

SDN 中基于可靠性评估的多控制器均衡部署策略

胡涛, 张建辉, 马腾, 赵伟

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

摘要: 针对分布式软件定义网络中控制平面可靠性差和控制器负载不均衡问题, 提出一种基于可靠性评估的多控制器均衡部署策略。首先, 通过综合权衡节点效能和路径质量评估节点可靠性, 并在网络中设定均衡因子, 优化控制器部署位置; 然后, 改进 k -center 聚类方法, 引入冗余函数, 根据节点吸引度和控制器负载均衡率完成交换机分配, 实现合理的 SDN 子域规划。仿真结果表明, 与现有策略相比, 所需控制器数量平均减少 22.1%, 控制平面弹性明显增强, 控制器负载均衡性能得到较大提升。

关键词: 软件定义网络; 控制器; 可靠性; 负载均衡

中图分类号: TP393

文献标识码: A

Multi-controller balancing deployment strategy based on reliability evaluation in SDN

HU Tao, ZHANG Jian-hui, MA Teng, ZHAO Wei

(National Digital Switching System Engineering R&D Center, Zhengzhou 450002, China)

Abstract: The control plane has low reliability and controller loads were unbalanced in distributed software defined networks, and a multi-controller balancing deployment strategy based on reliability evaluation was proposed. Firstly, node reliability was evaluated by weighting node efficiency and path quality, optimizing the controller location with balancing factor. Then based on improved k -center clustering, the redundant functions was introduced and the switch allocation was completed according to node attractiveness degree and controller load balancing rate, achieving a reasonable SDN sub-domain planning. The simulations show that compared with the existing strategy the number of required controllers is reduced by an average of 22.1%. The control plane elasticity is enhanced, and the controller load balancing performance has been improved significantly.

Key words: software defined networking, controller, reliability, load balancing

1 引言

软件定义网络 (SDN, software defined network)^[1] 作为一种新型的网络体系架构, 实现了控制平面和数据平面的完全解耦, 将束缚在转发设备内的控制功能抽象到上层, 具有集中式控制和网络可编程的特点, 受到人们的广泛关注。

随着网络规模的不断扩展和流量业务的迅速增长^[2], 集中式单控制器已无法满足现有的网络需求。因此, 为了提高控制器可扩展性, 避免单点失效问题^[3], 研究人员提出在原有集中式控制的基础上建立分布式 SDN, 通过在网络中部署多控制器, 将整个网络划分为多个 SDN 子域。在每个子域内, 控制器仅负责管理域内交换机, 形成集中式控制关

收稿日期: 2017-01-13; 修回日期: 2017-06-07

基金项目: 国家自然科学基金资助项目 (No.61521003, No.61372121, No.61572519); 国家科技支撑计划基金资助项目 (No.2014BAH30B01); 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 of China (No.61521003, No.61372121, No.61572519), The National Key Technology R&D Program of China (No.2014BAH30B01), The National High Technology Research and Development Program of China (863 Program) (No.2015AA016102, No.2013AA013505), Key Scientific and Technological Project of Henan Province (No.162102210034)

系。多控制器分布式部署在各个子域,共同构建逻辑上集中的全局网络视图来管理整个网络,最终形成逻辑上集中、但物理上分布的 SDN 多控制器网络架构(HyperFlow^[4]、Onix^[5]、Kandoo^[6])。

虽然分布式 SDN 架构的引入增强了网络的灵活性,但同时也给多控制器的部署设计带来了巨大挑战。在实际的网络拓扑中,节点和链路不可能一直维持正常状态^[7],外界物理破坏(链路中断)或传输拥塞会造成链路故障和节点孤立^[8];其次,流量的不均衡分布^[9]导致网络中控制器负载差异较大。因此,对于给定的网络拓扑,如何选择合理的控制器部署位置来构建 SDN 子域、提升控制平面可靠性、均衡控制器负载,是 SDN 控制平面的重要研究课题。

近年来,关于控制器部署的方案可以总结为 2 种。第一种方案^[10-15]考虑网络中时延、流量、跳数等因素,通过多目标混合优化求解控制器部署位置。这类方案以提升通信交互质量,降低响应时间为部署目标,但忽略了设备连通性对网络状态的影响,缺乏路径保护机制,一旦控制器发生故障,将会导致通信中断和跨域交互问题。第二种方案^[16-21]考虑网络的可靠性指标,以节点和链路的可靠性为度量完成控制器部署,不足之处在于对交换机分配及动态流量缺乏考虑,容易造成控制器负载不均衡,降低了网络吞吐量。

针对上述 2 种研究方案中存在的问题,本文提出一种基于可靠性评估的多控制器均衡部署(MCBD, multi-controller balancing deployment)策略,同时考虑控制平面可靠性和控制器负载均衡 2 个方面因素,在多控制器部署架构的基础上,通过设定控制器部署的相关指标,包括节点效能、路径质量、节点吸引力、控制器负载均衡率,完成控制器部署位置选取和 SDN 子域规划。本文的主要贡献和创新工作总结如下。

1) 在控制平面可靠性方面,设计了控制器位置选取算法,综合权衡节点效能和路径质量对网络中节点的可靠性进行评估,引入权值参数调节两者比重,同时设定均衡因子,避免控制器分布过于集中或分散,优化了控制器部署位置。

2) 在分布式子域构建方面,基于已选取的控制器部署位置,设计了 SDN 子域规划算法,通过改进 k -center 聚类方法^[22],根据节点吸引度和控制器负载均衡率完成交换机分配,设置冗余函数应对流

量突发,实现合理的 SDN 子域规划和多控制器负载均衡。

3) 综合可靠性和负载均衡评价指标,同时与现有的代表性部署策略进行对比,并在多种网络场景下仿真实验,验证 M CBD 策略性能。

2 相关工作

在控制器部署问题研究过程中,现有的研究方案主要从 2 个方面进行考虑。

1) 考虑网络中各类代价(时延、流量、跳数等)因素作为控制器部署依据。Heller 等^[10]首次提出控制器部署问题,注重网络中平均时延和最大时延,将交换机和控制器的连接问题转化为设备位置问题,构建相应数学模型,确定控制器最佳部署状态。Barim^[11]基于网络条件动态确定控制器的数量和位置,综合跳数和时延来寻求最优组合方案。Saallahi^[12]针对控制器部署问题提出相应的数学模型,详细分析了控制器最优部署数量、位置、控制器类型以及网络元素之间的相互联系。Adlen 等^[13]提出使用博弈论解决 SDN 控制器最优部署,在时延、通信开销、控制器负载均衡这 3 个目标之间进行博弈考虑,找到均衡度量、优化控制器部署数量和位置。张联镇^[14]提出基于时延优化的多控制器部署方案,通过聚类方法对控制器进行初步部署,向网络输入数据流,根据蜂群算法思想,考虑路径请求时延,流表下发时延,流建立时延,以最小化同步时延开销为目标优化控制器部署。Zhao 等^[15]对控制器在数据通信网络(DCN)中可用性进行量化评估,主要考量了可扩展性、拥塞概率、带宽利用率和故障恢复时间等网络参数,验证了基于 SDN 的 DCN 控制架构性能。

