

doi:10.11835/j.issn.1000-582X.2020.09.009

基于网络划分的 SDN 分布式控制器部署

王 坤¹, 吕光宏², 胥 林¹, 杨 晗¹

(1. 西南石油大学 信息学院, 四川 南充 637001; 2. 四川大学 计算机学院, 成都 610065)

摘要:针对大规模 SDN (software defined networking) 网络中分布式控制器部署问题, 以优化网络弹性和可靠性为目标, 提出两阶段控制器部署算法 (TSCP, two-stage controller placement): 利用节点相似度划分控制域, 使得控制域内设备之间的连通性强、连接紧密, 增强控制域的网络弹性; 选择控制路径平均失效率最小的控制器集合作为控制器部署, 提高网络可靠性。通过约束控制域的规模和设备 (交换机或控制器) 之间传播时延, 使控制域的交换机个数均衡, 控制器的部署合理。通过定义性能指标, 实验对比 GCP 算法、K * -means 算法, 结果表明 TSCP 算法可以优化控制域的规模, 均衡控制域的交换机个数, 减少控制器个数, 网络弹性和可靠性均表现较好。

关键词: 软件定义网络; 控制器部署; 网络划分; 相似度

中图分类号: TP393

文献标志码: A

文章编号: 1000-582X(2020)09-081-12

Distributed controller placement in SDN based on network partitioning

WANG Kun¹, LU Guanghong², XU Lin¹, YANG Han¹

(1. College of Information, Southwest Petroleum University, Nanchong 637001, Sichuan, P. R. China;

2. College of Computer Science, Sichuan University, Chengdu 610065, P. R. China)

Abstract: In this paper, the problem of distributed controller deployment in large-scale SDN (software defined networking) network is addressed. Aiming at improving resilience and reliability, a two stage controller placement (TSCP) method was proposed. The control domain was divided by the similarity of node to enhance the connectivity among device in the control domain for improving the resilience. Controller set with the minimum average rate of control path loss was selected as the controller placement for improving the reliability. By constraining the size of control domain and the propagation delay among equipment (switch or controller), the number of switch in control domain was made equalized and the controller placement reasonable. With the performance indexes defined, the comparison of GCP algorithm and K * -means algorithm with TSCP algorithm was made by experiments, and the results showed that TSCP algorithm could optimize the scale of control domain, balance the number of switches in control domain and reduce the number of controllers, thus performing better in network elasticity and reliability.

Keywords: software defined networking; controller placement; network partitioning; similarity

收稿日期: 2020-01-03

基金项目: 国家自然科学基金资助项目 (61373091); 四川省南充市科技资助项目 (19SXHZ0012)。

Supported by National Natural Science Foundation of China (61373091) and Science & Technology Program of Nanchong, China (19SXHZ0012).

作者简介: 王坤 (1987—), 男, 硕士, 主要从事软件定义网络方向研究, (Tel) 13982100493; (Email) wk_scu@163.com。

通讯作者: 吕光宏, 男, 教授, 主要从事网络通信、软件定义网络方向研究, (E-mail) lghong@scu.edu.cn。

软件定义网络(SDN, software defined networking)作为一个新型网络架构^[1-2],通过解耦网络的数据平面和控制平面,解决了传统网络体系结构中存在的设备配置复杂、管理结构臃肿、管控能力不足及可扩展性差等问题,引起了各界的广泛关注。随着 SDN 在大规模网络中的应用,单一集中式控制器的单点失效、负载过大等问题突显出来,研究学者们提出分布式控制的可扩展控制平面解决方案。分布式控制方式是多个控制器共同协调管理交换机,而控制器个数和位置影响网络的性能,从而如何部署分布式控制器成为目前 SDN 研究热点之一。

根据 SDN 网络应用场景的需求,依据性能尺度对控制器部署问题进行求解。当求解选取的性能尺度不同时,求解的结果也存在明显差异^[3]。在不同需求下如何确定控制器个数和位置、交换机与控制器之间的关联是控制器部署的研究关键,已经证明是一个 NP 难问题^[4]。求解控制器部署问题的经典算法包括启发式算法^[5-6]、基于最小割算法^[7]、贪婪搜索算法^[8-9]、聚类问题求解算法^[10-13]等。在聚类问题求解算法中,文献[10]首次将控制器部署抽象为聚类问题求解,利用 2 个代价函数分布表示管理域间连接链路和管理域内控制器与交换机间平均传输时延,通过最小化代价来对网络中管理域进行划分,算法只考虑传输时延。文献[11]通过对网络进行子域划分,以改进谱聚类方法来研究控制器的部署问题,提出在时延和容量限制下,负载均衡的 SDN 网络多控制器部署算法,但算法复杂度相对较高,且结果比较随机。文献[12]利用 K-critical 方法求解,通过选择出来的控制器构建高健壮性和负载均衡的控制网络,存在节点跨域问题。文献[13]提出层次聚类 $K * -means$ 算法最小化传输时延和控制器负载均衡,但未考虑可靠性问题。目前关于 SDN 分布式控制器部署研究从时延、可靠性、负载等性能指标进行研究,仍存在求解方法复杂、优化目标和约束条件过于单一等问题。在网络拓扑结构已知情况下,采用聚类思想,从提高网络弹性和可靠性两方面优化目标,提出两阶段部署控制器(TSCP, two-stage controller placement)算法:利用节点相似度划分控制域,增强控制域的网络弹性;选择控制路径平均失效率最小的控制器集合作为控制器的部署,提高网络的可靠性。

