

SDN 网络中控制器放置问题综述

高先明, 王宝生, 邓文平, 陶静

(国防科学技术大学计算机学院, 湖南 长沙 410073)

摘要: 为了改变当前 Internet 网络管理复杂度高的局面, 提出软件定义网络 (SDN, software defined network), 通过控制器生成流策略来直接控制数据流的转发行为。随着 SDN 网络部署与实践, 研究人员发现控制器在网络中部署位置能直接影响到网络性能。近年来, 围绕控制器放置问题的研究已逐步展开, 研究内容集中在性能尺度以及搜索算法这 2 个方面。结合当前研究现状, 对已有的控制器放置问题的研究进行系统性归纳总结, 为后续进一步研究提供参考。

关键词: 软件定义网络; 多控制器; 控制器放置; 性能尺度; 搜索算法

中图分类号: TP393.0

文献标识码: A

Survey of controller placement problem in software defined network

GAO Xian-ming, WANG Bao-sheng, DENG Wen-ping, TAO Jing

(Department of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: In order to change situation of high management complexity in current Internet, software defined network (SDN) was proposed, which mainly aimed to directly control forwarding behaviors of data-flow by using flow strategies generated by controllers. With the deployment and applications of SDN, research communities found that the controller placement in SDN network could directly affect network performance. In recent years, controller placement problem (CPP) has become a hot topic, where performance metric and searching algorithms are important research areas. Based on current researches, the existing controller placement problem was systematically analyzed and summarized, which was expected to be helpful for the follow-up research.

Key words: software defined network, multi-controller, controller placement, performance metric, searching algorithm

1 引言

早期, Internet 中路由器采用数据平面和控制平面紧耦合的架构, 致使网络设备的升级换代和新业务的部署异常困难, 以尝试对数据平面和控制平面进行物理解耦, 如 ForCES^[1]、ATM 网^[2]与主动网络^[3~5]的初期方案等。为了满足网络的可编程性、新业务快速部署等新需求, 一种新的网络设计理念——软件定义网络 (SDN, software defined network) 被提出。SDN 对数据面和功能面进行功能

解耦, 将底层物理基础设施抽象为可实时部署的网络服务与应用^[6]。SDN 在简化网络管理、业务自动化部署等方面具有显著优势。据行业预计, SDN 市场价值在 2019 年将达到 300 亿美元^[7]。SDN 在中小型数据网络中得到广泛应用, 但在大中型网络实践中面临着网络健壮性和可扩展性等方面的挑战, 因此, 研究人员提出了多控制器 (或分布式控制器) 架构, 如 HyperFlow^[8]、Onix^[9]、Master/slaves^[10]、ASIC^[11]和 Kandoo^[12]等, 使每个控制器负责自己管理域内的交换机, 且控制器间相互协调确保全局网

收稿日期: 2016-12-30; 修回日期: 2017-04-01

基金项目: 国家重点基础研究发展计划 (“973” 计划) 基金资助项目 (No.2012CB315906); 国家自然科学基金资助项目 (No.61373039)

Foundation Items: The National Basic Research Program of China (973 Program) (No.2012CB315906), The National Natural Science Foundation of China (No.61373039)

络状态的一致性。

随着 SDN 网络实践的深入, 研究发现控制器位置影响控制器处理网络事件的能力。例如, 在单控制器的 SDN 网络中, 控制器响应交换机请求的网络时延与部署位置密切相关; 当考虑 SDN 网络健壮性时, 至少需要部署 2 台控制器。Heller 等^[13]在 2012 年首次约束控制器放置问题(CPP, controller placement problem), 指出 CPP 求解的关键点如下。