2) 考虑控制平面的可靠性部署。为了解决单控制器部署性能的不足和单点失效问题,相关学者提出了扁平化多控制器部署方案^[4,5]和垂直化多控制器部署方案^[6]。在此基础上,Hu 等^[16]引入可靠控制路径百分比期望值来评价 SDN 的可靠性,并从数学角度证明 RCP(reliability-aware controller)为 NP-Hard 问题,使用贪婪算法进行求解。Müller 等^[17]提出了 Survivor 模型,主要考虑 3 个方面:路径连通性、控制器容量和快速恢复性。为了最大化路径连通性,Survivor 选择节点不相交路径最多的节点作为控制器部署位置,同时设定备份控制器来应对网络故障问题。Tao 等^[18]设计了控制器动态部署机

制来增强控制平面的稳定性，将控制器部署阐述为交换机和控制器之间的稳定匹配和转移问题，设计了分层式两段算法，从匹配原理和联合博弈角度解决控制器部署位置和数量。Beheshti 等^[19]提出一种弹性度量标准来连接控制器与交换机，构建路由树，当检测到输出接口失效时，一个受保护的交换机能立即使用备份接口重连控制器，实现失效流量的快速转移。Aguado^[20]提出将控制器部署与网络功能虚拟化 (NFV) 相结合，创建了 SDN/NFV 协调架构。控制器作为一个虚拟化网络功能 (VNF) 被移至云端，可以在数据中心网络中进行快速切换，有效地减少了故障状况下控制器备份和维护停机时间。Gianfranco 等^[21]探讨了故障模式下控制器部署对于 SDN 性能的影响，提出一种二层建模方法，顶层捕获网络的结构属性，下层是不同网元/子系统的可靠性特征。研究表明，控制器位置选择对于 SDN 可靠性具有重要影响，同时过量控制器部署也会增加不必要的运营商成本。

3 多控制器部署模型

本节对现有的多控制器部署问题进行重述，建立相应的数学模型，提出基于可靠性评估的多控制器均衡部署策略，并对相关指标进行定义与阐述。

3.1 问题重述

考虑控制器作为一种网络部件可以安装在各类服务器或主机上，在此采用带内通信模式，将控制器的部署位置定义为网络中交换机节点位置。在分布式网络场景中，为了简化分析，降低控制器安装代价，假设在每个控制器节点上只需要部署一个控制器即能满足子域内交换机的流处理需求。因此，分布式 SDN 中多控制器部署问题主要包括选取可靠的控制器部署节点以及控制器联合交换机构建 SDN 子域。1) 在选取控制器部署节点时，所

选取的节点本身应当具有低故障率和高流请求速率 (以 Packet-in 数据分组为主)，且流请求速率越高，该节点上控制器的数据分组处理性能收益越大 (因为控制器到该节点上交换机的跳数和时延可以忽略不计)。由于交换机和控制器通过物理链路进行交互，还必须考虑和控制器节点相连接的链路状态 (如较短的链路长度、低链路失效概率、高链路带宽)。2) 控制器部署位置确定后，构建 SDN 子域时，合理分配交换机，保证各子域内交换机部署数量和控制器负载的均衡性。

如图 1 所示，需要在具有 10 个节点的网络中选取控制器部署位置，考虑链路带宽和节点失效概率都相同。其中，灰色节点表示控制器部署位置，网络被划分为 2 个子域，分别用圆形和方形表示，节点附近的数字表示该节点上交换机的流请求速率。在图 1(a)中控制器部署在节点 1 和节点 8，2 个子域分别含有 4 个和 6 个交换机。当节点 1 和节点 2 之间的链路发生失效时，导致节点 1 成为孤立节点，控制器失效，严重损害网络的通信性能。出于负载均衡考虑，对控制器部署位置进行调整，如图 1(b)所示，由于节点 3 相较于其他节点具有更高的流请求速率，因此选择节点 3 作为控制器部署节点，同时调整节点 4 的分配，可以看出，调整后网络中控制器的负载分布更加均衡，且控制平面连通性得到较大提升。

3.2 模型构建

在上述问题分析的基础上，本文提出了基于可靠性评估的多控制器均衡部署 (MCBD) 策略，解决思路如下。首先，在全局网络拓扑中基于节点和链路特征优化控制器部署位置来最大化控制平面可靠性。然后，根据网络中流请求分布状况进行交换机分配，实现合理的 SDN 子域划分。

下面，对 MCB D 策略进行数学建模，并对网

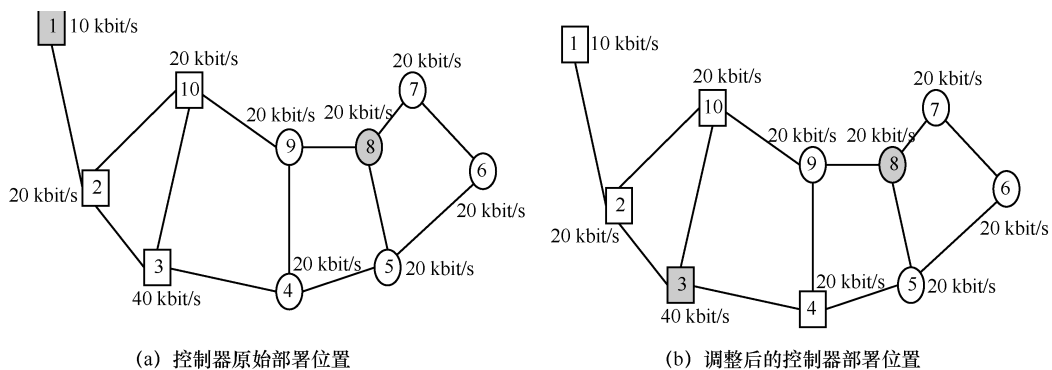


图 1 控制器部署位置选择

络故障状态下的相关指标进行定义与阐述。

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, 具体的网络参数及说明如表 1 所示, x_{ij} 和 $v_i^{c_j}$ 为二进制变量。 x_{ij} 表示相邻节点连通性, 如式(1)所示, $x_{ij}=1$ 表示相邻节点 v_i 和 v_j 连通。 $v_i^{c_j}$ 表示控制器部署位置, 如式(2)所示, $v_i^{c_j}=1$ 表示第 j 个控制器部署在节点 v_i 上。

$$x_{ij} = \begin{cases} 1, & v_i \text{ 和 } v_j \text{ 连通} \\ 0, & \text{其他} \end{cases} \quad (1)$$

$$v_i^{c_j} = \begin{cases} 1, & \text{第 } j \text{ 个控制器部署在节点 } v_i \text{ 上} \\ 0, & \text{其他} \end{cases} \quad (2)$$