1 问题描述及模型

1.1 问题描述

SDN 分布式控制器部署是考虑网络拓扑结构,根据相关需求,确定控制器的个数和位置,并将网络中的交换机划分到相应控制器下,形成多个控制域。控制器部署的位置实际上是某个交换机的位置附近,该交换机与控制器的传播时延可以忽略。假设 SDN 抽象为无向图 $G=(S, E, W)$ 中, S 为网络中交换机节点的集合, E 为连接网络中交换机的无向边的集合, W 为交换机之间传播时延大小的集合,其元素 $w(i, j)$ 表示交换机节点 i 与节点 j 之间时延大小,取 0 表示两者不可达,反之亦然。建立相应邻接矩阵 A ,元素 $a(i, j)$ 表示如下

$$a(i, j) = \begin{cases} 1 & w(i, j) > 0, \text{ s.t. } i, j \in S, \\ 0 & \text{others,} \end{cases} \quad (1)$$

其中: $a(i, j) = 1$ 表示节点 i 和节点 j 相邻, $a(i, j) = 0$ 表示两者不相邻。

为了便于描述问题,用 C 表示控制器节点集合,且 $C \subseteq S$ 。 $d(s, c)$ 表示网络中交换机 s 到控制器 c 的最小传播时延, σ 表示交换机与控制器之间传播时延的上限, δ 表示控制器与控制器的传播时延上限, $P(i, j)$ 表示交换机节点 i 与节点 j 之间最短路径。

1.2 部署模型

分布式控制器部署是确定控制器的个数和位置、交换机与控制器之间关系,与控制器容量大小、网络设备(交换机或控制器)之间传播时延大小等相关。

1.2.1 交换机与控制器之间关联

考虑一个交换机只受一个控制器控制,交换机与控制器之间的关系用二维数组 $X_{|S| \times |C|}$ 表示,其中 $|S|$ 、 $|C|$ 分别是网络中交换机和控制器的个数,定义如下

$$x(s, c) = \begin{cases} 1 & c \rightarrow s, \\ 0 & \text{others,} \end{cases} \quad (2)$$

式(2)中 $x(s,c)=1$ 表示交换机 s 受控于控制器 c ,反之,表示两者不存在关系。

1.2.2 相邻控制域

控制域是描述控制器所控制的交换机集合,即控制器 c 对应的控制域 $D(c)$ 表示为 $D(c)=\{s \mid x(s,c)=1, \forall s \in S\}$ 。

相邻控制域是指 2 个控制域的交换机之间存在链路,用 $n(i,j)$ 表示

$$n(i,j)=\begin{cases} 1 & \text{if } w(u,v) \neq 0 \quad \exists u \in D(i), v \in D(j), \\ 0 & \text{others,} \end{cases} \quad (3)$$

式(3)中 $n(i,j)=1$ 表示控制域 i 和控制域 j 相邻,反之,则不相邻。

1.2.3 时延约束

传播时延反映了网络设备之间传递消息的快慢。交换机与控制器的传播时延越小,控制器处理交换机的消息效率越高。控制器部署时尽量选择与它传播时延较小的交换机,需要约束交换机与控制器之间传播时延,即

$$d(s,c) \leq \sigma \quad \forall s \in S, \forall c \in C, \quad (4)$$

控制器与控制器之间需要同步状态等,则相邻控制器之间的传播时延不能过大,否则影响网络的性能。控制器 u 和控制器 v 的传播时延大小约束如下

$$d(u,v) \leq \delta \quad \forall u, v \in C, \quad n(u,v)=1. \quad (5)$$

1.2.4 控制域规模

假设网络中部署同种类型的控制器,则控制器的容量相同。控制器容量反映了可管理的交换机个数上限。为了均衡控制域的规模,需约束控制域的交换机个数,即

$$\eta_L \leq \sum_{s \in S} x(s,c) \leq \eta_U, \quad \forall c \in C, \quad (6)$$

式(6)中 η_L, η_U 分别为控制域的规模下限和上限。

2 TSCP 算法

两阶段控制器部署算法(TSCP, two-stage controller placement)是优化网络弹性和可靠性,分为控制域划分和控制器部署两阶段。控制域划分是先利用节点相似度划分控制域,且满足相关约束,形成初始的控制域;再通过分裂控制域和迁移节点,均衡控制域内交换机个数,优化控制域。控制器部署是在满足相关约束条件下选择可靠性强的节点作为控制器的位置,可靠性用控制路径平均失效率衡量。

2.1 相似度定义

网络划分中节点间的相似度是反映网络中节点连接的紧密程度或者节点间的关联程度。定义节点 i 的邻居节点集(包括自身节点 i)记为 $N(i)$,即

$$N(i)=\{j \mid a(i,j)=1\} \cup \{i\}. \quad (7)$$

节点 i 的权重记为 $nw(i)$,表示节点 i 的所有邻接边的权重倒数之和,即

$$nw(i)=\sum_{\substack{k \in N(i) \\ k \neq i}} \frac{1}{w(i,k)}. \quad (8)$$

节点 i 与 j 的关联权重记为 $nw(i,j)$,表示节点 i 与 j 的所有公共邻居节点之间的相邻边的权重倒数之和,即

$$nw(i,j)=\sum_{\substack{u,v \in N(i) \cap N(j) \\ u < v}} \frac{1}{w(u,v)}. \quad (9)$$

当节点 i 与 j 拥有较多的相同的邻居节点或者距离较近时,其相似度相对较大。对于任意节点 i 与 j ,如果它们直接相连,则存在较大的相似度,且其相似度随节点 i 与 j 之间的距离(即权重 $w(i,j)$)增大而减小;如果它们不是直接相连,只是存在共同的邻接点,则节点 i 与 j 的相似度与邻接节点相关。权重倒数 $1/w(i,j)$ 和 $nw(i,j)$ 分别反映了距离越小相似度越大以及公共节点个数越多相似度越大的关系,结合文献[14]提出的节点相似度模型,提出新的基于权重的节点相似度定义

$$\text{Sim}(i, j) = \begin{cases} \frac{\frac{1}{w_{ij}} * n\omega(i, j)}{n\omega(i) + n\omega(j) - \frac{1}{w_{ij}}}, & i \leftrightarrow j, \\ \prod_{\langle u, v \rangle \in P_{ij}} \text{Sim}(u, v), & i \rightarrow j, \\ 0, & \text{others,} \end{cases} \quad (10)$$

