

# 时延和可靠性感知的多控制器均衡部署策略

赵文文<sup>1</sup>, 孟相如<sup>1</sup>, 康巧燕<sup>1</sup>, 苏玉泽<sup>2</sup>

(1. 空军工程大学信息与导航学院, 西安, 710038; 2. 94303 部队, 山东潍坊, 261000)

**摘要** 当前的软件定义网络多控制器部署问题研究, 大多针对控制网络时延、可靠性和负载均衡等指标中的部分进行优化, 对上述因素的整体考虑较少。针对该问题, 首先分析了控制器部署对网络时延、可靠性和负载均衡的影响; 其次, 提出了以全网平均时延、控制路径可靠性和负载均衡度为参数, 以网络综合性能为目标的控制部署优化评价模型; 最后, 基于模拟退火-遗传算法提出一种时延和可靠性感知的控制器均衡部署方法, 在考虑网络综合性能的同时, 增强了解空间的全局搜索能力, 得出了控制器部署的全局非劣最优解集。仿真结果表明, 提出的部署策略在保证负载均衡的前提下, 提高了控制网络的可靠性, 降低了网络时延, 进而提高了网络整体性能。

**关键词** 软件定义网; 多控制器部署; 模拟退火-遗传算法; 时延; 可靠性; 负载均衡

**DOI** 10.3969/j.issn.1009-3516.2021.04.013

**中图分类号** TP393 **文献标志码** A **文章编号** 1009-3516(2021)04-0085-07

## A Delay and Reliability Aware Multi-Controller Balancing Deployment Strategy

ZHAO Wenwen<sup>1</sup>, MENG Xiangru<sup>1</sup>, KANG Qiaoyan<sup>1</sup>, SU Yuze<sup>2</sup>

(1. Information and Navigation College, Air Force Engineering University, Xi'an 710038, China;  
2. Unit 94303, Weifang 261000, Shandong, China)

**Abstract** Aimed at the problems that at present most of the current researches on multi-controller deployment in software-defined networks mainly focus on the optimization of the control network delay, reliability and load balancing, but less on the overall consideration of the above factors, this paper firstly analyzes the impact of controller deployment on network delay, reliability and load balancing. Secondly, the paper proposes an optimal evaluation model of controller deployment, which takes the average delay, control path reliability and load balance of the whole network as parameters while the comprehensive performance of the network as the objective. Finally, based on the simulated annealing genetic algorithm, the paper proposes a delay and reliability aware controller deployment method in consideration of the overall performance of the network, enhancement of the global search ability of the solution space, obtaining the non inferior optimal solution set of controller deployment from global. The simulation results show that on the premise of ensuring load balancing, the proposed deployment strategy improves the reliability of the control network, reduces the network delay and improves the overall performance of the network.

**Key words** software-defined networks; multi-controller deployment; simulated annealing genetic algorithm; time delay; reliability; load-balance

**收稿日期:** 2021-04-12

**基金项目:** 国家自然科学基金(61871396)

**作者简介:** 赵文文(1984—), 男, 宁夏固原人, 硕士生, 研究方向: 下一代网络、网络安全。E-mail: nudtzww@163.com

**引用格式:** 赵文文, 孟相如, 康巧燕, 等. 时延和可靠性感知的多控制器均衡部署策略[J]. 空军工程大学学报(自然科学版), 2021, 22(4): 85-91. ZHAO W W, MENG X R, KANG Q Y, et al. A Delay and Reliability Aware Multi-Controller Balancing Deployment Strategy[J]. Journal of Air Force Engineering University (Natural Science Edition), 2021, 22(4): 85-91.

传统网络因其数据和控制平面高度耦合,使得网络的实时动态调整与快速升级部署比较困难。软件定义网(software-defined networks, SDN)通过对网络数据平面和控制平面的功能解耦,让网络功能扩展更加灵活高效,并通过北向接口向上层用户提供程序接口,为网络运行维护的程序化、自动化提供了可能。

多控制器 SDN 中如何多控制器 SDN 数量和部署位置,并合理划分控制域等问题一直是当今研究的热点。Heller 等人在 2012 年首次提出 SDN 中控制器部署是 NP-Hard 问题,并研究了不同部署位置对系统平均时延和最大时延带来的影响<sup>[1]</sup>。文献[3~5]分别从不同角度,考虑了时延和负载指标下多控制器部署问题,而对可靠性问题涉及不足。文献[6]在考虑控制器负载和响应时间的基础上,研究了在考虑可靠性情况下的控制器部署策略,然而其仅考虑了最差时延,未考虑全网平均控制时延和控制器之间负载均衡差异,且其使用的细菌觅食算法收敛性较差。文献[7]通过综合权衡节点效能和路径质量评估节点可靠性,并设定负载均衡因子以优化控制器部署位置,而没有考虑控制时延因素。文献[8]建立了一种针对链路故障引起的控制时延变动的 SDN 控制网络可靠性模型,提出了一种控制器部署算法,却并未考虑负载均衡因素。文献[9]以交换机节点的负载和拓扑中心度为依据,建立网络可靠度模型,通过该模型计算满足可靠性要求的控制器部署方案,但其仅以节点拓扑中心度作为可靠性依据,对网络实际运行时的可靠性考虑较少。文献[10]在已知交换节点和控制器节点故障率的条件下,建立了一种可容错的控制器部署模型(fault tolerant controller placement, FTCP),但其仅考虑了可靠性,对时延、负载等因素没有考虑。

