

DOI: 10.7652/xjtuxb201512013

针对控制平面连通效能优化的控制器布置方法

王文博¹, 汪斌强¹, 王志明¹, 陈飞宇¹, 刘帅²

(1. 中国人民解放军信息工程大学国家数字交换系统工程技术研究中心, 450002, 郑州;

2. 中国卫星海上测控部, 214400, 江苏江阴)

摘要: 针对控制器间通信时延大、信息交互困难和全网视角难以维护的问题,提出了一种针对控制平面连通效能优化的控制器布置方法。该方法在保证负载均衡和服务质量的基础上将控制器布置问题建模成为一个 K -center 问题,并对指标优化下的控制器布置位置和个数进行了刻画。为求解该模型,设计了相应的基于连通效能的两阶段优化算法:第一阶段通过设置节点的优先级别和搜索范围参数,在确保控制器准确布置的前提下缩小了解的搜索范围,得到了线性松弛下的搜索半径;第二阶段输入第一阶段的结果,通过不断调整指针最终确定控制器的布置位置。仿真结果表明:该方法实现了控制器的最优化部署,减少了控制器间的平均时延,方便了控制层面的信息交互,与 CCP 方法相比,在保证时延和负载均衡的同时使运算复杂度和连通效能得到了明显改善,在 OS3E 拓扑中的解搜索范围减小了 30%,不同拓扑中的平均连通效能最大减小了 15%。

关键词: 软件定义网络;控制器布置;控制平面;连通效能

中图分类号: TP393 **文献标志码:** A **文章编号:** 0253-987X(2015)12-0077-06

A Placement Strategy of Controllers for Optimizing the Connectivity of Control Plane

WANG Wenbo¹, WANG Binqiang¹, WANG Zhiming¹, CHEN Feiyu¹, LIU Shuai²

(1. China National Digital Switching System Engineering and Technological R&D Center, The PLA Information

Engineering University, Zhengzhou 450002, China; 2. China Satellite Maritime Tracking and

Control Department, Jiangyin, Jiangsu 214400, China)

Abstract: A placement strategy of controllers for optimizing the connectivity of control plan is proposed to decrease communication latency among controllers, to improve the message communication and to remain a global view of the whole network. The controller placement problem is formulated as a K -center problem based on the load balance and QoS. The number of controllers and their placements are also studied with optimization. A 2-stage optimization algorithm based on connectivity is designed to solve the model. In the first stage of the algorithm, the node priority and the search variables are set to reduce the search scope and to ensure the accurate placement, until a search diameter is achieved under linear relaxation. The results of the first stage are used as the inputs of the second stage, and the final placement of controllers is obtained through the pointer adjustment. Simulation results show that the proposed strategy realizes the optimization of controller placement, reduces the average inner-controller latency and improves the communication in the control plane. A comparison with the CCP

收稿日期: 2015-04-16。 作者简介: 王文博(1991—),男,硕士生;汪斌强(通信作者),男,教授。 基金项目: 国家重点基础研究发展计划资助项目(2012CB315901,2013CB329104);国家自然科学基金资助项目(61372121,61309019);国家高技术研究发展计划资助项目(2015AA016102,2013AA013505)。

网络出版时间: 2015-11-18 网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20151118.1615.004.html>

strategy shows that the strategy decreases the algorithm complexity and improves the connectivity of control plane with the consideration of the latency and load balance. The search scope of the solution space reduces by 30% in OS3E, and the average connectivity for different topologies increases about 15%.