式(10)中 P_{ij} 表示节点 i 与 j 的最短路径, $\langle u, v \rangle$ 表示路径 P_{ij} 上有序的节点对。式(10)从 2 节点直接相连、不相连但存在路径和不存在路径 3 种情况定义 2 节点间的相似度: 当节点 i 与 j 直接相连时, 即权重 $w(i, j)$ 不为 0, $1/w(i, j)$ 说明节点间距离越大其相似度越小, $n\omega(i, j)$ 反映了公共节点越多其值相对较大。可简单证明分母是大于等于分子的, 因分子中 $n\omega(i, j)$ 表示 2 个邻居节点集的交集, 分母反映两者的并集。同时 $1/w(i, j)$ 值小于等于 1, 从而证明了整个分式的取值范围为 $(0, 1]$ 。当 2 节点不是直接相连时, 但节点 i 与 j 存在路径 P_{ij} , 其相似度与路径 P_{ij} 上的节点有关, 且任意两节点的相似度相互独立, 则选择其最短路径上的节点间的相似度的乘积作为两节点的相似度。当两节点不可达时, 其相似度为 0。理论证明, 式(10)满足非负性、自反性、传递性等特性, 且取值区间在 $[0, 1]$, 满足相似度定义。

节点间的相似度描述了节点之间的紧密关系, 则推广到节点与控制域之间的相似度定义。即节点 i 与控制域 $D(k)$ 的相似度为^[14]

$$\text{Sim}(i, D(k)) = \sum_{j \in D(k)} \text{Sim}(i, j). \quad (11)$$

2.2 基于相似度的控制域划分算法

2.2.1 初始划分

初始划分采用聚类思想, 初步形成控制域, 尽量使得控制域的交换机个数最大化, 从而减少控制器个数。

初始划分过程:

Step 1: 计算节点间的相似度; 初始迭代次数 N ; 初始队列 queue 为空; 初始交换机节点遍历标志为 false;

Step 2: 获取任意边缘节点, 以节点 i 形成第一个控制域 $D(i)$;

Step 3: 依次遍历与控制域 $D(i)$ 相邻的节点, 选择与它相似度最大的节点 j , 且满足条件: 1) 控制域 $D(i)$ 与节点 j 新形成的控制域 $D(i)$ 中存在某节点 v , 使得所有节点到某节点 v 的传播时延均小于等于上限 σ ; 2) 新形成的控制域 $D(i)$ 的交换机总个数满足上限 η_U , 则将节点 j 加入控制域 $D(i)$, 标记节点 j 为 true。重复执行, 直到没有任何节点可加入控制域 $D(i)$ 。

Step 4: 选择与控制域 $D(i)$ 相邻的所有节点, 加入队列 queue 中;

Step 5: 若队列 queue 为空, 跳到 Step 7; 否则, 出队列, 选择标记为 false 的节点 i 且满足当前节点与相邻控制域的控制域传播时延满足上限 δ , 形成下一个控制域 $D(i)$ 。

Step 6: 依次遍历与控制域 $D(i)$ 相邻节点 j ; 若节点 j 已标记, 判断节点 j 与控制域 $D(i)$ 的相似度是否大于与其他控制域的相似度, 从中选择相似度最大的节点 j , 且满足条件: 1) 控制域 $D(i)$ 与节点 j 新形成的控制域 $D(i)$ 中存在某节点 v , 使得所有节点到某节点 v 的传播时延均小于等于上限 σ ; 2) 满足条件(1)节点的同时需要满足控制域 $D(i)$ 与所有相邻域 $D(k)$ 对应 2 个控制器之间传播时延不大于上限 δ ; 3) 新形成的控制域 $D(i)$ 的交换机总个数满足上限 η_U , 则将节点 j 加入控制域 $D(i)$, 并从其他控制域中删除, 标记节点 j 为 true。重复执行, 直到没有任何节点可加入控制域。跳到 Step 4;

Step 7: 重置 queue 为空, 以及所有交换机节点标志为 false;

Step 8: 重复执行 N 次 Step 2 到 Step 7; 选择最优的划分结果, 需要优先选择控制域个数最少的, 其次选择控制域规模比较均衡的(选择控制器容量利用率方差最小的);

Step 9: 输出结果, 算法结束。

初始划分结果可能存在控制域的规模小于下限 η_L 情况,需要优化控制域。

2.2.2 优化划分

优化划分是为了减少控制域个数,平衡控制域规模。划分采用分裂和迁移方法,分裂是将可能重叠的控制域的所有交换机分配到邻居控制域,减少控制域个数。迁移是从相邻控制域中迁移部分节点到当前控制域,增加当前控制域的交换机个数,平衡各个控制域的规模。

1) 分裂控制域

Step 1: 对初始划分结果按控制域的规模升序排序;

Step 2: 遍历控制域 $D(i)$, 计算当前控制域的相邻控制域;

Step 3: 若控制域 $D(i)$ 的节点划分到其他相邻控制域,使得所有相邻控制域的交换机个数大于上限 η_U , 则分裂失败,跳到 Step 5;

Step 4: 将控制域 $D(i)$ 的所有节点进行集合划分,划分子集个数等于相邻控制域的个数,采用递归思想实现;