1) 控制器数量: 指满足网络管理的性能需求时的控制器部署数量。

2) 控制器在网络拓扑中的位置: 指控制器与 SDN 网络中节点的映射关系。

为了描述控制器在网络中放置问题的研究, 形式化地描述 CPP。利用 $G(V, E, S)$ 来表示 SDN 网络拓扑结构, 其中, V 表示网络节点, E 表示网络链路, S 表示控制器, $N=|V|$ 表示节点的数量, $K=|S|$ 表示控制器数量。控制器在网络拓扑中的位置与节点位置间关系为 $S \subseteq V$, 即控制器位于节点的位置。在 CPP 中求解 2 个问题: 1) K 值大小; 2) $S \rightarrow V$ 映射关系。本文主要对控制器放置问题进行系统性的介绍, 分析当前研究现状, 并加以归纳总结, 为后续的研究提供参考。

2 性能尺度

根据 SDN 网络应用场景的需求, 依据性能尺度对 CPP 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^[13,14]。目前, 求解 CPP 的性能尺度包括传输时延、负载、可靠性和开销等。

2.1 传输时延

网络时延包括处理时延、排队时延、传输时延以及发送时延。在网络未拥塞情况下, 排队时延忽略不计; 处理时延与控制器处理能力相关, 通常是固定值 T_p ; 发送时延受数据分组大小及端口速率的影响, 在数据分组大小不可知以及端口速率相同的情况下, 通常是固定值 T_t ; 传输时延与 2 个节点间距离有关, 用 T_m 表示。因此, 求解 CPP 通常仅考虑传输时延。

传输时延包括控制器与交换机间的传输时延、控制器间的传输时延等。传输时延通常用 2 个节点间最短路径的长度 $d(x, y)$ 来衡量。

2.1.1 控制器与交换机间传输时延

控制器与交换机间传输时延包括网络零故障

率时平均传输时延、网络零故障率时最坏传输时延以及网络故障率非零时最坏传输时延。

1) 网络零故障率时平均传输时延

在网络零故障率时, 控制器与交换机间平均传输时延指交换机与控制器间传输时延 $\min d(v, s)$ 的均值 T_{avg} , 如式(1)所示^[13,15,16]。文献[13]首次提出作为解决 CPP 的性能尺度, 用于反映 SDN 网络中控制器与交换机间传输时延的整体情况。但部分控制器与交换机间实际的传输时延要超过平均传输时延, 这导致控制器无法及时地响应网络事件。

$$T_{\text{avg}} = \frac{1}{N} \sum_{v \in V} \min_{s \in S} d(v, s) \quad (1)$$

2) 网络零故障率时最坏传输时延

在网络零故障率时, 控制器与交换机间最坏传输时延指交换机与控制器间分组传输时延 $\min d(v, s)$ 的最大值 T_{worst} , 如式(2)所示^[13-15]。文献[13]首次提出它作为解决 CPP 的性能尺度, 确保控制器与交换机间传输时延不会超过阈值, 确保所有控制器与交换机间传输时延都满足需求。因此, 通常选用控制器与交换机间最坏传输时延作为求解 CPP 的性能尺度。但 SDN 需部署较多的控制器, 这导致建设成本增加。

$$T_{\text{worst}} = \max_{v \in V} \min_{s \in S} d(v, s) \quad (2)$$

3) 网络故障率非零时最坏传输时延

网络设备的故障可能导致交换机与控制器失联。SDN 保护机制可确保失去连接的交换机与其他在线控制器连接。交换机与控制器间传输时延可能超过 T_{worst} 。Hock 等^[17]提出将网络故障率非零时最坏传输时延作为求解 CPP 的性能尺度。

在网络故障率非零时, 控制器与交换机间最坏传输时延指在当 SDN 中 $K-1$ 台控制器出现故障且一台控制器 ϕ 在线时, 控制器与交换机间最坏传输时延 T_{worst}^C , 如式(3)所示。它表示极端情况下的控制器与交换机间最坏时延, 求解的控制器位置通常位于网络中心。

$$T_{\text{worst}}^C = \max_{v \in V} \max_{\phi \in S} \min_{s \in S} d(v, s) \quad (3)$$

2.1.2 控制器间传输时延

分布式控制器架构是指控制逻辑分布在若干控制器。控制器间必须进行同步操作, 确保控制器间全局网络状态是一致的, 因此, 控制器间传输时延是求解 CPP 的重要性能尺度之一。目前, 通常选择控制