表 1 网络参数及说明

网络参数	说明
$V = \{v_1, v_2, \dots, v_n\}$	网络节点
$S = \{s_1, s_2, \dots, s_n\}$	节点上交换机
λ_i	交换机 s_i 的流请求速率
$E = \{e_{ij}\}$	节点间链路
x_{ij}	相邻节点 v_i 和 v_j 之间链路连通性
d_{ij}	相邻节点 v_i 和 v_j 之间链路距离
ω_{ij}	相邻节点 v_i 和 v_j 之间链路带宽
l_{mn}	任意节点 v_m 和 v_n 之间最短路径长度
h_{mn}	任意节点 v_m 和 v_n 之间最小跳数
$C = \{c_1, c_2, \dots, c_k\}$	控制器集合
CS_j	控制器 c_j 管理的交换机集合
$Domain_j$	控制器 c_j 联合交换机构建的 SDN 子域
n_j	子域 $Domain_j$ 包含的交换机个数
Ω	控制器处理容量
$v_i^{c_j}$	第 j 个控制器部署在节点 v_i 上

基于上述参数设定, 定义以下指标。

定义 1 节点接近度 (NC)。节点接近度是指网络中相邻节点间距离倒数之和的均值。如式(3)所示, 其中, n 为节点总数, 可以看出, 节点间距离越短, 则节点接近度越高。

$$NC = \sum_{i \neq j} \frac{1}{n(n-1)d_{ij}} \quad (3)$$

定义 2 节点可靠度 (R_i)。是指节点 v_i 与其他节点的平均接近程度, 如式(4)所示, 表示节点 v_i 到其他节点距离倒数之和的均值。

$$R_i = \sum_{j=1, j \neq i}^n \frac{1}{(n-1)d_{ij}} \quad (4)$$

定理 1 节点接近度是所有节点可靠度的算术平均值。

证明 网络中节点接近度计算如式(3)所示, 对其进行分解, 如式(5)所示。

$$NC = \sum_{i \neq j} \frac{1}{n(n-1)d_{ij}} = \sum_{i=1}^n \left(\sum_{j=1, j \neq i}^n \frac{1}{n(n-1)d_{ij}} \right) \quad (5)$$

因此, 对于指定的节点 v_i 有

$$NC = \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1, j \neq i}^n \frac{1}{(n-1)d_{ij}} \right) = \frac{1}{n} \sum_{i=1}^n (R_i) \quad (6)$$

通过式(6)可以看出, 节点接近度是所有节点可靠度的算术平均值, 因此, 原命题得证。节点可靠度越高, 则该节点的连通程度越好, 数据传输越稳定。

定义 3 节点效能。节点效能是对网络中节点性能的综合评定, 由链路数量比 (表示该节点连接链路占网络中所有链路数量的比重)、节点可靠度和节点上交换机请求速率 3 个部分构成, 设 v_i 的节点效能为 NE_i , 如式(7)所示。网络中控制器部署节点效能为 NE , 如式(8)所示。

$$NE_i = \left(\frac{\sum_{j=1}^n x_{ij}}{\sum_{i=1, j=1, i \neq j}^n x_{ij}} \right) R_i \lambda_i \quad (7)$$

$$NE = \sum_k NE_k \quad (8)$$

定义 4 节点吸引力。当控制器位置确定后, 需要为控制器分配交换机, 引入节点吸引力, 表示网络中交换机节点对控制器节点的吸引程度。对于控制器 c_j , 节点 v_i 的吸引力 $A_{v_i}^{c_j}$ 如式(9)所示, 其中, l_{ij} 表示节点 v_i 到控制器 c_j 的最短距离, 若节点 v_i 本身是控制器部署节点, 则 $A_{v_i}^{c_j} = \infty$ 。对于交换机节点 v_i 而言, 流请求速率越小, 距控制器 c_j 的距离越短, 则节点吸引力 $A_{v_i}^{c_j}$ 越大, v_i 越趋向于分配给控制器 c_j 。

$$A_{v_i}^{c_j} = \begin{cases} \frac{1}{(\lambda_i l_{ij})}, & \text{节点 } v_i \text{ 不是 } c_j \text{ 的控制节点} \\ \infty, & \text{节点 } v_i \text{ 是 } c_j \text{ 的部署节点} \end{cases} \quad (9)$$

定义 5 链路中断概率 ($P_{\text{break}}(e_{ij})$)。本文中中断是指相邻节点间链路发生物理中断。 $P_{\text{break}}(e_{ij})$ 与物理链路的长度有关, 链路长度越长, 则发生中断的概

率越大。 $P_{\text{break}}(e_{ij})$ 计算式如式(10)所示, 其中, p_{unit} 表示单位长度链路的失效概率。 $P_{\text{break}}(e_{ij})$ 的取值范围为 $[0,1]$, 当 $x_{ij} = 0$ 或 $p_{\text{unit}} = 1$, 有 $P_{\text{break}}(e_{ij}) = 1$ 。

$$P_{\text{break}}(e_{ij}) = \begin{cases} \frac{(d_{ij} p_{\text{unit}})}{x_{ij}}, & x_{ij} = 1 \\ 1, & x_{ij} = 0 \text{ 或 } p_{\text{unit}} = 1 \end{cases} \quad (10)$$

定义 6 链路拥塞概率 ($P_{\text{congest}}(e_{ij})$)。流量传输导致的拥塞会造成链路失效, $P_{\text{congest}}(e_{ij})$ 与链路带宽和链路上传输流量有关。链路上流量越少, 带宽越大则拥塞失效概率越小, 如式(11)所示, ρ_{ij} 为链路 e_{ij} 的流向因子, $\rho_{ij} = 1$ 表示流量 λ_i 由源节点 v_i 传输至目的节点 v_j , 否则 $\rho_{ij} = 0$ 。 $P_{\text{congest}}(e_{ij})$ 的取值范围为 $[0,1]$, 当 $x_{ij} = 0$ 时, 有 $P_{\text{congest}}(e_{ij}) = 1$ 。

$$P_{\text{congest}}(e_{ij}) = \begin{cases} \frac{(\rho_{ij} \lambda_i + \rho_{ji} \lambda_j)}{(x_{ij} \omega_{ij})}, & x_{ij} = 1 \\ 1, & x_{ij} = 0 \end{cases} \quad (11)$$