Step 5: 遍历所有集合划分结果,判断每个集合划分中的子集节点是否可以划分到其他相邻控制域 $D(k)$,且新形成的控制域 $D(k)$ 满足传播时延上限 σ 、时延上限 δ 、控制域的规模上限 η_U 3 个约束条件,若满足,则该集合划分满足条件,分裂成功。更新控制域集合,结束遍历;否则,继续遍历直到结束;

Step 6: 若控制域遍历未结束,继续 Step 2;

Step 7: 分裂结束。

分裂可以减少控制域的个数,仍可能存在规模小的控制域,则需从相邻控制域中迁移部分节点,使得控制域的交换机个数均衡。

2) 迁移节点

Step 1: 控制域的规模升序排序;

Step 2: 遍历控制域 $D(i)$, 判断相邻控制域 $D(k)$ 的节点,对于 $D(k)$ 中节点 j , 满足条件: 1) 控制域 $D(i)$ 与节点 j 新形成的控制域 $D(i)$ 中存在某节点 v , 使得所有节点到某节点 v 的传播时延均小于等于上限 δ ; 2) 满足条件(1)的节点同时需要满足控制域 $D(i)$ 与所有相邻域对应 2 个控制器之间传播时延不大于上限 δ ; 3) 新形成的控制域 $D(i)$ 的交换机总个数满足上限 η_U ; 选择与 $D(i)$ 相似度最大的节点迁移到 $D(i)$, 且保证被迁移控制域 $D(k)$ 节点的连通性; 对于所有满足条件的节点, 构成迁移节点集;

Step 3: 遍历迁移节点集, 选择与当前控制域相似度最大的节点构成新的控制域, 且迁移节点个数至少使得控制域的容量不小于下限 η_L , 均衡相邻控制域的规模。直到遍历结束;

Step 4: 若控制域遍历未结束, 继续 Step 2;

Step 5: 迁移结束。

经过 2 次控制域划分, 控制域内结点连接紧密, 控制域交换机个数最大化, 控制域个数最小化。

2.3 基于可靠性的控制器部署算法

2.3.1 可靠性描述

SDN 中网络设备的控制平面和数据平面的控制消息传递主要通过控制路径, 网络中交换机与控制器的控制消息、控制器与控制器之间的状态同步消息都需要通过控制路径传递。当某条控制路径失效时, 中断的交换机可能处于非控制状态, 控制器与其他控制器的消息同步失败。因此, 控制路径失效越多, 控制流量中断的可能性越大。控制器的个数和位置不同, 控制路径也不同, 影响网络的可靠性。

用控制路径失效率描述网络的可靠性, 失效率越小, 可靠性越大。控制路径的失效率与控制路径上的节点和链路有关, 任意节点或链路失效, 控制路径失效。所指的控制路径是控制器与交换机之间、控制器与控制器之间的最短路由。

定义节点 i 和节点 j 通过路径 $P(i, j)$ 连通失效的概率

$$q(i, j) = 1 - \prod_{\substack{\forall v, e \in P(i, j) \\ v \in S, e \in E}} (1 - r_v)(1 - r_e), \quad (12)$$

式(12)中 r_e 表示链路 e 失效率, r_v 表示节点 v 失效率。

网络控制路径平均失效率是两部分控制路径失效率之和的均值,包括控制器与控制器之间的控制路径失效率和控制域内控制器与交换机的控制路径失效率。其值越小,网络可靠性越强。网络控制路径平均失效率 ρ

$$\rho = \frac{1}{m} \left[\sum_{u \in C} \sum_{\substack{v \in C \\ u < v}} q(u, v) + \sum_{u \in C} \sum_{i \in D(u)} q(u, i) \right], \quad (13)$$

式(13)中 m 表示控制路径数,等于控制器与控制器之间的控制路径数和控制域内控制器与交换机之间的控制路径数之和,即 $m = (|C|^2 - |C|)/2 + |S|$ 。

2.3.2 控制器部署算法

考虑到控制域内的控制节点选择不同,控制路径不同,网络控制路径平均失效率 ρ 不同,网络可靠性也不同,从而控制器节点的选择对网络可靠性影响较大。控制域划分阶段结束之后,形成相对独立的控制域,根据相关需求,选择最优的节点作为控制域的控制部署位置。当前控制域的控制选择影响其他控制域的控制选择,所有控制域对应的控制器是一一对应关系。假设 3 个控制域,分别从各自控制域的候选控制器节点选择一个节点,若节点 i, j, k 满足控制器之间的时延约束,而节点 i, j, l 不满足,说明所有控制域的控制节点是一一对应,只有满足约束条件的每个控制域对应控制器节点组合才是可行解,可能存在多组。控制器部署需要先计算控制域中满足交换机与控制器之间传播时延约束的控制器候选节点,再根据相邻控制域的控制选择,计算各个控制域可行的控制器节点组合构成一个可行解,最后计算每一个可行解,从中选择可靠性最强的可行解作为控制器的部署位置,完成控制器的部署。

控制器部署算法:

Step 1: 初始划分形成控制域,初始化每个控制域的候选控制节点集为空;初始化可行解集合,且每个可行解是所有控制域的控制节点;

Step 2: 遍历每个控制域,计算所有满足交换机与控制器之间传播时延上限 σ 约束的节点集,加入其候选控制节点集;直到遍历结束;

Step 3: 递归计算每个控制域可行的控制器节点组合:遍历控制域 $D(i)$ 的候选控制节点 u ,递归计算每个相邻控制域 $D(j)$ 满足相邻控制域的控制选择与控制器之间的传播时延上限 δ 约束节点,依次存入可行解集合。

Step 4: 遍历每个可行解,计算控制路径平均失效率 ρ 值,选择 ρ 最小的可行解作为控制器的部署位置;直到遍历结束;