文献[7]、文献[11]均提出了随机部署算法,在满足控制器负载容量限制的前提下,采用一定随机策略在交换节点中选取控制器部署位置,随机部署策略在计算速度上有一定优势,但由于没有考虑其他因素,导致所得解空间的整体性表现不佳。文献[12]提出了用  $k$ -means 聚类算法解决控制器部署问题,将控制器部署和控制域划分转换为类簇中心点的确定和类簇的划分问题,然而其仅依据物理距离进行控制域划分,没有考虑负载均衡等问题,且初始类簇数量和类簇中心点的选取对聚类效果的好坏有很大的影响。

综上所述,针对控制器部署问题,当前研究只考虑了网络时延、负载均衡和可靠性中的部分因素,而对上述因素综合考虑较少<sup>[13-15]</sup>,针对该问题,本文

在综合衡量时延、可靠性和负载均衡因素的基础上,建立了控制器部署评价模型,提出了基于模拟退火-遗传算法(simulated annealing genetic algorithm, SAGA)的时延和可靠性感知的多控制器均衡部署策略(delay and reliability aware balancing strategy based on SAGA, SAGA-DRABS)。

## 1 问题描述与模型构建

SDN 按照控制信息流经的路径不同,分为带内控制 SDN 和带外控制 SDN。本文讨论采用带内控制方式时 SDN 的多控制器部署问题,即按照一定的评价策略和指标,在底层交换网络选取合适的交换机节点,部署 SDN 控制器,并对控制器所管理的交换节点合理划分,使网络性能达到最优。

底层交换网络用加权有向图  $G(S, E)$  表示,其中  $S = \{s_1, s_2, \dots, s_n\}$ ,  $S_i$  为交换机节点,  $n$  为交换机的数量,  $E = \{e_{ij} \mid i, j \leq n\}$  为图  $G$  中边的集合,其中  $e_{ij} = \langle s_i, s_j \rangle$  为节点  $s_i$  与  $s_j$  之间的边,边的权值  $w_{ij}$  为路径的物理距离。将交换网络划分为  $k$  个控制域等价于将图  $G$  划分为  $k$  个子图,可以表示为:

$$G_{\text{cut}} = \{G_1, G_2, \dots, G_k \mid k \leq n\}, \bigcap G_j = \emptyset (G_j \in G_{\text{cut}}) \quad (1)$$

控制器用集合  $C = \{c_1, c_2, \dots, c_k \mid k \leq n\}$  表示,其中  $c_j (j \leq k)$  为控制域  $G_j$  的控制器。控制域  $G_j$  的大小为  $|G_j|$ , 表示  $G_j$  中交换机的数量。如前所述,在对  $G$  进行分割时,主要考虑以下因素:控制时延、负载均衡度、可靠性和 DRB 度量( $M_{\text{DRB}}$ )。

### 1.1 控制时延

SDN 控制时延是 SDN 南向接口运行效率的重要表征,也是 SDN 控制网络最重要性能指标之一。SDN 控制时延包括控制器与控制器之间的时延和控制器与交换机之间的时延,在实际应用中,控制器东西向接口之间通常用高速度大容量高带宽专线连接,与控制器与交换机之间的时延相比,控制器之间的时延几乎可以忽略不计,因此本文主要考虑控制器与交换机之间的时延,用网络平均时延进行衡量,为此,先定义各节点时延如下:

**定义 1** 节点控制时延( $T_{\text{total}}^i$ )。节点控制时延是指当有未知流流入交换机  $s_i$  时,  $s_i$  向控制器  $c_j$  发送 packet\_in 请求,  $c_j$  接收并处理后,向  $s_i$  返回 packet\_rm 应答的时间,其值为发送时延( $T_{\text{trans}}^i$ )、等待时延( $T_{\text{wait}}^i$ )和传播时延( $T_{\text{prop}}^i$ )之和,如式(2)所示。

$$T_{\text{total}}^i = \alpha \cdot 2T_{\text{trans}}^i + \beta \cdot T_{\text{wait}}^i + \gamma \cdot T_{\text{prop}}^i \quad (2)$$

$$\alpha + \beta + \gamma = 1 \quad (3)$$

$$0 \leq \alpha, \beta, \gamma \leq 1 \quad (4)$$

式中:  $\alpha$ 、 $\beta$ 、 $\gamma$  为比例参数,针对不同场景可以设置不同的比例参数,比如,在数据中心网络中,由于各交换节点部署较为集中,传播时延对控制器部署影响不大,可设置  $\gamma$  的值为 0。