Keywords: software defined networks; controller placement; control plane; connectivity

软件定义网络 (software defined networks, SDN)^[1]分离了转发层与控制层,将原本束缚在转发设备(交换机、路由器)之内的智能环节抽离到上层,使网络具备了协议标准开放、管理集中灵活和直接可编程等特点^[2]。随着网络规模的不断增大和业务需求的多元化,多控制器网络中的控制平面设计^[3]面临着新的巨大挑战。其中,控制平面的节点布置^[4]关系到服务质量和网络的可扩展性,是控制平面设计的重要方面。

近年来,控制器节点的布置方法大致分为3类。第1类方法主要针对节点到控制器的传播时延进行布置^[4],例如针对平均时延优化的 K -median 模型和针对最差时延优化的 K -center 模型^[4],这类方法将服务质量作为布置的重要考虑因素,在逻辑上认为整个控制器集合的“角色”相同,实际上,大规模网络通常分区域进行扁平控制,需要权衡其他因素进行综合考虑。第2类方法通过对可靠性的刻画,将控制器布置问题抽象成为一个 NP-hard 的最优化问题。文献[5-10]分别根据控制路径损失^[5-6]、网络容错性^[7]、交换机节点的关联性^[8]和路径失效性^[9]对分布式网络中控制平面的可靠性进行评估分析,设计并提出了相应的数学模型和布置算法,例如 1-w 贪心算法^[5]、基于模拟退火算法的布置算法^[6]、 K -Critical 算法^[10]等。第3类方法主要考虑了网络中负载的分布情况^[11-12],例如针对负载均衡的 CCP 模型^[11]和针对级联故障的布置模型^[12]等。后两类方法虽然在保证服务质量的前提下,增强了控制平面的可靠性,但是未能对多控制器之间的连通性能进行有效的描述分析。

控制节点间的连通性^[10]是连接 SDN 控制平面和数据平面的“纽带”和“桥梁”,既连接交换机和控制器,又传达了整个平面的控制信息,对方便域间信息交换、完成资源共享和保证逻辑集中的全局视角^[3]很有意义,是提升服务质量和控制平面可靠性的重要因素^[10],而目前的布置方法对该问题的描述均相对欠缺。

本文在保证通信时延和负载均衡的前提下,提出了一种针对增强控制平面连通效能的控制器布置

方法(connectivity optimized controller placement strategy, COCP strategy),并在此基础上给出了一个基于连通效能的两阶段优化算法(connectivity based 2-stage algorithm, CB2A),对指标优化下的控制器布置位置和个数进行了研究。CB2A 的第一阶段通过事先设置候选控制器集合和搜索半径缩小了问题的求解范围,用线性松弛解给出搜索半径的大致数值;第二阶段通过不断调整指针最终确定控制器的布置位置。通过在 OS3E^[13]和 topology zoo^[14]进行仿真,与 capacitated K -center 算法^[11]相比, CB2A 算法在保证节点布置准确性的基础上降低了算法的复杂度;与 CCP 方法^[11]相比,连通效能明显改善。就目前的文献调查情况分析,根据控制平面的连通效能进行控制器布置是一种新的尝试,可为控制平面设计提供新的数学模型和思路。

1 模型的建立

通过引入控制平面网络的概念,给出控制平面连通效能的定义,继而介绍目标优化下的控制器布置模型。

1.1 控制平面的连通效能

软件定义网络在逻辑上类似于“client-server”网络,交换机完全依靠接收控制器的异步信息来实现数据的转发。同时,在多控制器 SDN 架构中,例如 Onix、Hyperflow 等,控制逻辑分布在各个控制器上,共同维护控制的一致性和唯一的全局视角。

本文提出软件定义网络控制平面连通效能的意义主要体现在以下几个方面。①维护网络的全局视角。优化连通性能后控制器间能够更好的协同合作,“透明的”管理和控制网络,方便各域内的拓扑信息交换、流表安装以及节点间的隶属关系和角色的转换。②保证控制的一致性。当不同控制器隶属下的交换机需要相互通信时,控制器需要根据全网的拓扑信息进行流表安装,但是由于控制器角色和下发顺序的不同,流表到达交换机的时间可能会发生错误。优化连通效能后,控制器间能够更快地传递信息,保证了控制的一致性。③对控制平面的可靠性做了权衡。优化控制器间连通性后,控制器布置

方法将权衡交换机到控制器之间的时延和多控制器之间的时延,使得一旦某控制器发生故障,故障信息、交换机隶属和迁移信息能够更快地进行交换,促进快速恢复,减少交换机失联,以保证控制平面的可靠性。