定义 7 路径质量。在给定的网络拓扑中, 根据链路距离 d_{ij} , 链路容量 ω_{ij} 和链路失效概率 $P_{\text{failure}}(e_{ij})$ 对链路进行评判, $P_{\text{failure}}(e_{ij}) = P_{\text{break}}(e_{ij}) + P_{\text{congest}}(e_{ij})$, 其取值范围为 $[0,1]$ 。通过对定义 5 和定义 6 进行分析可知, $P_{\text{failure}}(e_{ij})$ 与 d_{ij} 和 ω_{ij} 有关, 因此, 定义链路 e_{ij} 的路径质量为 PQ_{ij} , 如式(12)所示。

$$PQ_{ij} = 1 - (P_{\text{congest}}(v_i, v_j) + P_{\text{break}}(v_i, v_j)) \quad (12)$$

$$PQ_i = \sum_{j=1, j \neq i}^n PQ_{ij} \quad (13)$$

$$PQ = \sum_k PQ_k \quad (14)$$

$$PQ_{ij} = PQ_{ji}, \forall i, j \in n \quad (15)$$

$$PQ_{ii} = 0, \forall i \in n \quad (16)$$

其中, 式(13)表示节点 v_i 所连链路的路径质量和, 式(14)表示与控制器节点相连链路的路径质量之和, 式(15)表示网络中链路具有无向性, 式(16)表示不存在节点回环路径。

定义 8 控制器负载均衡率。在计算控制器负载时, 需要同时考虑交换机个数和流请求速率 2 个方面因素, 则全局网络中控制器负载均衡率 (CLB) 如式(17)所示, 其中, η_j 表示第 j 个子域的

交换机个数, λ_i 为交换机的流请求速率。在理想状态下, 即交换机请求速率相同, 每个控制器所连接的交换机数量也相同时, $CLB^* = \frac{1}{k}$ 。

$$CLB = \sum_{j=1}^k \left[\frac{\eta_j}{n} \left(\frac{\sum_{m=1}^{\eta_j} \lambda_m}{\sum_{i=1}^n \lambda_i} \right) \right] \quad (17)$$

基于上述定义, 在多控制器部署过程中, 需要同时考虑节点效能、路径质量和控制器负载均衡率这 3 个方面因素, 其中, 节点效能和路径质量保证可靠的控制器位置选取, 控制器负载均衡率保证合理的子域规划。设 MCB D 策略的目标函数为 *Object*, 如式(18)所示, 目的是通过寻求合理的控制器部署方法使 PQ 、 NE 、 CLB 达到最大。

$$\text{Object: } \max[PQ, NE, CLB] \quad (18)$$

$$\text{s.t.: } \sum_{i=1}^n \lambda_i \leq k \Omega \quad (19)$$

$$\sum_{i=1}^{\eta_j} \lambda_i \leq \Omega \quad (20)$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} > 0 \quad (21)$$

其中, 式(19)表示 k 个控制器已经能满足网络中所有交换机的流请求速率需求, 式(20)保证单个控制器所要处理的流请求之和不会超过控制器处理能力, 式(21)表示网络中不会出现所有链路同时失效的状况。

4 算法设计

基于多控制器部署模型, 为了最大化控制平面可靠性, 同时实现控制器负载均衡, 本文为 MCB D 策略设计了相应的算法: 针对 $\max[PQ, NE]$, 设计了控制器位置选取 (CLS, controller location selection) 算法, 对控制器的部署位置进行优化; 在 CLS 算法的基础上, 针对 $\max[CLB]$, 改进现有的 k -center 方法, 设计了 SDN 子域规划 (SSP, SDN subdomain partition) 算法, 实现交换机的合理分配。下面对这 2 种算法进行详细介绍。

4.1 控制器位置选取

在选取控制器位置时, 必须对网络中所有节点进行考量。若某个节点适合部署控制器, 可能是因

为该节点的效能较高或所连接的路径质量较好。因此 CLS 算法在选取控制器位置时,同时考虑节点效能 NE 和路径质量 PQ 这 2 种因素,并引入权值参数 δ ($0 \leq \delta \leq 1$),用于调节 NE 和 PQ 二者之间的比重,则 $\max[PQ, NE]$ 转化为和矩阵 $SUM = [\delta NE + (1 - \delta)PQ]$ 。当 $0 \leq \delta \leq 0.5$ 时,CLS 主要依据链路的路径质量来选取控制器位置;当 $0.5 < \delta \leq 1$ 时,CLS 侧重于节点本身的节点效能。

CLS 算法的核心思想是通过迭代比较的方式逐步精确网络中所需控制器数量,并存储对应的控制器节点,同时设定均衡因子 ε 对控制器位置进行优化,如式(22)所示,避免网络中控制器分布过于集中或分散。

$$\varepsilon = \sum_{i=1}^n \sum_{j=1}^n \left(\frac{d_{ij}}{x_{ij}} \right) \quad (22)$$

CLS 算法的流程如下。首先,收集网络信息,设定控制器数量的上下界(第 2)行~第 3)行)。然后从 k_{\min} 开始以 $\frac{1}{k}$ 长度对权值参数 δ 进行分割,并计算相应节点的 (NE_i, PQ_i, δ) ,将结果存储到向量 SUM_{δ}^i (第 6)行),形成矩阵 SUM (第 8)行)。筛选出 SUM 中每一行元素最大值构建列矩阵 SUM^* 并计算 $\overline{\max SUM}$ (第 9)行~第 10)行)。对所有节点进行迭代,依据选取条件(第 13)行),挑选出控制器预部署节点存储于集合 C^* 中(第 15)行)。以二分法在上下界之间搜索控制器标签 tag 值,同时引入均衡因子(第 18)行~第 21)行),通过不断迭代调整,最终得到控制器集合 C 和位置集合 CP (第 24)行~第 25)行)。算法伪代码如算法 1 所示。

算法 1 控制器位置选取 (CLS)

输入 网络拓扑 $G = (V, E)$, 交换机请求速率 λ_i , 链路带宽 ω_j

输出 控制器集合 C , 控制器位置集合 CP

1) 收集网络中节点和链路信息 (n, x_{ij}, d_{ij})

$$2) k_{\min} = INT \left[\frac{\sum_{i=1}^n \lambda_i}{\Omega} \right], k_{\max} = n$$

3) 设定控制器部署数量范围 $[k_{\min}, k_{\max}]$

4) 初始化 $k = k_{\min}$

5) for $(\delta = 0; \delta \leq 1; \delta = \delta + \frac{1}{k})$

6) 计算 $SUM_{\delta}^i \leftarrow (NE_i, PQ_i, \delta)$

7) end for

8) 矩阵 $SUM = [SUM_{\delta=0}^i, SUM_{\delta=\frac{1}{k}}^i, SUM_{\delta=\frac{2}{k}}^i, \dots,$

$SUM_{\delta=1}^i]_{[n \times (k+1)]}$

9) $SUM^* = [\max SUM^1, \max SUM^2, \dots, \max SUM^n]^T$

10) 计算 $\overline{\max SUM} = \frac{1}{n} \sum_{i=1}^n \max SUM^i$

11) $tag = 0, C^* = \{\phi\}$ (C^* 表示控制器预部署位置集合)

12) for $i = 1$ to $i = n$

13) if $\max SUM^i \geq \overline{\max SUM}$

14) $tag = tag + 1$

15) $C^* = C^* \cup \{c_{tag, v_i}\}$

16) end if

17) end for

18) if $k_{\min} < tag < \frac{(k_{\min} + k_{\max})}{2}$

19) then $k++$ return 第 5)行

20) else if $\exists c_{i, v_m}, c_{j, v_n} \in C^*$ 有 $(0 < l_{mn} < \varepsilon) \cup (l_{mn} > 2\varepsilon)$