Step 5: 输出结果,算法结束。

2.4 性能指标

为了验证部署模型和 TSCP 算法的有效性,提出相关性能指标分析控制器部署结果。

1) 控制器容量利用率

控制器容量是控制器允许连接交换机的最大个数(即上限 η_u),用控制器容量利用率衡量控制域的规模均衡性,等于控制域的交换机个数与控制器容量上限 η_u 的比值。控制器 c 容量的利用率 $R(c)$ 表示如下

$$R(c) = \frac{\sum_{s \in S} x(s, c)}{\eta_u}, \quad \forall c \in C. \quad (14)$$

控制器容量利用率值越大,控制器控制的交换机个数越多,整个网络所需控制器个数越少。控制器容量利用率越高,则控制域的规模越均衡。

控制器容量利用率方差可以衡量全局的控制域规模均衡性,则

$$H = \frac{1}{|C|} \sum_{c \in C} (R(c) - \bar{R})^2, \quad (15)$$

式(15)中 \bar{R} 表示控制器容量利用率平均值。

2) 连通强度

网络中节点之间的连接边越多,网络的连通性越强,弹性越强。连通强度是指控制域内边的数量与这些

节点在网络中的边的数量的比值,其值越大,表示控制域内节点联系越紧密,连通性越强。控制域 $D(c)$ 的连通强度 $K(c)$

$$K(c) = \frac{\sum_{i \in D(c)} \sum_{j \in D(c)} a(i, j)}{\sum_{i \in D(c)} \sum_{\substack{j \in S \\ i < j}} a(i, j)} \quad \forall c \in C. \quad (16)$$

网络弹性等于各个控制域的连通强度的平均值,即

$$\varphi = \frac{1}{|C|} \sum_{c \in C} K(c). \quad (17)$$

3) 网络覆盖时延

网络覆盖时延表示整个网络的控制时延,其值越小,网络的整体性能越强。网络覆盖时延是指所有控制器覆盖所属交换机的传播时延之和,再加上控制器与控制器之间同步状态所需时延。网络覆盖时延 T 如下

$$T = \sum_{c \in C} \sum_{s \in D(c)} d(s, c) + \sum_{\substack{u \in C \\ u < v}} \sum_{v \in C} d(u, v). \quad (18)$$

2.5 算法理论分析

考虑到度较大的节点作为控制器位置的可能性较大,提出一种基于贪婪策略的控制器部署(GCP, greedy controller placement)算法,其算法思想是依次选择未标记节点中度最大的节点作为控制器的位置,在满足相关约束条件下,将距离最近的节点划分到控制域,多次迭代,直到所有节点完成划分。

K^* -means 算法^[13]是基于最小化交换机与控制器之间时延、平衡控制器负载的网络划分算法。算法基本思想:建立时延和负载的目标函数,初始化一定数量的中心节点,划分交换机节点到离聚类中心最近的类中,合并目标函数最小的两个聚类,多次迭代直到结束。

K^* -means 算法、GCP 算法都是划分交换机到距离最近的控制域中,没有考虑节点之间的关系;而 TSCP 算法以节点相似度为划分条件,将距离最近、关联紧密的节点划分到一个控制域,实现时延最小化和网络弹性最大化。三种算法都考虑负载均衡, K^* -means 算法将负载均衡作为目标函数最小化,GCP 算法、TSCP 算法通过约束控制域的规模实现负载均衡,但 GCP 算法只考虑规模界限,而 TSCP 算法在初始划分基础上,通过分裂和迁移机制的优化划分,负载均衡效果优于 GCP 算法。

K^* -means 算法利用控制域内最小最短路径确定控制器位置,GCP 算法是选择节点度最大的节点为控制器,TSCP 算法在优化划分基础上,引入可靠性指标,选择可靠性高的位置部署控制器。

对比三种算法的控制域划分和控制器位置确定的过程,从算法理论分析可知,TSCP 算法相比其他 2 种算法,能优化网络时延、网络弹性、可靠性 3 个性能指标,整体效果较好。

3 实验结果与分析

为了验证算法的有效性和性能,笔者用开源的真实网络 GEANT 网络(22 个节点、36 条链路)、INTERNET2(34 个节点、42 条链路)网络数据,对比文献[13]的 K^* -means 算法、GCP 算法、TSCP 算法的多项性能指标:控制器个数 Φ 、控制器容量利用率方差 H 、网络弹性 φ 、网络覆盖时延 T 、网络可靠性 ρ 。实际网络中,利用节点的经度和纬度计算节点之间的距离描述节点之间的传播时延。在不同的网络拓扑运行部署算法,验证算法的普遍性和适用性。实验算法采用 Java 语言编写,在电脑(CPU Intel Xeon 3.3G 8 核,内存 16G,Win10 操作系统)的运行环境实现。

3.1 TSCP 算法分析

为了验证算法的稳定性和收敛性,对传播时延、控制域的规模参数进行实验分析。失效参数 r_e 、 r_o 只影响网络可靠性 ρ 值,由式(12)可知,失效参数值的变化与 ρ 值成反比,但控制器位置选择是不变的,故这里不再分析。

控制域的规模下限 η_L 会影响控制域负载均衡,从图 1 看出 η_L 小于 3 时,控制器容量利用率方差 H 值较高,表明控制域的规模不均衡;当大于 4 时, H 值再无变化,控制域的规模均衡,且不随 η_L 值增大而变化。