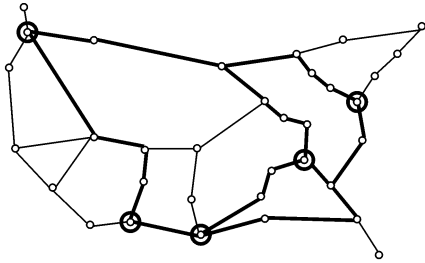
1.2 模型的描述

假设将网络建模成一个无向图 $G=(V,E)$,其中 V 是无向图中节点的集合, E 是边的集合。节点集合 $V=\{v_1, v_2, \dots, v_n\}$, 控制器集合 $C=\{c_1, c_2, \dots, c_k\}$ 。为较好地刻画控制平面的连通效能,给出以下定义。

定义 1 控制平面网络即控制器及其间最短路径组成的控制层网络,如图 1 所示。

$$G_c = (C, E') \quad (1)$$

式中: G_c 表示控制平面网络; E' 表示控制器之间最短路径集合。



●:控制器节点; ○:交换机节点; —:控制平面网络

图 1 控制器布置示意图

定义 2 连接状态矩阵 $\mathbf{X}=[x_{ij}]_{n \times n}$ 用于描述控制器布置位置、个数及 openflow 交换机的分配关系。其中,矩阵元素中 c, v 仅用于标识控制器与交换机; i, j 用于标定节点位置。当 $(x_{ij})_{ij}=1$ 时,即控制器 c 布置在第 j 个位置上,且将第 i 个交换机 v 分配给该控制器。

$$(x_{ij})_{ij} = \begin{cases} 1, & \text{if } v \text{ is assigned to } c \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

定义 3 控制平面的连通效能定义在控制平面网络上,即控制器间最短距离的倒数的平均值

$$E_{G_c} = \frac{1}{k(k-1)} \sum_{c_i \in C} \sum_{c_j \in C} 1/d(c_i, c_j) \quad (3)$$

$d(v_1, v_2)$ 是节点 v_1 与 v_2 之间的最短距离,引入定义 2 中的连接状态矩阵后,式(3)可表示为

$$E_{G_c} = \frac{1}{k(k-1)} \sum_{i \in J} \sum_{j \in J} \frac{1}{d(i, j)} \quad (4)$$

$$J = \{j \mid \sum_i x_{ij} \neq 0\}$$

式中:集合 J 即是连接状态矩阵描述下的控制器集合 C 。

用控制器节点之间连接跳数的平均距离来描述控制平面的连通效能,既能够保证逻辑集中的全网视角,又能够衡量控制器间信息传递。

当归属在不同控制器内的某两个交换机进行通信时,必需先通过控制平面的信息交互,再向各自的数据平面下发流表^[8]。这一过程产生的通信开销必然涉及到控制器节点对地理位置的描述。当衡量控制器布置对整个控制平面网络带来的通信开销时,显然定义 3 是合理的。

由式(4)可知,连通效能随着控制器布置位置不断变化。控制器与交换机的分配关系虽然不能直接改变连通效能,但是会对负载均衡和通信时延造成影响。

1.3 模型的建立

该模型在保证服务质量和负载均衡的基础上,优化控制平面的连通效能。该模型旨在网络搭建之初给出控制器的布置位置,不涉及具体的网络通信细节与服务机制。模型中采用平均时延^[4]对服务质量进行刻画。下面给出负载均衡中关于控制器负载的定义。

定义 4 控制器负载即分配交换机向控制器流请求数的总和。该模型使用 PACKET_IN 消息的数量对控制器负载进行描述^[11]。第 j 个控制器的负载可以表示为

$$L_j = \sum_{i \in V; d(i, j) \leq \epsilon} l_i x_{ij} \quad (5)$$

式中: l_i 是第 i 个交换机向控制器的流请求数; x_{ij} 用于描述控制器与交换机的分配关系。

为了保证交换机和其所属控制器的传播时延,模型对其间的跳数作了约束,要求控制器到其管辖的交换机的最大跳数不得超过 ϵ 。这样能够使控制器位于一簇交换机的中心,同时保证控制器尽量均匀地分布在拓扑以内。 ϵ 的取值与拓扑的半径和控制器布置个数有关,这将在后续算法中给予进一步说明。