**定义 2** 网络平均控制时延  $T_{\text{average}}^N$ 。定义为网络总控制时延除以交换节点个数。

$$T_{\text{average}}^N = \frac{1}{n} \cdot T_{\text{total}}^N = \frac{1}{n} \cdot \sum_{i=1}^n T_{\text{total}}^i \quad (5)$$

式中:  $T_{\text{total}}^N$  表示网络总控制时延。网络平均控制时延决定了 SDN 处理未知流的效率,进而对全网的端对端时延产生影响。 $T_{\text{average}}^N$  越小,SDN 网络处理未知流的平均时间越小,进而表示网络的整体性能越优。

## 1.2 负载均衡度(BL)

本文用负载均衡度来衡量 SDN 控制网络中各控制器的负载均衡情况,为此定义各控制器的负载和负载均衡度如下所示:

**定义 3** 控制器  $c_j$  的负载( $RN_j$ )。控制器负载主要由四部分组成:①处理事件中 packet\_in 请求数据包并将处理结果传递给应用程序和交换机;②维护控制域内网络视图;③与其他控制器交互后形成全局网络视图;④向南向接口安装由北向接口产生的流量。网络场景不同,各部分的权重也不同,然而处理事件中 packet\_in 请求数据包通常被认为是控制器负载的最重要部分<sup>[12]</sup>。因此,本文用控制器处理的 packet\_in 请求数据包的数量作为控制器负载的度量。在控制域  $G_j$  中,设交换机  $s_i$  向控制器  $c_j$  发送 packet\_in 请求的数量为  $p_i$ ,控制器  $c_j$  的负载  $RN_j$  可定义为式(6),其中  $\Omega_j$  为控制器  $c_j$  的额定负载。

$$RN_j = \sum_{i=1}^{|G_j|} p_i, RN_j \leq \Omega_j \quad (6)$$

$$BL = \frac{\tau \cdot BL_1 + \xi \cdot BL_2}{k} \quad (7)$$

$$\tau + \xi = 1 \quad (8)$$

式中: $\tau$ 、 $\xi$  为权重参数; $BL_1$  表示各控制器之间的负载差额, $BL_1$  越小表示各控制器之间负载越趋于均衡; $BL_2$  表示各控制器负载方差。 $BL_1$  和  $BL_2$  定义见式(9)、(10)。

$$BL_1 = \frac{1}{2} \cdot \sum_{i=1}^k \sum_{j=1}^k (RN_i - RN_j)^2, i \neq j \quad (9)$$

$$BL_2 = \sum_{j=1}^k (RN_j - \frac{1}{k} \sum_{i=1}^k RN_i)^2 \quad (10)$$

**定义 4** 控制器  $c_j$  的负载率( $\eta_j$ )。指控制器  $c_j$  中,当前负载占额定负载的比率,当控制器达到额定负载时,控制器对未知数据包的请求开始排队处理。

$$\eta_j = \frac{RN_j}{\Omega_j} \quad (11)$$

在对 RN 进行控制域划分时,为控制器  $c_j$  的负载率  $\eta_j$  设定负载阈值  $\eta_{\text{thr}}$ ,使  $\eta_j \leq \eta_{\text{thr}}$ ,当  $\eta_j > \eta_{\text{thr}}$  时,重新调整控制域。

## 1.3 可靠性

SDN 控制网络中,控制器为每个交换节点分别维护一条控制路径,控制路径的可靠性表示控制信息和流表信息从控制器到交换节点的可达程度,流表和控制信息的延迟可能造成控制时延和网络时延抖动。在带内控制的情形下,当底层交换节点的可靠性不同时,控制器的部署位置会导致各控制路径的可靠性也不尽相同<sup>[7-10]</sup>,图 1 中,A、B、C 表示交换机节点,D 代表控制器,假设交换机的可靠性分别为 0.9、0.6、0.3,橙色表示控制器部署位置,在忽略链路可靠性、仅考虑节点可靠性的前提下,3 种部署方案所形成控制路径的可靠性分别为: $R_{(a)} = 0.9 + 0.9 \times 0.6 + 0.9 \times 0.6 \times 0.3 = 1.602$ ; $R_{(b)} = 0.6 + 0.6 \times 0.3 + 0.6 \times 0.9 = 1.32$ ; $R_{(c)} = 0.3 + 0.3 \times 0.6 + 0.3 \times 0.6 \times 0.9 = 0.642$ 。

可以看出,对同一交换网络,不同的控制器部署位置会导致控制路径的可靠性也不同。本文在假定各交换节点的可靠度已知的前提下,分别定义了控制路径、控制域和网络整体可靠性。进行控制器部署就是选取合适的部署策略,使得网络的整体可靠性达到最高。