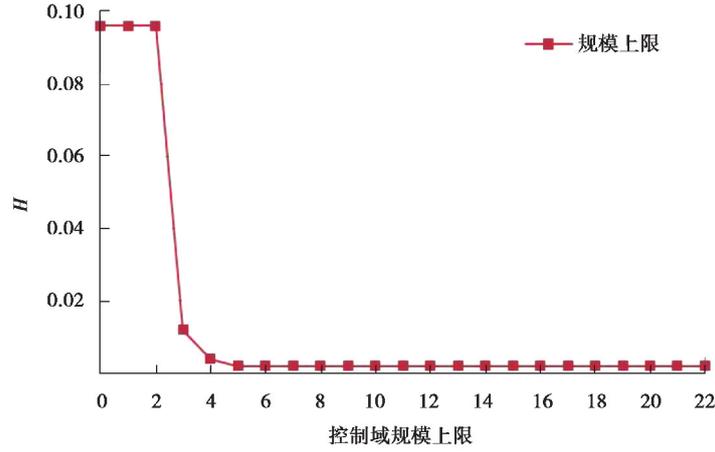


图 1 GEANT 网络参数 η_L 分析
 Fig. 1 Parameter analysis of η_L in GEANT

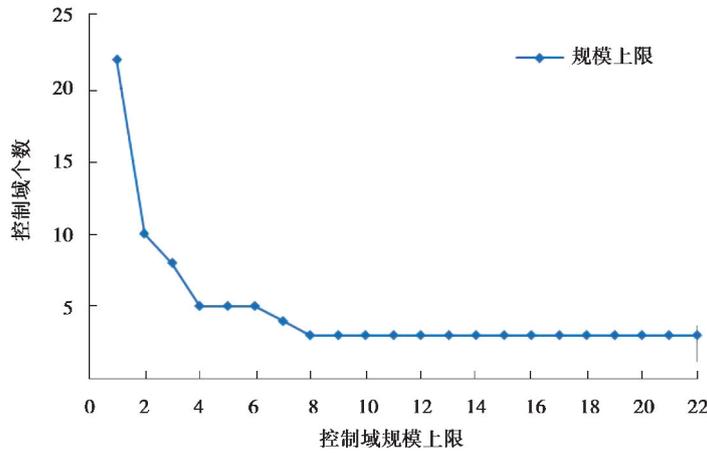


图 2 GEANT 网络参数 η_U 分析
 Fig. 2 Parameter analysis of η_U in GEANT

控制域的规模上限 η_U 会影响控制域划分个数,从图 2 可知 η_U 小于 4 时,划分较多的控制域;当大于 8 时,控制域个数再无变化,说明控制域划分恒定,不随 η_U 值变化。

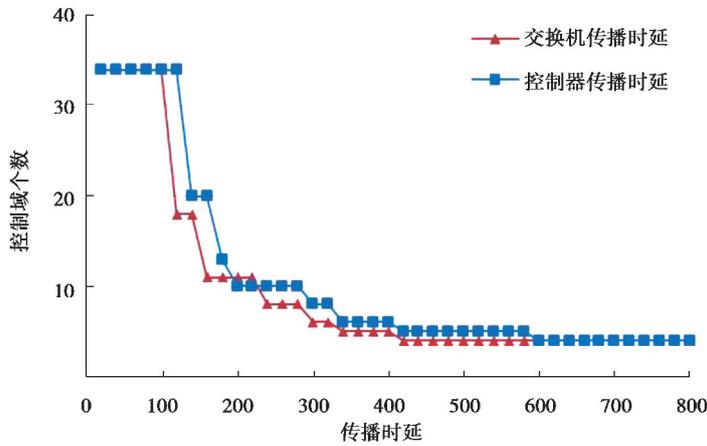


图 3 INTERNET2 网络传输时延分析
 Fig. 3 Parameter analysis of propagation delay in GEANT

交换机与控制器传播时延上限 σ , 控制器与控制器的传播时延上限 δ 影响控制域划分个数。图 3 中得知 σ 、 δ 值小于 240 时, 控制域个数较多; 随着时延上限增大, 单个控制域中交换机个数增加, 控制域个数减少。当 σ 大于 400, δ 大于等于 600 时, 控制域个数不变化。

上述 3 个实验结果表明, 随着传播时延、控制域的规模参数增加, 控制域划分结果逐渐变化; 变化到一定值时, 控制域划分结果不再变化, 表明 TSCP 算法是收敛且稳定的。

通过对传播时延、控制域的规模参数多次测试, 根据上述收敛性结论, 选择算法收敛时的参数值作为后续测试。则设置参数: 传播时延上限 $\sigma=400$, 播时延上限 $\delta=600$, 控制域的规模下限 $\eta_L=4$, 规模上限 $\eta_U=10$ 。失效率取值范围一般为 $(0, 0.1]$, 为了一致性, 节点失效率和链路失效率取值相同, 通常 $r_v=0.05$, $r_e=0.05$ 。

3.2 控制域划分分析

控制器容量利用率 R 反映控制域内交换机个数与控制器容量上限的比值, 其值越大说明控制域的规模越大, 控制器的利用率越高。网络所有控制器容量利用率均较高时, 所需控制器的个数较少。

表 1 控制器容量利用率 R 分析

Table 1 Analysis of R

Net	Algorithm	D(1)	D(2)	D(3)	D(4)	D(5)	D(6)
GEANT (22/36)	K * -means	0.7	0.8	0.7	—	—	—
	GCP	0.6	0.4	0.7	0.5	—	—
	TSCP	0.7	0.7	0.8	—	—	—
INTERNET2 (34/42)	K * -means	0.7	0.8	0.6	0.6	0.7	—
	GCP	0.5	0.5	0.9	0.6	0.4	0.5
	TSCP	0.8	0.8	0.9	0.9	—	—

从表 1 可以看出 TSCP 算法计算结果中每个控制域的规模是相对均衡的, 利用率至少大于等于 70%, 控制域个数最少, 达到算法目的。GCP 算法只考虑节点的度, 容易出现边缘节点, 导致控制域构成不平衡, 且控制域个数多。K * -means 算法考虑负载均衡, 控制器利用率相差不大, 但是 INTERNET2 网络中控制域个数比 TSCP 算法多。