由式(5)可以看出,控制器交换机的分配关系虽然不能直接影响控制层面的连通效能(E_{G_c} 由控制器布置位置直接影响),但是与时延和负载有关,将影响控制器布置位置,进而对连通效能产生影响。

优化控制平面连通效能下的模型为

$$\max E_{G_c} \quad (6)$$

$$\text{s. t. } \sum_j x_{ij} = 1, \forall i \quad (7)$$

$$L_j \leq C_j, \forall j \quad (8)$$

$$x_{ij} \in \{0, 1\}, \forall i, j \quad (9)$$

式(6)是模型的优化目标,即控制平面的连通效能最大值;式(7)~式(9)是模型的约束条件。式(7)保证了将全网内的所有交换机分配给控制器;式(8)中 C_j 为第 j 台控制器的容量,它保证了每台控制器的负载均不超过其负载上限,以免引发控制器故障。同时,式(8)还隐含了定义4中的约束条件,使得交换机分配服从就近原则,以调整传播时延来确保服务质量。式(9)对连接状态矩阵的元素进行了约束,保证元素的值非0即1。

至此,模型已经描述完毕。本文将模型抽象成为一个复杂的0-1整数规划问题。该问题在刻画交换机分配的基础上,对控制平面的连通效能进行描述。为了求解该问题,给出一个改进的两阶段算法。

2 算法分析

模型是对经典 p -center 问题的修正^[11,15],是一个典型的 NP-hard 问题。其中,控制器布置个数和搜索半径相互制约。文献[15]给出了一般情况下(拓扑节点没有重要性区分,即无控制器节点与交换机节点之分)的两阶段算法,并描述了 p -center 问题的两个子问题;文献[11]中的 capacitated K -center 算法对 p -center 算法进行了改进,缩小了搜索半径的变化范围。本文在上述两算法的基础上提出 CB2A 算法,针对连通效能的优化指标,对控制器集合的搜索范围做了调整,相比之前算法进一步缩小了算法的复杂度,具体说明详见仿真部分。

算法的第一阶段根据拓扑的规模对 ϵ 进行限制。这一过程是一个典型的二分法求解过程,指针首先指向搜索范围的中间点,然后通过不断调整指针指向,对原问题进行松弛条件下的求解,确定 ϵ 的大概范围。其中,线性松弛过程主要体现在对式(9)中的二进制取值松弛为小数。基于控制器布置个数 K 和 ϵ 的相互影响,当选定 ϵ 时,比较实际布置个数 h_{num} 与预计布置个数 K 之间的关系,如果 h_{num} 较大,说明 ϵ 束缚了交换机节点的“运动半径”,应将 ϵ 增大;反之,则减小 ϵ 。

算法的第二阶段输入第一阶段得到的搜索起点,通过修正指针位置,最终确定控制器布置方法。算法的具体流程如下。

阶段1有6个步骤。

步骤1 求出拓扑中所有源、目的节点的最短距离并按照降序排列,存入数组 q ;同时,找到搜索下限 $l=0$,并取 q 的长度赋值给搜索上限 ϵ 。

步骤2 计算 ϵ 的搜索起点, $m=(u+l)/2$,同时将

$\lfloor m \rfloor$ 的长度赋值给 ϵ 。

步骤3 确定控制器布置的候选节点。先将控制器节点按照度降序排列,取集合的前 w (w 属于0到1)个节点组成候选布置位置序列。

步骤4 代入候选控制器集合,使用线性规划方法求解 ϵ 下最优化问题的线性松弛解,得到此时的控制器布置个数 h_{num} 。如果 $h_{\text{num}} > K$, $l=m+1$;否则, $u=m$ 。