21) then $C^* = C^* \setminus \{c_{i, v_m}, c_{j, v_n}\}, tag = \frac{(tag + k_{\max})}{2},$

return 第 13)行

22) end if

23) end if

24) $C \leftarrow C^*$

25) 得到控制器位置信息 CP

定理 2 CLS 算法是收敛的。

证明 CLS 算法包括 3 个部分。构建 SUM 矩阵,挑选控制器预部署节点集合 C^* 和优化控制器部署位置。初始阶段,算法每次迭代都会动态改变权值 δ 从而得到相应的 SUM 元素值,假设当前迭代没有选出最大值,则会增加控制器部署节点,同时对 $[0, 1]$ 进行更细粒度划分,重新变换 SUM 。调整后,每个节点的 SUM_{δ}^i 通过与均值 $\overline{\max SUM}$ 进行比较,去除综合性能评价较低点,由于考虑到均衡因子 ε 的影响,淘汰节点的范围仅局限于 C^* 。对于确定的节点数 n ,节点组合有限,整个算法会在有限的迭代步数内得到集合 C ,具有收敛特性,同时,通过提高迭代初始阈值至 k_{\min} ,加快了算法的收敛速度,证毕。

CLS 算法复杂度分析,在算法初始时刻,需要对所有节点和链路的状态进行统计,其复杂度分别

为 $O(n)$ 和 $O((n-1)^2)$ ，构建 **SUM** 矩阵的复杂度为 $O(nk)$ ，选取 **SUM** 中列向量所需的复杂度为 $O(n)$ ，构建集合 C^* 的复杂度为 $O(tag)$ ，构建集合 CP 的复杂度为 $O(k)$ ，因此整个算法的总复杂度为 $O(n^2 + nk)$ ，由于节点选取方式有限，运算方式仅为简单的乘、加运算，因此算法具有实时性和有效性。

4.2 SDN 子域规划

在 $\max[PQ, NE]$ 的基础上，SSP 算法主要实现 $\max[CLB]$ ，保证各子域控制器的负载均衡。SSP 算法以 k -center 算法为基础， k -center 算法依据节点间距离对节点进行聚类，其目标是最小化节点到其最近控制器的最大距离，但没有考虑节点的流请求速率。在本文所设计的 SSP 算法中，基于节点吸引力(定义 4)，将交换机分配给部署位置确定的控制器，同时设定冗余函数 $\Delta(c_j)$ ，如式(23)所示，保证控制器预留出部分处理容量来应对流量突发和控制器状态同步，最终实现 $|CLB - CLB^*| \leq \xi$ ， ξ 表示实际状况与理想状况得到的控制器负载均衡率差值， ξ 越小，则负载分布越均衡，此时网络具有 $\max[CLB]$ 。

$$\Delta(c_j) = \sum_{v_m \in Domain_j} \left(\frac{\lambda_m}{\sum_{i=1}^n \lambda_i} \right) \quad (23)$$

SSP 算法的流程如下。首先，对子域和控制器部署节点进行初始化(第 1)行)，设置子域交换机平均数量 A ，对于每个控制器都构建节点吸引力矩阵(第 4)行)，并通过降序比较，构建节点矩阵(第 6)行和第 7)行)。检验子域中是否存在重复节点，并根据子域负载进行选择删除(第 9)行~第 18)行)。检验网络中是否存在不属于任何子域的孤立节点，如果存在则就近分配该节点到负载较轻的相邻子域(第 20)行~第 26)行)。和理想状况下 CLB^* 进行比较，重新分配节点或直接输出 G (第 27)行~第 30)行)。算法伪代码如算法 2 所示。

算法 2 SDN 子域规划 (SSP)

输入 网络拓扑 $G=(V, E)$ ，控制器集合 C ，控制器数量 k ，控制器位置集合 CP

输出 SDN 子域规划 $G = \bigcup_{j=1}^k Domain_j$

1) 初始化 $V = \{v_1, v_2, \dots, v_n\}, \forall c_j \in C, Domain_j = \{\emptyset\}$

2) 设置 $A = INT\left(\frac{n}{k}\right)$

- 3) for ($j=1, j \leq k; j++$)
 - 4) 构建 c_j 的节点吸引力矩阵 $A_j = [A_{v_1}^{c_j}, A_{v_2}^{c_j}, \dots, A_{v_n}^{c_j}]_{1 \times (n-k)}$
 - 5) 降序排列，矩阵变换 $A_j^* \leftarrow A_j$
 - 6) 挑选出 A_j^* 中前 A 个元素 $B_j \leftarrow A_j^*$
 - 7) $B_j = [v_1, v_2, \dots, v_A]_{1 \times A}$
 - 8) end for
 - 9) while ($\forall c_p, c_q \in C, B_p \cap B_q \neq \emptyset$)
 - 10) $T_p = \sum_{B_p} \lambda_i, T_q = \sum_{B_q} \lambda_i$
 - 11) if ($T_p \geq \Delta(c_p)\Omega \cap (T_q \geq \Delta(c_q)\Omega)$)
 - 12) then $B_p \leftarrow B_p \setminus \{B_p \cap B_q\}$ and $B_q \leftarrow B_q \setminus \{B_p \cap B_q\}$
 - 13) else ($T_p \geq \Delta(c_p)\Omega \cap (T_q < \Delta(c_q)\Omega)$)
 - 14) then $B_p \leftarrow B_p \setminus \{B_p \cap B_q\}$ or $B_q \leftarrow B_q \setminus \{B_p \cap B_q\}$
 - 15) else $T_p < T_q < \sum_{B_q} \lambda_i$
 - 16) then $B_q \leftarrow B_q \setminus \{B_p \cap B_q\} A = A + 1$ return
- 第 6)行
- 17) end if
 - 18) end while
 - 19) $CS_j \leftarrow B_j, G^* = \bigcup_{j=1}^k Domain_j$
 - 20) while ($G^* \neq G$)
 - 21) $Isolate = G \setminus G^*$
 - 22) $v_i \in Isolate$ ， v_i 的相邻子域为 $Near_i = \{Domain\}$
 - 23) if $Domain_j \in Near_i$ ，有 $\min \Sigma \lambda_j$
 - 24) $Domain_j = Domain_j \cup \{v_i\}$
 - 25) end if
 - 26) end while
 - 27) if $|CLB - CLB^*| > \xi$
 - 28) 返回第 9)行
 - 29) else 输出 $\max(CLB), G \leftarrow G^*$
 - 30) end if

SSP 算法复杂度分析，算法开始时，构建网络中所有控制器的节点吸引力矩阵，其复杂度为 $O(nk)$ ；子域重复节点检验过程的复杂度为 $O(k^2 - k)$ ；孤立节点检验过程的复杂度为 $O(|Isolate|)$ 且 $O(|Isolate|) \leq O(k)$ ，因此整个算法

