏延时WL。其中, AL定义为: $L_{avg}(C) = \frac{1}{n} \sum_{v \in V} \min_{(c \in C)} d(v, c)$, 表示网络中的每个交换机向距离其欧几里得距离最近的控制器发送请求; WL定义为: $L_{wc}(C) = \max_{(v \in V)} \min_{(c \in C)} d(v, c)$, 表示网络中控制器与交换机的最大通信距离。仿真结果表明, MCDALB的延时情况介于AL与WL之间, 且与AL接近。由于MCDALB在划分好控制区域后, 利用式(13)与式(14), 确定控制器的位置, 保证了控制域内的最小延时。其中, 在网络中部署3个控制器时, MCDALB, AL和WL算法中, 单个控制器管理的最大交换机数量分别为16, 17, 22; 在网络部署8个控制器时, 单个控制器管理的最大交换机数量分别

为5, 9, 9。实验结果表明, 在部署3个控制器时, MCDALB延时与AL延时差别最大, 为0.65 ms, 原因是为了保证负载均衡, 需要将部分交换机调整至距离其相对较远的控制器, 因而增加了网络延时。

图5表示在 Internet2(Internet OS3E)网络中, 不同网络负载情况。实验环境, 如表3所示。部署2, 3, 4, 5, 6, 7, 8个控制器时, 比较3种方案的负载近似比。控制器的部署结果和图4一致。实验仿真结果表明, MCDALB能更好地实现各个控制器间的负载均衡。采用WL方案, 在交换机密集区域内, 控制器管理的交换机数量较多, 控制器易过载, 对于式(5), 控制器数量为3和8时, λ 取值分别为1.00, 1.06; 同样AL方案以最小化延时为目标也存在控制器过载情况, 控制器数量为8时, λ 的取值为1.06。综合图4和图5分析, MCDALB在保证传输延时的同时, 更好的实现了负载均衡。

5 结束语

本文提出了一种软件定义网络中, 基于控制器负载均衡的多控制部署算法。理论分析以及仿真实验表明, 本文算法在不同网络中, 其控制器最大请求负载与最优负载的近似比为2。算法在满足负载均衡的同时, 能够最小化网络延时。然而, 本文提出的算法还存在以下不足, 本文提出的是一种静态的网络负载均衡算法, 在实际网络中, 交换机的请求量不断变化^[16,17], 需要考虑运行时的网络过载情况, 这也是本文未来的研究方向之一。

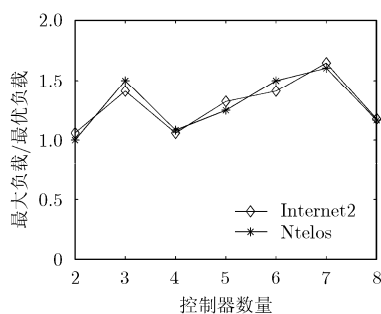


图3 控制器负载情况

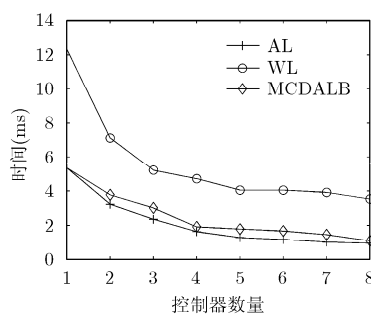


图4 网络延时情况

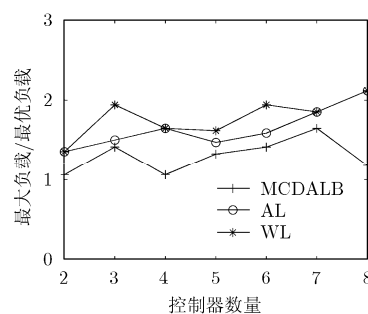


图5 不同算法负载情况

参考文献

- [1] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, *et al.* OpenFlow: Enabling innovation in campus networks[J]. *ACM SIGCOMM Computer Communication Review*, 2008, 38(2): 69-74. doi: 10.1145/1355734.1355746.
- [2] HELLER B, SHERWOOD R, and MCKEOWN N. The controller placement problem[C]. *The First ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, New York, USA, 2012: 7-12. doi: 10.1145/2342441.2342444.
- [3] TOOTOONCHIAN A, GORBUNOV S, GANJALI Y, *et al.* On controller performance in software-defined networks[C]. *Usenix Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, San Jose, CA, USA, 2012: 10-15.
- [4] TOOTOONCHIAN A and GANJALI Y. HyperFlow: A distributed control plane for OpenFlow[C]. *Internet Network Management Conference on Research on Enterprise*