步骤5 计算上限与下限的关系,如果 $u > l$,转到步骤2;否则,转到步骤6。

步骤6 阶段1得到距离 ϵ 的粗略取值,需要重新对搜索起点赋值,即将 l 的值赋给指针 p 。

阶段2有2个步骤。

步骤7 求解此时控制器布置个数。根据 p 和候选控制器集合求解该条件下的线性解,也即控制器放置个数 h_{num} 。

步骤8 对指针进行调整。如果 $h_{\text{num}} > K$,将指针加1,也即 $p=p+1$ 。随后跳转至步骤7;否则结束阶段2,得到最终的控制器布置方法。

值得注意的是,阶段1的步骤3中,CB2A 算法改进了原算法中的无差别搜索,将节点的度作为重要的考虑因素。先按照节点度的大小对所有节点进行降序排列,然后设计节点搜索范围参数 w ,以控制算法计算的搜索范围。其中, w 是需要搜索的节点数占节点总数的百分比,按照度优先原则选取,即步骤3。

使用度作为优先级别划分的原因是在考虑连通性能时,度越大的节点对连通各个节点的贡献越大,能够使控制节点更容易与其他节点通信,同时,度大的节点往往处在一簇节点的中心。这一特征在 SDN 中意义明显,能够对交换机与控制器之间的连接跳数进行约束。明显地,拓扑末端的叶子节点(度为1),不可能作为中心节点布置控制器。后续的仿真结果也证实了该想法,与没有进行控制器节点约束的 capacitated K -center 算法^[11]相比,当选定合适的 w 和控制器布置个数 K ,两者的控制器布置位置基本重合,说明度大的节点被选作控制器布置节点的可能性很大。

3 仿真与分析

首先说明仿真环境和仿真设置,然后分析仿真结果:①对比了 CB2A 算法和 capacitated K -center 算法在同一优化条件下的控制器布置位置的重合度,说明了模型中节点搜索范围参数 w 设置的合理

性;②在实验拓扑中采用不同控制器布置方法,研究了控制平面连通效能的变化趋势;③研究了控制器布置个数与连通效能之间的关系,发现了一些与想象不同但合乎情理的实验结果。

3.1 仿真环境搭建

(1)关于拓扑。为了使实验更具说服力,实验中的拓扑均取自 OS3E 和 topology zoo。其中,OS3E 网络抽象自美国的实际骨干网络,来自 Internet2 项目,由美国 120 多所大学、研究机构和公司为建设下一代互联网共同推进。Topology zoo 由澳大利亚政府和阿德雷德大学共同推进,抽象并收集了全球超过 250 个公共拓扑,在学界认可度高。

(2)关于仿真环境搭建和仿真步骤。为简化实验操作,本文类比文献[11]中关于平均控制器负载的实验设置。用 PACKET_IN 消息的数量作为衡量控制器负载的关键因素,平均流产生速率服从 280 KB/s 到 320 KB/s 的均匀分布。同时考虑到控制器挂载服务器之间的相互差异,对控制器容量做了差异化设置,即设为 8~11 MB。由于问题中对负载均衡程度没有衡量,仿真中只要不超过控制器容量即满足网络搭建要求。同时,实验中使用跳数作为衡量网络半径的标准,式中的节点距离一律用跳数进行代替。

(3)CB2A 中的线性规划的求解采用 IBM CPLEX Optimizer^[16]进行求解。实验环境为 Intel Core i7 CPU 2.67 Hz,PC 为 RAM 8 GB,整个实验平台在 C++ 语言下进行搭建。具体步骤是,通过 C++ 进行拓扑搭建和输入数据的设置,调用 CPLEX 中的 Concert Technology 进行模型建立,使用其内核中的优化算法求解式(6)中的优化目标来完成整个实验过程。

在上述设置下,实验分别对算法性能、控制平面连通性能和控制器布置个数进行了研究分析。其中,控制平面连通性能采用式(4)进行计算,所得解经验证均能满足第二阶段的约束条件,保证了解的正确性。