的总复杂度为 $O(nk + k^2)$, 在数据处理时, 只需要进行线性运算, 包括求和、相乘等, 无复杂计算, 最终算法快速收敛, 具有实时性。

5 仿真分析

5.1 仿真环境建立

关于仿真环境搭建和实验参数设定, 本文做出如下说明。

1) 实验平台介绍

本文选取 Opendaylight 控制器作为实验控制器, 同时采用斯坦福大学研发的 Mininet^[23]测试平台。Opendaylight 控制器基于 Java 语言编写, 运行于 JVM 上, 支持多种版本的 OpenFlow 协议。本文设计的 CLS 和 SSP 算法均在 OpenDaylight 应用层以 Java 编程的方式进行实现。为了避免性能冲突, 将 Mininet 和 Opendaylight 部署在不同的物理设备上。实验机器的配置为 Intel Corei7 3.3 GHz, 8 GB RAM, 2 Gbit/s 网卡, 装载 Ubuntu14.04 系统。

2) 拓扑选择

在拓扑选择方面, 为了使实验更具代表性, 实验中拓扑模型均取自 Topology zoo^[24], 该项目由澳大利亚政府和阿德雷德大学共同创建, 选取的拓扑信息如表 2 所示。

拓扑	节点数	链路数
abilene	12	15
cost266	37	57
germany50	50	88
columbus	70	85
tw	96	118

3) 参数设定

为了模拟真实的网络流量状况, 交换机的流请求都具有文献[15]所示的流量特征, 平均流产生速率为 500 kbit/s。网络中所有控制器的处理性能相同, 设置控制器处理容量上限为 5 Mbit/s, 控制器资源利用率范围为 0~1。设链路带宽范围为 10~15 Mbit/s。不同设备发生故障的概率相互独立^[25], 控制器负载均衡率差值 $\xi = 0.1$ 。

5.2 仿真实验分析

为了说明本文所设计的 MCB D 策略性能, 在仿真实验中与随机部署 (RD, random deployment) 策略、 k -center 策略^[22]和 Survivor 策略^[17]进行比较,

其中, RD 策略在网络中随机选取控制器部署位置, 将交换机平均分配给每一个控制器。 k -center 策略根据节点距离进行聚类, 同时设定聚类中心作为控制器部署位置。Survivor 策略选择不相交路径最多的节点作为控制器部署位置, 分配交换机给距离最近的控制器。实验评估度量包括控制器数量、连通失效率、交换机到控制器平均时延和控制器负载均衡率。

5.2.1 控制器数量

基于表 2 中不同的网络拓扑, 对比 4 种部署策略得到的控制器数量。实验结果如图 2 所示。可以看出, 在网络规模较小时, 4 种策略的控制器部署数量 (2 个) 相同。随着网络规模不断扩大, 控制器数量均呈现递增趋势。从拓扑规模最小 (12 个节点) 的 abilene 到规模最大的 (96 个节点) tw, RD 策略的控制器增量最为明显, 从 2 个增长到 12 个。整体来看, k -center 和 Survivor 的增长幅度近似, 但在 germany50 和 columbus 网络中, Survivor 基本维持不变, k -center 持续上升, 这是因为 Survivor 的控制器部署与不相交路径有关, 虽然 germany50 和 columbus 的节点数量不同 (50 和 70), 但链路数量基本相同 (88 和 85), 因此在这 2 个拓扑中 Survivor 得到控制器数量保持不变 (7 个)。相比上述 3 种策略, MCB D 通过计算节点效能和路径质量, 并引入权值参数 δ , 以二分法的方式进行迭代, 且每次迭代都会细化搜索粒度, 同时设定均衡因子 ε 来寻找控制器部署最优解。无论对于小规模网络还是大规模网络, MCB D 策略都能较好地优化控制器部署, 所需控制器数量平均减少 22.1%。

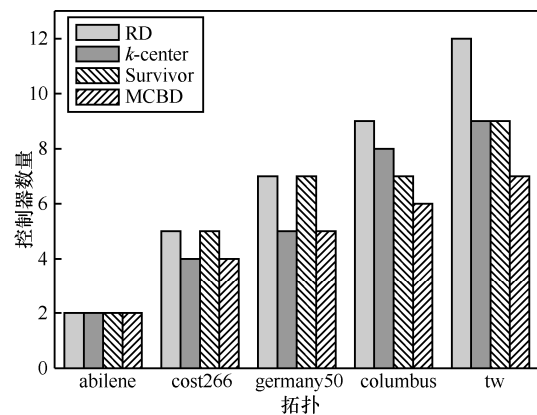


图 2 不同策略得到的控制器数量

5.2.2 连通失效率

连通失效率 (CLP) 是指控制器和交换机之间数

据传输中断的概率，如式(24)所示，同时考虑节点可靠性和链路故障 2 个方面因素，其中， R_i 、 $P_{break}(e_{ij})$ 和 $P_{congest}(e_{ij})$ 的计算分别如式(4)、式(10)和式(11)所示。

$$CLP = \frac{1}{k} \sum_{i=1}^k [R_i + \sum_{j=1, j \neq i}^n (P_{break}(e_{ij}) + P_{congest}(e_{ij}))] \quad (24)$$

为了方便分析，设定 p_{unit} 为每 10 km 的链路故障率。选取 germany50 作为实验拓扑，如图 3 所示，图 3 中所有链路上数字表示 2 个节点之间距离，单位为 km。

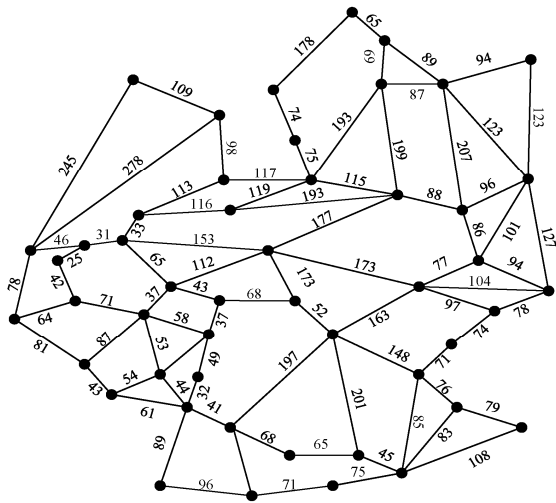


图 3 germany50

观察 CLP 随 p_{unit} 的变化情况，结果如图 4 所示，在图 4(a)中， p_{unit} 的范围值为 [0.01,0.1]。当 $p_{unit} = 0.01$ 时，4 种策略的连通失效率基本为 0，随着 p_{unit} 逐渐增大，连通失效率基本呈线性增长趋势。当 $p_{unit} = 0.1$ 时，随机部署 RD 的 CLP 值最大 (0.168)，其余 3 种策略都考虑到节点和链路可靠