- Networking, San Francisco, CA, USA, 2010: 3–8.
- [5] KOPONEN T, CASADO M, GUDE N, *et al.* Onix: A distributed control platform for large-scale production networks[C]. Usenix Symposium on Operating Systems Design and Implementation, Vancouver, Canada, 2010: 351–364.
- [6] VOCHIN M, BORCOCI E, and AMBARUS T. On multi-controller placement optimization in software defined networking based WANs[C]. The Fourteenth International Conference on Networks ICN, Barcelona, Spain, 2015: 261–266.
- [7] HOCK D, HARTMANN M, GEBERT S, *et al.* Pareto-optimal resilient controller placement in SDN-based core networks[C]. IEEE 2013 25th International Teletraffic Congress (ITC), California, USA, 2013: 1–9. doi: 10.1109/itc.2013.6662939.
- [8] LANGE S, GEBERTS, ZINNER T, *et al.* Heuristic approaches to the controller placement problem in large scale SDN networks[J]. *IEEE Transactions on Network & Service Management*, 2015, 12(1): 4–17. doi: 10.1109/tnsm.2015.2402432.
- [9] SAHOO K S, SAHOO B, DASH R, *et al.* Optimal controller selection in software defined network using a greedy-SA algorithm[C]. 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, New Delhi, India, 2016: 2342–2346.
- [10] 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]. Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Sydney, Australia, 2014: 1–6. doi: 10.1109/WoWMoM.2014.6918987.
- [11] MATTOS D M F, DUARTE O C M B, and PUJOLLE G. A resilient distributed controller for software defined networking[C]. 2016 IEEE International Conference on Communications (ICC). IEEE, Kuala Lumpur, Malaysia, 2016: 1–6. doi: 10.1109/icc.2016.7511032.
- [12] 姚琳元, 陈颖, 宋飞, 等. 基于时延的软件定义网络快速响应控制器部署[J]. 电子与信息学报, 2014, 36(12): 2802–2808. doi: 10.3724/SP.J.1146.2014.00211.
- YAO Linyuan, CHEN Ying, SONG Fei, *et al.* Delay-aware controller placement for fast response in software-defined network[J]. *Journal of Electronics & Information Technology*, 2014, 36(12): 2802–2808. doi: 10.3724/SP.J.1146.2014.00211.
- [13] XIAO P, LI Z Y, GUO S, *et al.* Self-adaptive SDN controller placement for wide area networks[J]. *Frontiers of Information Technology & Electronic Engineering*, 2016, 17(7): 620–633. doi: 10.1631/fitee.1500350.
- [14] LIU J and XIE R. Reliability-based controller placement algorithm in software defined networking[J]. *Computer Science and Information Systems*, 2016, 13(2): 547–560. doi: 10.2298/csis160225014l.
- [15] The internet topology zoo [EB/OL]. <http://www.topology-zoo.org>. 2013.
- [16] 刘彩霞, 卢干强, 汤红波, 等. 一种基于 Viterbi 算法的虚拟网络功能自适应部署方法[J]. 电子与信息学报, 2016, 38(11): 2922–2930. doi: 10.11999/JEIT160045.
- LIU Caixia, LU Ganqiang, TANG Hongbo, *et al.* Adaptive deployment method for virtualized network function based on viterbi algorithm[J]. *Journal of Electronics & Information Technology*, 2016, 38(11): 2922–2930. doi: 10.11999/JEIT160045.
- [17] 汤红波, 袁泉, 卢干强, 等. 一种支持节点分割的 vEPC 虚拟网络功能部署模型[J]. 电子与信息学报, 2017, 39(3): 546–553. doi: 10.11999/JEIT160507.
- TANG Hongbo, YUAN Quan, LU Ganqiang, *et al.* A model for virtualized network function deployment based on node-splitting in vEPC[J]. *Journal of Electronics & Information Technology*, 2017, 39(3): 546–553. doi: 10.11999/JEIT160507.
- 史久根: 男, 1963 年生, 副教授, 研究方向为嵌入式系统、计算机网络和无线传感器网络.
- 郝伟: 男, 1992 年生, 硕士生, 研究方向为软件定义网络、网络负载均衡和嵌入式系统.
- 贾坤荣: 女, 1994 年生, 硕士生, 研究方向为软件定义网络、网络虚拟化和嵌入式系统.
- 徐颖: 女, 1993 年生, 硕士生, 研究方向为软件定义网络和嵌入式系统.