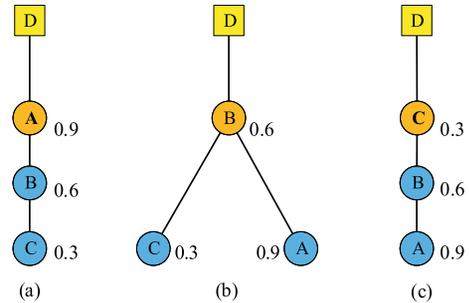


图 1 部署位置对可靠性的影响

**定义 5** 控制域可靠度( $R^{G_j}$ )。SDN 中,每个控制域由多条控制路径组成,本文定义控制域的可靠性度量( $R^{G_j}$ )为各控制路径可靠性之和,如式(12)所示。

$$R^{G_j} = \sum_{i=1}^{|G_j|} R_{ij} = \sum_{i=1}^{|G_j|} \left( \prod_{s_k \in P_{ij}} R_k \right), 0 \leq R_k \leq 1 \quad (12)$$

式中: $R_k$  表示交换节点  $s_k$  的可靠度,为 0 到 1 区间的正数; $R_{ij}$  指控制器  $c_j$  到交换节点  $s_i$  的控制路径  $P_{ij}$  的可靠性,该可靠性度量了控制信息和流表信息到交换节点的可达程度。

**定义 6** 控制网络可靠度 ( $R_{\text{total}}$ )。控制网络可靠度是全网的可信性度量,为 SDN 中所有控制域的可信性之和,如式(13)所示:

$$R_{\text{total}} = \sum_{j=1}^k R^{G_j} = \sum_{j=1}^k \sum_{i=1}^{|G_j|} \left( \prod_{s_k \in P_{ij}} R_k \right) \quad (13)$$

式中:  $R^{G_j}$  为控制域  $G_j$  的可信度。

#### 1.4 DRB 度量 (MDRB)

基于上述定义,本文提出用 DRB 度量  $M_{\text{DRB}}$  来衡量控制器部署策略的整体性能,其与时延和负载均衡度成反比,与可信性成正比,见式(14):

$$M_{\text{DRB}} = \sigma \cdot \frac{Q_T}{T_{\text{average}}^N} + \nu \cdot \frac{Q_B}{BL} + \omega \cdot Q_R \cdot R_{\text{total}} \quad (14)$$

$$\sigma + \nu + \omega = 1 \quad (15)$$

$$0 \leq \sigma, \nu, \omega \leq 1 \quad (16)$$

式中:  $Q_T$ 、 $Q_B$ 、 $Q_R$  为缩放系数,用于平衡量纲,  $\sigma$ 、 $\nu$ 、 $\omega$  分别为比例系数,总和为 1。

根据以上指标,基于时延和可信性感知的多控制器均衡部署策略就是找到一个合适方案,使得目标值达到最大,为此,建立带约束的最优化模型如下式所示:

$$\max \{ M_{\text{DRB}} \} = \text{Max} \left\{ \sigma \cdot \frac{Q_T}{T_{\text{average}}^N} + \nu \cdot \frac{Q_B}{BL} + \omega \cdot Q_R \cdot R_{\text{total}} \right\} \quad (17)$$

$$\sigma + \nu + \omega = 1 \quad (18)$$

$$\text{s. t. } 0 \leq \sigma, \nu, \omega \leq 1 \quad (19)$$

$$\eta_i \leq \eta_{\text{thr}} \quad (20)$$

式中:  $Q_T$ 、 $Q_B$ 、 $Q_R$  为缩放系数,用于平衡量纲;  $\sigma$ 、 $\nu$ 、 $\omega$  分别为比例系数,总和为 1,式(20)表示控制器的负载不超过其额定负载。

## 2 策略描述

### 2.1 模拟退火-遗传算法 (SAGA)

遗传算法借鉴生物进化相关理论,将生物种群染色体编码、选择、交叉等进化机制引入复杂非线性问题解空间的寻优过程,同时通过变异操作帮助算法跳出局部最优陷阱,一定程度上保证了解空间的全局特性。然而遗传算法由于其子代种群的个数与父代适应度大小成正比,且在算法早期因个体适应度差异较大,容易使少数优秀个体的后代占满种群造成“早熟”,而在后期又由于适应度趋于一致,无法较好地发挥优秀个体的遗传作用,导致算法较早陷入局部最优解。

模拟退火算法 (SA) 将待优化问题求解过程模拟为统计热力学中的物理降温过程,在搜索到

最优解(达到热平衡)之前,反复降温,每次降温采用 Metropolis 接受准则,使算法跳离局部最优的“陷阱”。

本文将模拟退火算法与遗传算法相结合用于控制器部署位置选择,由于模拟退火算法和遗传算法可以互相取长补短,通过在遗传算法后期适应度趋于一致时,引入模拟退火的降温过程,对种群采用 Metropolis 接受准则进行取舍,较好地克服了传统遗传算法的早熟现象,使该算法更有效、快速地搜寻到全局较优解。