性，因此在低链路故障率下， CLP 均处于较低值， k -center, Survivor 和 MCBD 分别为 0.076、0.063、0.042，MCBD 策略的 CLP 值最小。

将链路故障率的最大值由 0.1 调整至 1，结果如图 4(b)所示。随着链路故障率不断增大，就 CLP 增长速率而言，RD 增长最快， k -center 次之，接下来是 Survivor，MCBD 增长最慢，这是因为 RD 在部署时未考虑节点和链路可靠性，因此，当发生链路故障时，数据传输会很快中断，连通失效率最高。 k -center 和 Survivor 分别考虑链路距离最短和链路数量最多，能在一定程度上预防连通失效。MCBD 综合节点效能和路径质量，选取具有最大可靠度的节点部署控制器，同时保证所选节点相连链路的失效概率最低，因此，能够有效应对链路故障所引发的连通失效， CLP 增长趋势最为缓慢，当 $p_{unit} = 0.3$

(图 4(b)中虚竖直线所示)，此时，4 种策略的 CLP 差值达到最大。

5.2.3 交换机到控制器平均时延

在本实验，基于 germany50 拓扑，比较不同策略中交换机到控制器平均时延随控制器数量变化状况。为了简化分析并突出主要参数对于时延因素的影响，在此将交换机到控制器的时延划分为 3 个部分，包括转发时延 t_r 、控制器处理时延 t_d 和传输时延 t_s 。其中，转发时延与交换机的转发性能以及节点间距离和跳数有关。考虑现有的 OpenFlow 交换机均为多核交换机，支持多端口多 VLAN 背板上数据的并行处理^[26]，因此，数据分组能被节点上交换机快速处理和转发。同时，由于实验拓扑 germany50 的节点分布较为集中，节点间距离相对平均，所以跳数因素对于数据分组转发的影响基本

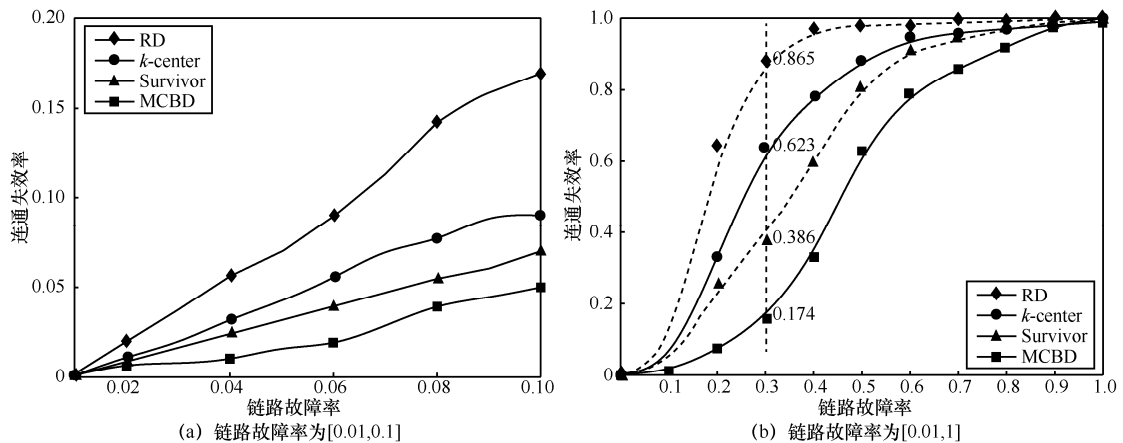


图 4 链路故障率对 CLP 的影响

相同。综合上述分析可知, 由于 OpenFlow 交换机的高处理性能和网络拓扑节点的密集分布, 不同节点间转发时延相差不大, 基于相关统计数据^[13], 设定 $t_r = 0.05$ ms。OpenDaylight 作为实验控制器, 采用多线程处理方式, 设定控制器处理时延 $t_d = 0.01$ ms。传输时延 t_s 与控制路径长度有关, 每 10 km 路径引入 1 ms 传输时延。实验结果如图 5 所示。

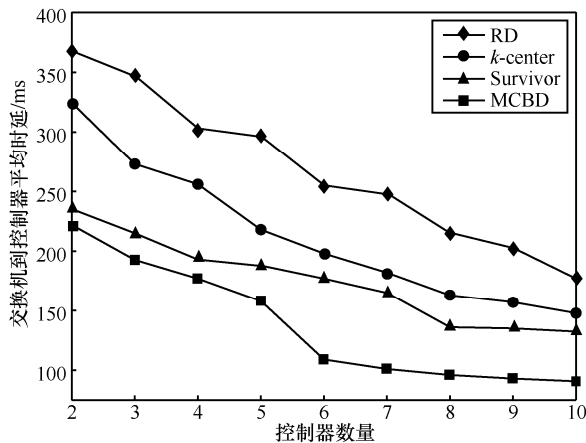


图 5 不同策略得到的交换机到控制器平均时延

在控制器数量相同的情况下, MCBD 策略在 4 种部署策略中具有最小的平均时延。当控制器数量较少时, Survivor 策略的平均时延与 MCBD 策略接近, 随着控制器数量逐步增加, 达到 MCBD 策略的最优数量 5 个 (实验数据来源于图 2), 此时 MCBD 策略对于时延的优化效果明显优于 Survivor。k-center 策略在控制器数量较多时得到的平均时延接近 Survivor 策略, 但相比于 MCBD 策略, 优化效果仍然较差。通过调整负载均衡率差值 ξ , 令 $\xi = 0.05$ 和 $\xi = 0.01$, 重复上述实验过程, 可以实现不同策略平均时延的细粒度比较, 但精度相比于 $\xi = 0.1$ 提升效果有限。因此本文预先设定的 $\xi = 0.1$ 已经能够较好地体现出不同策略对于平均时延的优化和比较。

5.2.4 控制器负载均衡率

该实验使用控制器负载均衡率 CLB 来评判不同策略得到的控制器负载分布状况, CLB 越高, 则部署策略性能越好。基于 germany50 拓扑, 节点间流量服从泊松分布, 该实验设定网络中流量总和维持在相对稳定状态。通过改变控制器部署数量, 观察 4 种策略的控制器负载均衡率变化情况, 实验结果如图 6 所示。由于控制器处理容量限制, 因此当控制器数量较少 ($k \leq 2$) 时, 所有控制器均处于