表 2 控制域连通强度 K 分析

Table 2 Analysis of K

Net	Algorithm	D(1)	D(2)	D(3)	D(4)	D(5)	D(6)
GEANT (22/36)	K * -means	0.38	0.54	0.61	—	—	—
	GCP	0.5	0.38	0.6	0.67	—	—
	TSCP	0.67	0.62	0.69	—	—	—
INTERNET2 (34/42)	K * -means	0.7	0.5	0.55	0.63	0.78	—
	GCP	0.5	0.5	0.5	0.63	0.5	0.5
	TSCP	0.69	0.57	0.62	0.8	—	—

表 2 可以看出,由于 TSCP 算法在控制域划分时引入节点相似度,将相似度较大的节点划分到同一个控制域,控制域的连通性较强,连通强度值均大于等于 0.57。说明 TSCP 算法的控制域内节点联系紧密,交换机与控制器之间可以存在多条备份路径,提高网络的弹性。K * -means 算法未考虑,使得可能存在某些控制域内节点之间连接不紧密,其效果欠佳。

3.3 控制器位置分析

为了对比 INTERNET2 网络的实际部署结果^[15](采用 4 个控制器控制整个网络),研究采用 TSCP 算法部署结果如图 4。

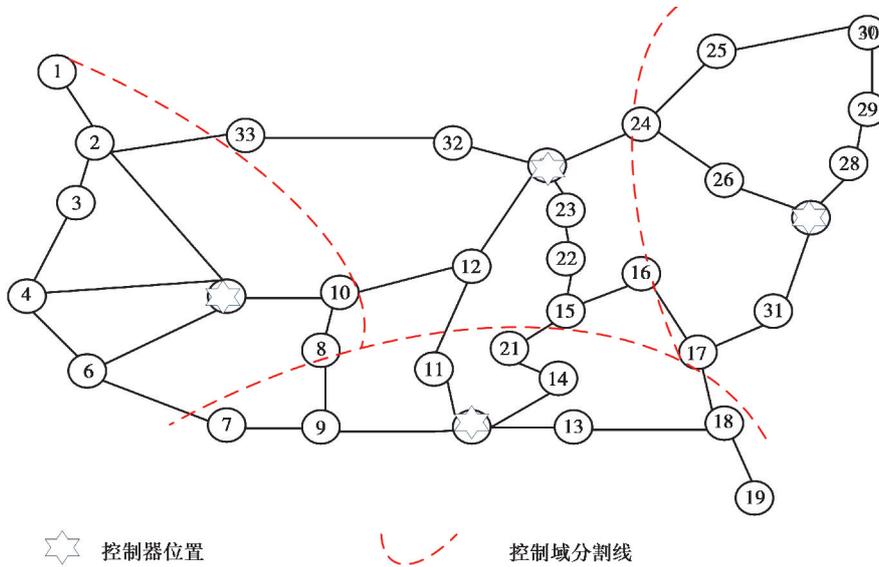


图 4 INTERNET2 网络的部署结果示意
Fig. 4 Deployment result in INTERNET2

图 4 中 TSCP 算法在节点 5、20、27、34 分别部署控制器,所需控制器个数为 4 个,与 INTERNET2 网络中实际部署控制器个数一致,每个控制域中的节点数为 8 到 9 个,控制域的规模相对均衡。控制器节点分别选择节点 5、20、27、34,其控制路径平均失效率 $\rho=0.235$,网络可靠性较高。

3.4 控制器部署结果综合分析

为了更全面验证 TSCP 算法的有效性,分别从控制器个数 Φ 、控制器容量利用率方差 H 、网络弹性 φ 、网络覆盖时延 T 、网络可靠性 ρ 的性能指标测验并分析。性能指标具体内容如下

表 3 性能指标对比

Table 3 Performance index comparison

Net	Algorithm	Φ	H	φ	T	ρ
GEANT (22/36)	K * -means	3	0.002	0.51	164	0.27
	GCP	4	0.013	0.54	245	0.28
	TSCP	3	0.002	0.66	169	0.19
INTERNET2 (34/42)	K * -means	5	0.006	0.63	284	0.30
	GCP	6	0.026	0.52	436	0.37
	TSCP	4	0.003	0.67	277	0.26

控制器个数 Φ 与控制域个数一致,选择较少的控制器部署有利于减少网络资源成本。从性能指标 Φ 可知,TSCP 算法通过初始划分和优化划分 2 步,利用节点相似度的原理,将关联更紧密的交换机划分到

一个控制域,减少控制域个数。而 GCP 算法、K*-means 算法未考虑节点关联性,控制域个数比 TSCP 算法较多。

控制器容量利用率方差 H 能反映全局控制域的规模均衡性,其值越小说明各个控制域的规模越均衡。TSCP 算法和 K*-means 算法均考虑负载均衡性, H 值较小,说明 TSCP 算法能满足负载均衡要求。而 GCP 算法未考虑负载均衡指标, H 值较大。在 INTERNET2 网络中 K*-means 算法控制域个数为 5,控制器容量利用率最大和最小相差为 0.2,影响 H 值。

网络弹性 φ 值的大小说明控制域内处理网络故障的能力,其值越大,节点连通性越强,交换机与控制器之间容错率越高,备份路径越多。从表中可以看出,TSCP 算法的 φ 值均最大,均大于等于 0.66,验证节点相似度的有效性,说明 TSCP 算法能提高网络的弹性。GCP 算法以最大节点度为控制器位置,其网络弹性效果在 GEANT 网络中比 K*-means 算法较好。

网络覆盖时延 T 是衡量控制器部署的重要指标之一,反映了控制器部署的好坏。其值越小,网络的整体性能越强。K*-means 算法以时延为优化目标,其 T 值较小。而 TSCP 算法中节点相似度把时延作为最短路径指标引入,相似度越大,其时延较小。从而说明 TSCP 算法通过节点相似度划分控制域,能使网络覆盖时延较小。在 INTERNET2 网络中 K*-means 算法控制域个数比 TSCP 算法多,增加了控制器之间时延,影响网络覆盖时延。