### 2.2 部署策略

针对上述模型,本文引入模拟退火遗传算法提出了一种时延和可信性感知的多控制器均衡部署策略 SAGA-DRABS。图 2 为策略的具体流程。

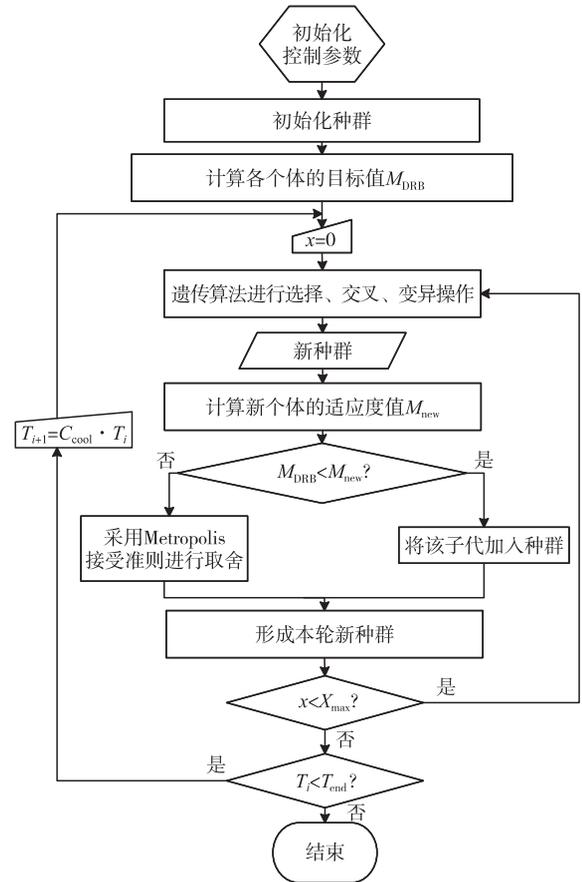


图 2 SAGA-DRABS 策略流程示意图

SAGA-DRABS 首先采用随机选取的办法确定控制器部署位置,组成原始解空间并编码后,形成初始种群,利用式(15)计算各部署位置的  $M_{\text{DRB}}$  值,即该种群所有个体的目标值,得出 SAGA 的适应度值  $\text{fitValue}$ 。而后进入退火降温过程,对由不同部署位置的解空间形成的初始种群利用迭代方法进行交叉、选择、变异等进化操作形成子代个体,每一次进化操作(交叉、选择、变异)后,计算新个体的适应度值  $M_{\text{new}}$ 。若  $M_{\text{new}} > M_{\text{DRB}}$ ,说明进化后的子代优于父

代个体,将该子代个体加入种群,否则,利用 Metropolis 接受准则进行取舍,形成本轮最优种群。继续进入下一轮降温过程,每一轮降温过程都是在上一轮最优种群的基础上,进行交叉、选择、变异等进化操作后,以概率  $p$  进行取舍得出最优种群,如此反复,直至达到最终温度,此时从所得最优种群中选取最优个体即所求结果。

$$p_{\text{choose}} = \exp((M_{\text{new}} - M_{\text{DRB}})T_i) \quad (21)$$

式中:  $T_i$  为每轮降温过程的当前温度值;  $M_{\text{new}}$  和  $M_{\text{DRB}}$  分别表示每次进化形成个体的适应度值。算法伪代码如表 1 所示。

表 1 SAGA-DRABS 策略伪代码

输入:  $G(S, E)$ , 控制器个数  $k$ , 各额定负载等网络基本参数。

输出: 控制器部署位置集合  $C$ , 控制域  $G_{\text{cut}}$ 、 $M_{\text{DRB}}$  值。

1. 初始化控制参数: 分别设置控制域  $G_{\text{cut}} = \phi$ , 初始种群  $\text{POP} = \phi$ , 控制器数量  $k = i$ , 种群个体大小  $N_{\text{pop}}$ , 最大进化次数  $X_{\text{max}}$ , 交叉概率  $P_{\text{inter}}$ , 变异概率  $P_{\text{het}}$ , 退火初始温度  $T_{\text{int}}$ , 温度冷却系数  $C_{\text{cool}}$ , 终止温度  $T_{\text{end}}$ 。
2. while  $i < N_{\text{pop}}$
3.  $C \leftarrow \text{randomGenControllers}(G)$  //随机确定控制器初始位置
4.  $i = i + 1$
5.  $v_{\text{ind}} = \text{genIndiv}(G_{\text{cut}})$  //由部署位置形成的解空间产生个体
6.  $\text{pop} \leftarrow \text{genPop}(v_{\text{ind}})$  //形成  $X_{\text{max}}$  种群
7. end while
8.  $T_i = T_{\text{ini}}$  //设定初始退火温度
9. while  $T_i > T_{\text{end}}$
10.  $x = 0$
11. while  $x < X_{\text{max}}$
12. for  $s$  in  $S$
13. for  $c$  in  $C$
14.  $d_{sc} = \text{shortestPath}(s, c)$  //计算  $s, c$  之间的最短路径
15.  $T_{\text{total}}^c = \text{calcDelay}(s, c)$  //计算时延
16. end for
17.  $T_{\text{total}}^c = T_{\text{total}}^c \cup T_{\text{total}}^c$  //将时延  $T_{\text{total}}^c$  并入时延集合  $T_{\text{total}}^c$
18.  $T_{\text{sorted}}^c = \text{sort}(T_{\text{total}}^c)$  //对  $T_{\text{total}}^c$  进行升序排序得  $T_{\text{sorted}}^c$
19. for  $c$  in  $T_{\text{sorted}}^c$
20. if  $\eta_c + \eta_s \leq \eta_{\text{thr}}$  //当前控制器负载率小于负载阈值
21.  $G_c = G_c \cup s$  //将交换机加入控制器  $c$  的控制域
22. else //负载率超过阈值
23.  $c = \text{next}(c)$ ; //选取下一个控制器
24. end if
25. end for
26. end for
27.  $R_{\text{total}} = \text{calcR}(v_{\text{ind}})$  //根据式(18)计算可靠度