过载状态, 此时 4 种策略得到的 CLB 值最高。随着控制器数量增加, 随机部署 RD 的 CLB 值普遍较低, 且具有不稳定性。其他 3 种策略的 CLB 值在 $k = 3$ 时达到最小, 随后有所上升。这是因为在网络流量总和不变的情况下, 虽然 $k = 3$ 已经实现了控制器数量的增长, 但增长幅度有限, 仅新增一个控制器仍旧难以满足过量的流处理需求。其次, 由于网络流量具有过收敛特性, 新增控制器处于轻载状态, 能够分担的处理流量较少。网络中过载控制器比例远高于轻载控制器 (过载:轻载=2:1), 控制器间负载差异相比于 $k = 2$ (控制器全部过载) 更加明显。因此在 $k = 3$ 时, k-center、MCBD 和 Survivor 的 CLB 值普遍降低。当控制器数量增长到一定程度时, 过载控制器负载由大量新增控制器进行分担, 负载差异减小, CLB 值又重新上升。由于 k-center 策略只根据节点距离进行交换机聚类, 未考虑交换机流量状况, 因此 CLB 值大约保持在 0.77。Survivor 策略考虑连通性和控制器容量因素, 且随着控制器数量增多, 该策略设定的不相交路径逐渐减少, CLB 值处于较低水平。MCBD 在可靠性部署的基础上, 通过计算节点吸引力合理分配交换机, 同时引入冗余函数来预留控制器处理容量, CLB 值稳定在 0.92, 相比于其他 3 种策略, 最少提升了 19.4%。

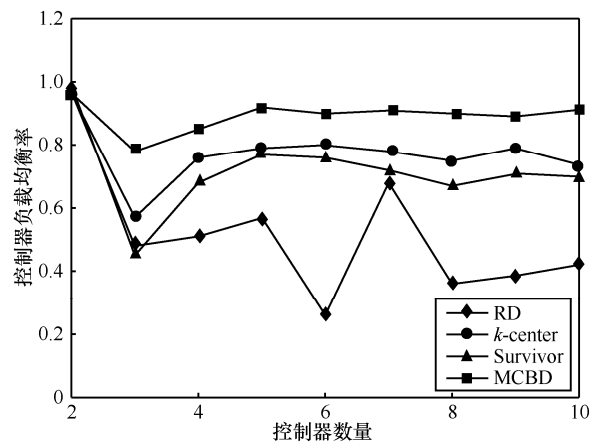


图 6 控制器负载均衡率随控制器数量变化

6 结束语

本文研究了分布式 SDN 中控制器可靠部署及其负载均衡问题, 提出了一种基于可靠性评估的多控制器均衡部署 (MCBD) 策略。通过综合节点效能和路径质量 2 种因素, 设定权值参数和均衡因子, 优化控制器部署位置, 提高控制平面可靠性; 在此

基础上,改进 k -center 聚类方法,依据节点吸引度、控制器负载均衡率和冗余函数,完成交换机弹性分配,实现合理的 SDN 子域规划。实验结果表明,本文提出的控制器部署策略能够适应不同规模的网络拓扑,具有节约控制器数量、控制平面可靠性高、时延小和负载均衡的优点。下一步研究工作将围绕失效转移和路径保护 2 个方面内容展开。

参考文献:

- [1] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: Enabling innovation in campus networks[J]. Seattle ACM Sigcomm Computer Communication Review, 2008, 38(2):69-74.
- [2] GUO Z H, 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.
- [3] GUO Z H, LIU R, XU Y, et al STAR: preventing flow-table overflow in software-defined networks[J]. Computer Networks, 2017: 1-11.
- [4] TOOTOONCHIAN A, GANJALI Y. HyperFlow: a distributed control plane for OpenFlow[C]//The 2010 Internet Network Management Workshop on Research on Enterprise Networking. 2010:1-6.
- [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. Onix: a distributed control platform for large-scale production networks[C]//Usenix Conference on Computer Systems Design and Implementation. 2010:351-364.
- [7] HU Y, WENDONG W, GONG X, et al. Reliability-aware controller placement for software-defined networks[C]//2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013). 2013:672-675.
- [8] VIZARRETA P, MACHUCA C M, KELLERER W. Controller placement strategies for a resilient SDN control plane[C]//The 8th International Workshop on Resilient Networks Design and Modeling (RNDM). 2016:253-259.
- [9] GUO Z H, XU Y, CELLO M, et al. JumpFlow: reducing flow table usage in software-defined networks[J]. Computer Networks, 2015, 92: 300-315.
- [10] HELLER B, SHERWOOD R, MCKEOWN N. The controller placement problem[C]//HotSDN. 2012: 7-12.
- [11] BARIM F, ROYA R, CHOWDHURYS R, et al. Dynamic controller provisioning in software defined networks[C]//The 9th International Conference on Network and Service Management. 2013: 18 -25.
- [12] SALLAHI A, ST-HILAIRE M. Optimal model for the controller placement problem in software defined networks[J]. IEEE Communications Letters, 2015, 19(1):30-33.
- [13] ADLEN K, MILOUD B, TARIK T. On using bargaining game for Optimal Placement of SDN controllers[C]//IEEE ICC Next- Generation Networking and Internet Symposium. 2016:1-6.
- [14] 张联镇, 黄传河, 曾毓琮, 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(5): 1121-1125.
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(5): 1121-1125.
- [15] ZHAO Y, HE R, CHEN H, et al. Experimental performance evaluation of software defined networking (SDN) based data communication networks for large scale flexigrid optical networks[J]. Optics Express, 2014, 22(8):9538-9547.
- [16] HU Y, WANG W, GONG X, et al. On reliability-optimized controller placement for software-defined networks[J]. China Communications, 2014, 11(2):38-54.
- [17] MÜLLER L F, OLIVEIRA R R, LUIZELLI M C, et al. Survivor: an enhanced controller placement strategy for improving SDN survivability[C]//2014 IEEE Global Communications Conference. 2014: 1909-1915.
- [18] TAO W, LU F M, GUO J. Dynamic SDN controller assignment in data center networks: stable matching with transfers[C]//35th Annual IEEE International Conference on Computer Communications. 2016: 1-9.
- [19] BEHESHTI N, ZHANG Y. Fast failover for control traffic in software-defined networks[C]//Global Communications Conference (GLOBECOM). 2012: 2665-2670.
- [20] AGUADO A. First Experimental demonstration of secure NFV orchestration over an SDN-controlled optical network with time-shared quantum key distribution[C]//The 42nd European Conference on Optical Communication. 2016: 1-3.
- [21] GIANFRANCO N, BJARNE E. ANDRES J. G, et al. Impact of SDN Controllers Deployment on Network Availability[J]. NUTU, 2016, 18(03):66-74.
- [22] 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.
- [23] LANTZ B, HELLER B, MCKEOWN N. A network in a laptop: rapid prototyping for software-defined network[C]// HotNets. 2010: 1-6.
- [24] 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.
- [25] LE K, LI V. Modeling and analysis of systems with multimode components and dependent failures[J]. IEEE Transactions on Reliability, 1989, 38(1):68-75.
- [26] OGASAWARA S, TAKAHASHI Y. Performance analysis of traffic classification in an OpenFlow switch[C]//2016 Cloudification of the Internet of Things (CIoT). 2016: 1-6.

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。

马腾(1980-),男,陕西西安人,国家数字交换系统工程技术研究中心,主要研究方向为网络虚拟化、流量调度。

赵伟(1990-),男,河南郑州人,国家数字交换系统工程技术研究中心硕士生,主要研究方向软件定义网络、拓扑还原。