3.2 实验结果分析

实验分别在 OS3E 网络上运行 CB2A 和 capacitated K -center 算法,此时保证优化目标相同。调整节点搜索范围参数 w 改变控制器布置个数,统计不同情况下控制器布置位置的重合程度以说明 CB2A 算法设计流程的合理性。其中,布置位置的重合程度指在两种算法下,布置在相同物理节点的控制器个数占控制器布置总个数的百分比,具体结

果如图 2 所示。维持 w 在较低值时,随着控制器布置个数增多,重合度曲线呈现波动。这种现象出现的原因是,候选节点单纯按照度的大小进行降序排列,对于度相同的节点仅随机排列,并没有进一步划分优先级,导致某些重要节点始终无法进入搜索范围。随着控制器布置个数增多,对重合度造成的影响增大,所以重合度曲线波动下滑。随着 w 值不断增大,重合度曲线向上漂移,这是因为算法搜索范围的增加带来了布置位置重合度的增加。当 $w = 0.7$,两种算法搜索结果完全相同,此时相当于设计算法的解搜索范围减小了 30%。

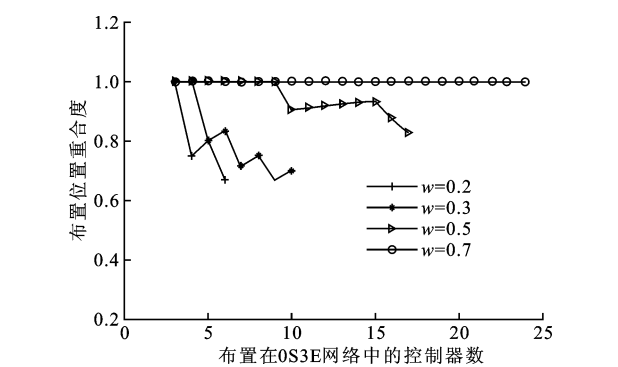


图 2 两种算法下的控制器位置重合度

事实上,与节点总数相比,控制器布置个数往往很少^[6]。当控制器布置个数较少时,即使 w 取值较小,也能够满足控制器布置的准确性。可见,CB2A 算法的设计是合理有效的。同时,算法的复杂度主要体现在对线性规划的求解上,原算法中该整数规划问题的时间复杂度为 $O(2^n)$,本文算法中由于 w 的引入使其复杂度下降为 $O(2^{wn})$ 。

在 OS3E 和 topology zoo 中的若干拓扑上重做实验,其拓扑信息见表 1。

表 1 实验拓扑数据				
拓扑名称	OS3E	Garr201104	Columbus	Tw
节点数	34	59	70	76
链路数	42	83	85	118

在不同拓扑中,对相同条件下的 CCP 布置方法^[11]和本文提出的 COCP 布置方法进行仿真,结果如图 3 所示,横轴是控制器个数 k 占拓扑中节点总数 n 的比例(k/n),纵轴是控制平面的连通效能 E 。实验布置的控制器个数不超过节点总数的 20%。

实验结果显示,COCP 布置方法明显提升了控制平面的连通效能,在 4 个不同拓扑中,其平均值最大提升了 15%。在同一拓扑中,连通效能值的大小

与控制器布置个数无明显对应关系。这一结果与事先的预测结论差距较大:凭借经验,控制器布置个数越多,连通性能越好。由于控制平面网络(定义1)是原拓扑的一个子图,其节点、链路的不确定性导致定义在该网络上的连通效能与控制器布置个数并无直接联系。同时,随着控制器布置个数增多,两种布置方法差异越来越不明显。由于控制器布置个数增多,两种布置方法的控制器位置重合概率增大。