28.  $\text{fitValue} = \text{calcObjFit}(v_{\text{ind}})$  //计算目标值及适应度
29.  $v_{\text{new}} \leftarrow \text{evolve}(\text{pop})$  //选择、变异、较差等进化操作
30.  $M_{\text{new}} = \text{calcObj}(v_{\text{new}})$  //计算新种群的适应度
31. if  $M_{\text{new}} - M_{\text{DRB}} > 0$
32.  $\text{pop} \leftarrow \text{genPop}(v_{\text{new}})$  //将新个体插入种群
33. else
34.  $p_{\text{choose}} = \text{calcP}(M_{\text{new}}, M_{\text{DRB}})$  //计算取舍概率  $p$
35.  $\text{pop} \leftarrow \text{genPopByP}(v_{\text{new}}, p_{\text{choose}})$  //以取舍概率选择新个体
36. end if
37. end while //进化过程结束
38.  $T_{i+1} = T_i C_{\text{cool}}$  //降温
39. end while //退火过程结束

### 3 实验仿真

#### 3.1 仿真设置

基于 Matlab 平台进行实验仿真,仿真在随机生成的包含 80 个交换节点、604 条物理链路,分布在 2 500 km 范围内的模拟网络(见图 3)上进行。实验中,假设到达交换机的未知流数量服从以  $\lambda = 1/14$ 、 $\mu = 1/20$  为参数的泊松分布。控制器额定负载服从以  $\sigma_{\text{load}} = 1\ 000$ 、 $\mu_{\text{load}}$  为参数的高斯分布,见式(23):

$$\mu_{\text{load}} = \frac{n_p}{k \cdot \eta_j} \quad (23)$$

式中:  $n_p$  为未知流数量;  $k$  为控制器数量;  $\eta_j$  为控制器额定负载率。设节点可靠度  $R_i$  服从  $\mu_{\text{reliability}} = 0.95$ 、 $\sigma_{\text{reliability}} = 0.1$  为参数的高斯分布。网络传输带宽为 1 Gbits/s。

SAGA 算法变量初始化方面,设定种群大小  $N_{\text{pop}} = 10$ , 最大进化次数  $X_{\text{max}} = 10$ , 交叉概率  $P_{\text{inter}} = 0.6$ , 变异概率  $P_{\text{het}} = 0.05$ , 退火初始温度  $T_{\text{int}} = 100$ , 变异概率  $C_{\text{cool}} = 0.6$ , 终止温度  $T_{\text{end}} = 0$ 。

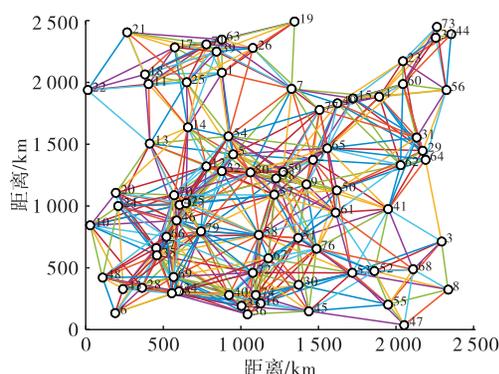


图 3 仿真网络

### 3.2 仿真结果与分析

为对比不同部署策略对 SDN 性能的影响,本文将 SAGA-DRABS 分别与采用  $k$ -means 聚类策略、贪心策略(greedy deployment, GRE)和随机部署(random deployment, RD)策略进行控制器部署时的性能进行了比较。对所有策略进行 50 次计算后取各自结果的算数平均。

图 4 显示当控制器数量为 4 时,通过 SAGA-DRABS 策略对图 3 所示网络进行控制器部署和控制域划分的结果,图中不同颜色表示不同控制域,其中控制器部署位置用相应颜色的“★”表示,编号用红色标出。