器间最坏传输时延作为 CPP 求解的性能尺度^[17~20]。控制器间最坏传输时延是指控制器间传输时延的最大值 T_{worst}^S , 如式(4)所示。控制器间最坏传输时延可以确保控制器间网络状态同步在规定时间内完成。

$$T_{\text{worst}}^S = \max_{s_1, s_2 \in S} d(s_1, s_2) \quad (4)$$

2.2 负载

控制器负载除与自身实现的软件架构相关外, 还受限于服务器本身硬件资源的制约。控制器负载过大导致网络事件的响应时延增加和控制器故障率增加等问题^[21,22]。因此, 控制器负载是求解 CPP 的重要性能尺度之一。

控制器负载包括网络零故障率时额定负载、网络零故障率时预期负载最大差值、网络零故障率时实时负载最大差值以及网络故障率非零时预期负载最大差值。

1) 网络零故障率时额定负载

在网络零故障率时, 控制器的额定负载指在单个控制器 s 的管理域 $S_\phi(s)$ 内交换机策略配置需求 $l(v)$ 的总和不超过控制器的额定负载能力 $L(s)$, 如式(5)所示^[14]。当选取额定负载作为 CPP 求解的性能尺度时, 有效避免了控制器额定负载的局限性导致分组丢失、网络事件无法实时响应等。但实际上, $l(v)$ 可能会超过控制器的额定负载, 应用场景具有局限性。

$$\sum_{v \in S_\phi(s)} l(v) \leq L(s) \quad (5)$$

2) 网络零故障率时预期负载最大差值

通过评估控制器实际负载^[17]来更好求解 CPP, 如城市人口、接入用户数量等。Hock 等^[17~19]提出了网络零故障率时控制器的预期负载差值作为 CPP 求解的性能尺度。在网络零故障率时, 控制器的预期负载最大差值指控制器最大负载 $\max l(s)$ 与控制器最小负载 $\min l(s)$ 的差值, 用 Ω' 表示, 如式(6)所示。网络零故障率时预期负载最大差值通过使 Ω' 最小化来确保控制器负载均衡。

$$\Omega' = \max_{s \in S} l(s) - \min_{s' \in S} l(s') \quad (6)$$

3) 网络零故障率时实时负载最大差值

控制器根据 SDN 中控制器的实际负载来重新计算控制器与交换机的映射关系。因此, 控制器的实时负载最大差值也可作为 CPP 求解的性能尺度。在网络零故障率时, 控制器的实际负载最大差值指控制器实际的最大负载 $\max l'(s)$ 与控制器实

际的最小负载 $\min l'(s)$ 的差值, 用 Ω'' 表示, 如式(7)所示^[16,23]。网络零故障率时实时负载最大差值通过使 Ω'' 最小化来调整控制器与交换机映射关系, 确保控制器负载均衡。

$$\Omega'' = \max_{s \in S} l'(s) - \min_{s' \in S} l'(s') \quad (7)$$

4) 网络故障率非零时预期负载最大差值

在网络故障率非零时预期负载最大差值将网络故障及 SDN 保护机制考虑在内, 指在任意 $K-2$ 个控制器发生故障且 2 个控制器在线时, 2 个控制器间负载的最大差值, 用 Ω^C 表示, 如式(8)所示^[25]。当 SDN 对系统健壮性要求特别严格时, 网络故障率非零时预期负载最大差值也是一种选择。

$$\Omega^C = \max_{s \in S} (\max_{s' \in S} l(s) - \min_{s' \in S} l(s')) \quad (8)$$

2.3 可靠性

SDN 故障通常导致控制器与交换机间中断、控制器与控制器间中断。管理与控制 SDN 的命令是通过控制路径传输的, 控制路径的可靠性直接关系到 SDN 的可靠性^[17~19,24~26]。因此, 可靠性是求解 CPP 的重要性能尺度之一, 侧重于控制路径的可靠性。