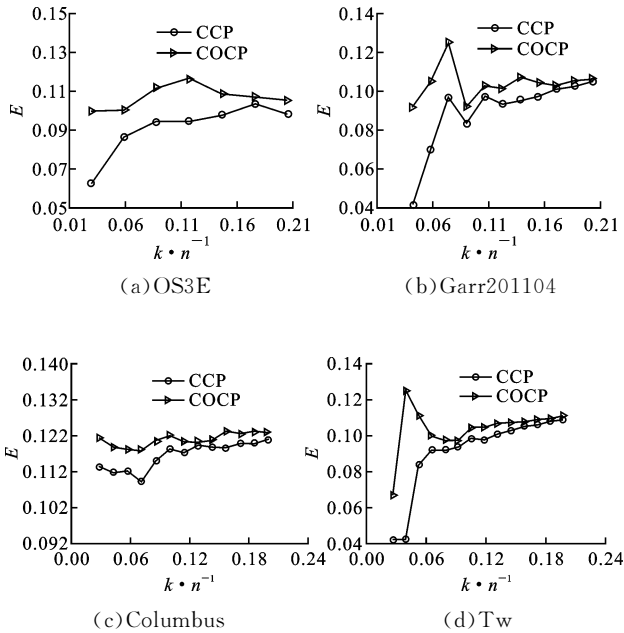


图3 不同拓朴中控制器布置方法比较

4 结论

本文针对现存模型无法对控制平面连通性进行描述的不足,定义了SDN控制平面的连通效能指标,提出一种针对增强控制平面连通效能的控制器布置方法——COCP方法,对指标优化下的控制器布置位置和个数进行了研究。该方法在保证连通性能和负载均衡的基础上对控制器进行布置。设计了一种两阶段算法CB2A,通过调整节点搜索范围参数改善了现有capacitated K -center算法的运算复杂度。在OS3E和topology zoo中的公共拓朴上分别进行仿真实验,证明了算法设计的合理性和COCP方法较CCP方法在提高控制平面连通效能方面的明显优势。最后,研究了连通效能与控制器布置个数、拓朴差异之间的关系。本文对控制平面的连通性能进行描述,可为控制平面的设计和实现提供新的数学模型和努力方向。

参考文献:

- [1] 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.
ZUO Qingyun, CHEN Ming, ZHAO Guangsong, et al. Research on OpenFlow-based SDN technologies [J]. Journal of Software, 2013, 24(5): 1078-1097.
- [2] 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.
- [3] LEVIN D, WUNDSAM A, HELLER B, et al. Logically centralized?: state distribution trade-offs in software defined networks [C]// Proceedings of the First Workshop on Hot Topics in Software Defined Networks. New York, USA: ACM, 2012: 1-6.
- [4] HELLER B, SHERWOOD R, MCKEOWN N. The controller placement problem [C]// Proceedings of the First Workshop on Hot Topics in Software Defined Networks. New York, USA: ACM, 2012: 7-12.
- [5] HU Yannan, WANG Wendong, GONG Xiangyang, et al. On the placement of controllers in software-defined networks [J]. The Journal of China Universities of Posts and Telecommunications, 2012, 19(S2): 92-97.
- [6] YANNAN H, WENDONG W, GONG X, et al. Reliability-aware controller placement for software-defined networks [C]// Proceedings of the 2013 IEEE International Symposium on Integrated Network Management. Piscataway, NJ, USA: IEEE, 2013: 672-675.
- [7] ROS F J, RUIZ P M. Five nines of southbound reliability in software-defined networks [C]// Proceedings of the Third Workshop on Hot Topics in Software Defined Networks. New York, USA: ACM, 2014: 1-6.
- [8] HOCK D, GEBERT S, HARTMANN M H, et al. POCO-framework for Pareto-optimal resilient controller placement in SDN-based core networks [C]// Proceedings of the 2014 IEEE Network Operations and Management Symposium. Piscataway, NJ, USA: IEEE, 2014: 1-2.
- [9] ZHANG Ying, BEHESHTI N, TATIPAMULA M. On resilience of split-architecture networks [C]// Proceedings of the 2011 IEEE Global Telecommunications Conference. Piscataway, NJ, USA: IEEE, 2011: 1-6.
- [10] YURY J, CRISTINA C, AURELIO J G. On the controller placement for designing a distributed SDN control plane [C]// Proceedings of the 2014 IFTN Networking Conference. Piscataway, NJ, USA: IEEE, 2014: 1-9.

(下转第143页)