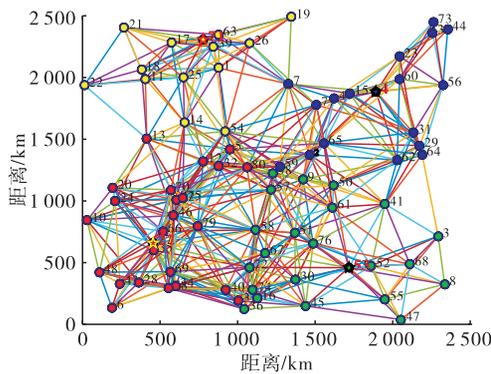


图 4 控制器部署结果

图 5 显示 SAGA-DRABS 的寻优曲线,本次计算共经过 11 次降温,每次降温过程中,分别进行 10 轮遗传算法迭代,每轮迭代求出当前局部最优解,后一次降温均在前一次降温取得最优解的基础上进行新的遗传算法迭代。SAGA-DRABS 中每次降温可以看作一个完整的传统遗传算法流程,可以看出,该策略在传统遗传算法的基础上,又进行了多次降温过程,使当前最优解跳出局部陷阱,进一步增强了解空间的全局特性。

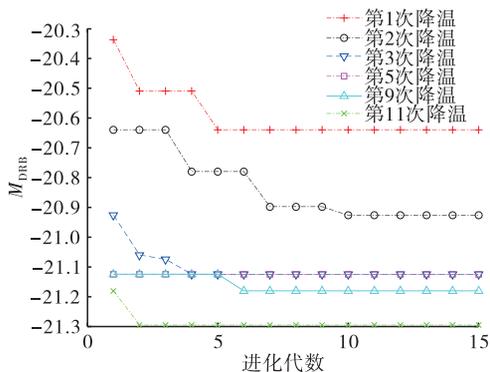


图 5 SAGA-DRABS 寻优曲线

图 6 显示不同控制器数量情况下,各算法所得部署策略的  $M_{DRB}$  值。可以看出,相比其他算法,通过 SAGA-DRABS 所得的  $M_{DRB}$  值更高,主要原因是该策略在考虑网络平均控制时延最小的同时,也将

可靠性和负载均衡度纳入评价指标,同时通过 SAGA 算法使所求结果具有较好的全局特性。作为对比, $k$ -means 算法主要是基于距离影响的全网平均控制时延因素所得的部署策略,在整体目标衡量上要劣于 SAGA-DRABS, GRE 算法主要基于当前最优解寻求控制器最佳部署策略,其  $M_{DRB}$  值与 SAGA-DRABS 算法相比,明显缺乏全局最优特性。而 RD 算法主要采取在满足负载率的情况下的随机部署方法,其 DRB 值也相应为最小。

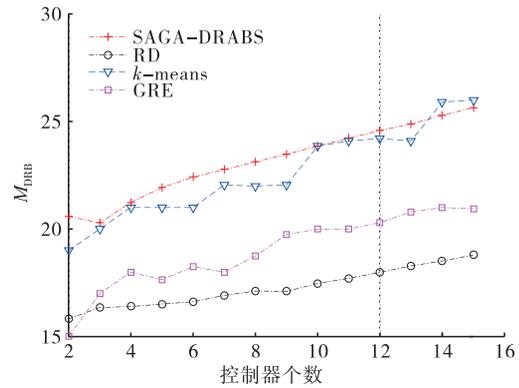


图 6 各策略 DRB 度量指标

图 7~图 9 显示在不同控制器数量情况下,各算法所得部署策略在全网平均控制时延、负载均衡和可靠性度量上的表现,可以看出,由于  $k$ -means 算法主要基于距离因素进行控制器部署,因此在时延表现上要略优于 SAGA-DRABS。而在负载均衡方面,由于贪心算法主要基于当前局部的 DRB 值,既没有考虑 DRB 值的全局性,也缺乏对负载均衡的单独考量,导致该算法相较其他算法,所得策略在负载均衡方面表现较差,而随机部署算法由于在部署时仅考虑控制器负载容量限制,导致该算法在负载均衡度量上表现较优。在可靠性方面可以看出, SAGA-DRABS 所求策略的可靠性较  $k$ -means 算法有较为明显的优势,分析原因,也是因为 DRA-SAGA 既在目标函数中考虑了可靠性因素,也强化了全局最优解的搜索能力,保证了部署策略的可靠性。