控制路径平均失效率 ρ 衡量网络可靠性,值越小,整个网络可靠性越高。表 3 中数据得知 GCP 算法的可靠性较差,K*-means 算法其次,TSCP 算法可靠性较高。GCP 算法和 K*-means 算法实现控制器部署时并没有考虑控制路径的失效率。TSCP 算法均衡划分控制域,减少控制器个数,从每个控制域中选择全局性的控制路径平均失效率最小的控制器集合确定控制器的位置,提高网络可靠性。

上述实验结果说明 TSCP 算法通过两阶段划分策略,可以优化控制域的规模,均衡控制域的交换机个数,减少控制器个数,网络弹性和可靠性均表现较好。总之,TSCP 算法与其他算法相比,对于 SDN 控制器部署具有一定的优势和效果。

4 结 语

只考虑网络拓扑,以优化网络弹性和可靠性为目标,提出两阶段控制器部署方案。首先,分析了约束控制器部署的相关因素,如控制域规模、网络设备(交换机或控制器)之间的传播时延,建立对应的部署模型;其次,定义了节点相似度,提出了控制域划分:初始划分和优化划分。再次,基于控制路径失效率计算可靠性最大的控制器部署位置。最后,提出相关性能指标用于测试算法,举例分析算法的可行性。基于两阶段的控制器部署算法可以有效地提高网络控制器利用率,平衡控制域的交换机个数,尽量减少控制器的个数。然而实际 SDN 网络是流量动态变化的,如何通过引入流量矩阵实现控制器的动态部署是未来要做的工作。

参考文献:

- [1] McKeown N. Software-defined networking[J]. Infocom Keynote Talk,2009,17(2):30-32.
- [2] 张朝昆,崔勇,唐嵩祎,等. 软件定义网络(SDN)研究进展[J]. 软件学报,2015,26(1):62-81.
ZHANG ChaoKun, CUI Yong, TANG HeYi, et al. State-of-the-art survey on software-defined networking (SDN)[J]. Journal of Software, 2015, 26(1): 62-81. (in Chinese)
- [3] 高先明,王宝生,邓文平,等. SDN 网络中控制器放置问题综述[J]. 通信学报,2017,38(7):155-164.
GAO Xianming, WANG Baosheng, DENG Wenping, et al. Survey of controller placement problem in software defined network[J]. Journal on Communications, 2017, 38(7): 155-164. (in Chinese)
- [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 Press, 2012: 7-12.
- [5] 杨耀通,汪清,高丽蓉,等. 基于蝙蝠算法的 SDN 多控制器部署[J]. 重庆大学学报,2018,41(9):57-65.
YANG Yaotong, WANG Qing, GAO Lirong, et al. A bat inspired controller placement algorithm in software defined

- network[J]. Journal of Chongqing University, 2018, 41(9): 57-65.(in Chinese)
- [6] Narmanlioglu O, Zeydan E. Network virtualization for mobile operators in software-defined based LTE networks[C/OL]. 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). Piscataway, NJ: IEEE, 2017 (2017-07-24)[2020-05-25]. <https://doi.org/10.23919/INM.2017.7987334>
- [7] Zhang Y, Beheshti N, Tatipamula M. On resilience of split-architecture networks [C/OL]. 2011 IEEE Global Telecommunications Conference Houston. Piscataway, NJ: IEEE, 2011(2012-01-19)[2020-05-25]. <https://doi.org/10.1109/GLOCOM.2011.6134496>.
- [8] Riggio R, Marina M K, Rasheed T. Interference management in software-defined mobile networks[C/OL]. 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). Piscataway, NJ: IEEE, 2015(2015-07-02)[2020-05-25]. <https://doi.org/10.1109/INM.2015.7140347>
- [9] Hu Y N, Wang W D, Gong X Y, et al. On reliability-optimized controller placement for software-defined networks[J]. China Communications, 2014, 11(2): 38-54.
- [10] Zhang Y, Beheshti N, Tatipamula M. On resilience of split-architecture networks [C/OL]. 2011 IEEE Global Telecommunications Conference-GLOBECOM 2011. Piscataway, NJ: IEEE, 2011(2012-01-19)[2020-05-25]. <https://doi.org/10.1109/GLOCOM.2011.6134496>.
- [11] 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 Kuangyu, HUANG Chuanhe, WANG Caihua, et al. Balanced multiple controllers placement with latency and capacity bound in software-defined network[J]. Journal on Communications, 2016, 37(11): 90-103.(in Chinese)
- [12] Aoki H, Shinomya N. Controller placement problem to enhance performance in multi-domain SDN networks[C]// The Fifteenth International Conference on Networks. Lisbon, Portugal: IARIA, 2016: 108-109.
- [13] Kuang H L, Qiu Y W, Li R F, et al. A hierarchical K-means algorithm for controller placement in SDN-based WAN architecture [C] // 2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). Piscataway, NJ: IEEE, 2018(2014-04-16)[2020-05-25]. <https://doi.org/10.1109/ICMTMA.2018.00070>
- [14] 王坤, 吕光宏, 梁召伟, 等. 基于相似度的加权复杂网络社区发现方法[J]. 四川大学学报(自然科学版), 2014, 51(6): 1170-1176.
- WANG Kun, LV Guanghong, LIANG Zhaowei, et al. Detecting community in weighted complex network based on similarities[J]. Journal of Sichuan University(Natural Science Edition), 2014, 51(6): 1170-1176.(in Chinese)
- [15] 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 Press, 2012: 7-12.

(编辑 侯 湘)