可靠性包括无保护机制时控制路径平均故障率、无保护机制时控制路径最高故障率以及有保护机制时控制路径平均故障率。

1) 无保护机制时控制路径平均故障率

在 SDN 无保护机制下, 控制路径平均故障率指控制器与交换机间失去连接概率的均值^[27~30]。如文献[27]中利用控制路径失联的预期率 (expected percentage of control path loss) δ 表示控制路径平均故障率, 通过使 δ 值最小化实现对 CPP 的求解。 δ 值由控制路径故障率 d_i 和网络设备 (包括控制器、交换机和物理链路等) 的故障率 p_i 这 2 个参数决定, 如式(9)所示。 m 表示控制路径数量, 与控制器架构相关, 如控制器采用层次式架构时^[31], m 值由控制器与交换机连接的 m 条控制路径以及控制器间连接的 $\frac{K(K-1)}{2}$ 条控制路径组成。

$$\delta = \frac{1}{m} \sum_{i \in (V, E)} d_i p_i \quad (9)$$

另外, 文献[28]提出用控制路径有效性的预期率 (expected percentage of valid control paths) δ_p 来反映控制路径平均故障率, 通过使 δ_p 最大化来确保控制路径具备良好的健壮性。 δ_p 与 δ 满足条件为

$\delta_p = 1 - \delta$ 。控制路径平均故障率仅表示由控制路径组成的控制网络的整体故障率，无法反映网络最坏情况。

2) 无保护机制时控制路径最高故障率

无保护机制时控制路径最高故障率指控制器与控制器间的控制路径以及控制器与交换机间的控制路径的故障率的最大值，用 δ' 表示。通过使最高故障率 δ' 最小化来提升 SDN 的健壮性。控制路径最高故障率与控制路径上节点故障率 p_{vj} 和链路故障率 p_{ei} 相关，如式(10)所示。文献[32]在假定网络链路或节点故障率相同时，使尽可能多的物理链路位于管理域内。同样，文献[17]提出 controller-less 的概念，确保网络故障不影响控制器对单个管理域内交换机的控制。由于分布式控制器架构中控制器间交互网络状态信息，现有研究未考虑控制器间路径可靠性。

$$\delta' = \max(1 - \prod(1 - p_{ei})(1 - p_{vj})) \quad (10)$$

3) 有保护机制时控制路径平均故障率

SDN 保护机制用于确保在网络故障时控制器与交换机依旧正常通信。在选择可靠性作为 CPP 求解的性能尺度时，应考虑到 SDN 保护机制对可靠性的影响。有保护机制时控制路径平均故障率指当运行 SDN 保护机制时，控制器与交换机间失联概率的均值。文献[33]利用控制器与交换机间路径的数量 δ'' 来反映网络平均故障率，如式(11)所示。 δ'' 值越大，网络平均故障率则越低。但当备份路径数量不一致时，无法反映控制路径的平均故障率。

$$\delta'' = \frac{\sum_{v \in V'} \sum_{s \in S} d(v, s)}{N} \quad (11)$$

同时，文献[34]在控制路径失联的预期率的基础上，提出了在考虑 SDN 保护机制时控制路径平均故障率 δ''' ，如式(12)所示。 S 表示所有网络情景集合（失败情景和一个无故障情景）， m 表示控制路径数量；情景 s 发生的概率为 p_s ； θ_s 表示所有控制路径的端点是故障点的总数； ϕ_s 表示经过故障点的控制路径总数； q_s 表示控制路径的备份路径的数量。

$$\delta''' = \sum_{s \in S} p_s \frac{(\theta_s + \phi_s q_s)}{m} \quad (12)$$

2.4 开销

为了更好地衡量控制器位置对 SDN 的影响，开销作为 CPP 求解的性能尺度也不失为一种好选

择，它包括 SDN 成本开销和流策略部署开销等。

2.4.1 成本开销

控制器的部署位置影响 SDN 建设的成本。控制器数量越少，其成本相对越低。文献[35]首次提出利用成本开销求解 CPP，使网络建设成本最小化。成本开销由控制器开销 $C_c(x)$ 、控制器与交换机间链路连接开销 $C_l(x)$ 以及控制器间连接开销 $C_i(x)$ 组成，如式(13)所示。当采取带内连接方式时，成本开销仅包括控制器开销，仅能求解控制器数量的最小值，却无法求解 CPP 最优方案。