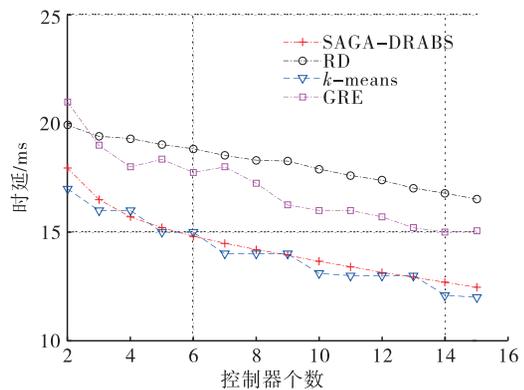


图 7 各策略时延指标对比

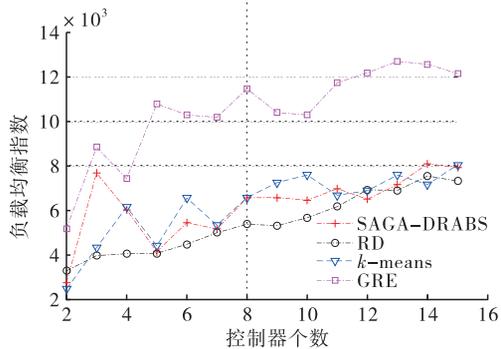


图 8 各策略负载均衡指标对比

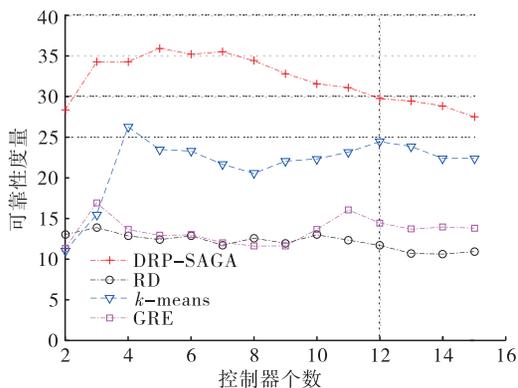


图 9 各策略可靠性指标对比

### 4 结语

本文研究了SDN中多控制器部署问题,通过建立相应模型,研究了不同部署策略对网络时延、负载均衡和可靠性方面的影响,提出了一种时延和可靠性感知的控制器均衡部署策略。仿真结果表明,提出的部署策略在保证负载均衡的前提下,提高了控制网络的可靠性,降低了网络时延,进而提高了网络的整体性能,从而为面向QoS的网络应用提供了可靠的保障。

### 参考文献

[1] HELLER B, SHERWOOD R, MCKEOWN N. The Controller Placement Problem [J]. ACM Sigcomm Computer Communication Review, 2012, 42(4): 473-478.

[2] SHI J G, XIE Y J, SUN L, et al. Multi-Controller Placement Strategy Based on Latency and Load in Software Defined Network [J]. Journal of Electronics & Information Technology, 2019, 41(8): 1869-1876.

[3] 史久根, 谢熠君, 孙立, 等. 软件定义网络中面向时延和负载的多控制器放置策略[J]. 电子与信息学报, 2019, 41(8): 1869-1876.

[4] 史久根, 徐皓, 张径, 等. 软件定义网络中基于效率区间的负载均衡在线优化算法[J]. 电子与信息学报,

2019, 41(3): 694-701.

[5] LANGE S, GEBERT S, ZINNER T, et al. Heuristic Approaches To The Controller Placement Problem in Large Scale SDN Networks[J]. IEEE Transactions on Network and Service Management, 2015, 12(1): 4-17.

[6] JALILI A, KESHTGARI M, AKBARI R. Optimal Controller Placement in Large Scale Software Defined Networks Based on Modified NSGA-II[J]. Applied Intelligence, 2018, 48(9): 2809-2823.

[7] ZHANG B, WANG X W, HUANG M. Multi Objective Optimization Controller Placement Problem in Internet-Oriented Software Defined Network[J]. Computer Communications, 2018, 123: 24-35.

[8] 胡涛, 张建辉, 马腾, 等. SDN中基于可靠性评估的多控制器均衡部署策略[J]. 通信学报, 2017, 38(11): 188-198.

[9] 胡宇翔, 王立业, 胡涛, 等. SDN中基于控制时延可靠性的控制器部署策略[J]. 通信学报, 2018, 39(S2): 144-150.

[10] MOAZZENI S, KHAYYAMBASHI M R, MOVAHHEDINIA N, et al. On Reliability Improvement of Software-Defined Networks[J]. Computer Networks, 2018, 133: 195-211.

[11] WANG G, ZHAO Y, HUANG J, et al. A K-Means Based Network Partition Algorithm for Controller Placement in Software Defined Network [C]//2016 IEEE International Conference on Communication. Kuala Lumpur, Malaysia: IEEE, 2016: 1-6.

[12] ROS F, RUIZ P. On Reliable Controller Placements in Software Defined Networks[J]. Computer Communications 2016, 77: 41-51.

[13] 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.

[14] PIZZUTI C. A Multi-Objective Genetic Algorithm to Find Communities in Complex Networks [J]. IEEE Transactions on Evolutionary Computation, 2012, 16(3): 418-430.

[15] 汪民乐. 遗传算法的收敛性研究[J]. 计算技术与自动化, 2015, 34(1): 58-62.

[16] 何庆, 吴意乐, 徐同伟. 改进遗传模拟退火算法在TSP优化中的应用[J]. 控制与决策, 2018, 33(2): 219-225.

[17] FAN Z F, YAO J, YANG X H, et al. A Multi-Controller Placement Strategy Based on Delay and Reliability Optimization in SDN [C]//2019 28th Wireless and Optical Communications Conference. Beijing, China: IEEE, 2019: 58-62.

(编辑:徐楠楠)