$$sum' = \min(C_c(x) + C_l(x) + C_i(x)) \quad (13)$$

2.4.2 流策略部署开销

为了确保 SDN 性能最优，控制器应在最短时间内响应网络事件，并完成流策略的部署。因此，流策略部署也可是 CPP 求解的性能尺度之一^[36,37]。流策略部署开销由交换机向控制器请求的消息数量 $w(n)$ 和交换机到控制器间传递开销 $f(n, s)$ 组成，如式(14)所示。 $w(n)$ 可用网络节点所在城市人口来衡量； $f(n, s)$ 可用控制器与交换机传输时延或控制器负载表示。

$$sum'' = \sum_{n \in N} (w(n) f(n, s)) \quad (14)$$

3 搜索算法

在求解控制器放置问题的过程中，搜索算法根据性能尺度来确定最优情况下控制器数量以及控制器位置。但控制器放置问题的求解是 NP 难问题^[13,27,34]。另外，控制器选取的搜索算法与部署机制相关，如当控制器支持动态调整部署机制时，研究重点是在有限的时间内搜索出精确结果。若干求解 CPP 的搜索算法被提出，如随机搜索算法、完全搜索算法、贪婪搜索算法、博弈算法、整型规划算法和启发式搜索算法等。

3.1 随机搜索算法

在随机搜索算法^[14,27,29]中，通过随机性假设控制器数量以及控制器位置 2 个参数，结合选取的性能尺度对比评估值达到 CPP 求解目的。

随机搜索算法的时间复杂度为假设的次数，即 Num 。同时，求解 CPP 的精确性与假设次数相关。假设的次数越多，搜索结果的精确性也相应增加。只有满足 $Num \rightarrow \infty$ 时，随机搜索算法才能获取到最精确的搜索结果。因此，随机搜索算法难以求解

出 CPP 最优解。

3.2 完全搜索算法

完全搜索算法(或穷举算法)^[13,27]依据选取的性能尺度对控制器数量及控制器与节点间映射关系的所有组合进行遍历搜索,从中选取出控制器放置问题的最优解。在控制器数量从 1~ K 逐次增加的过程中,计算出控制器在 SDN 网络中位置所有组合的评估值。完全搜索算法求解的 CPP 的结果是最精确的,用于衡量其他搜索算法的精确性。

完全搜索算法的时间复杂度为 $\sum_{k=1}^n n^k \approx O(n^n)$ 。

尽管完全搜索算法时间复杂度相对较大,但计算集可拆分为互不相交的若干计算子集,通过分布式计算将计算子集由不同计算节点求解,可以大幅度地降低完全搜索算法的求解时间。通常,完全搜索算法适用于离线计算的场景。

3.3 贪婪搜索算法

贪婪搜索算法^[13,24,28,29,34,38,39]从 CPP 的一个初始解出发,根据选择性能尺度的约束条件逐步逼近给定目标,从而求出更优解。

但贪婪搜索算法在求解 CPP 时也存在一些不足。首先,贪婪搜索算法不能求出 CPP 中控制器最少数量,仅在已知控制器数量时求解控制器与交换机间映射关系;其次,只能在某些约束条件下求解出 CPP 的可行解范围,无法保证其所求结果是最优解。例如,Zhang 等^[32]首次将贪婪搜索算法引入到 CPP 求解;Hu 等^[27,28,34]提出基于贪婪搜索的 l-w-greedy 算法。

3.4 博弈算法

为适应在线求解的需求,非零和博弈算法被引入到 CPP 的求解^[23]。在非零和博弈算法中,每个控制器扮演选手角色,并选择时延和负载 2 个性能尺度作为代价函数 f_x 的参数。控制器间利用代价函数 f_x 进行博弈竞争:当控制器的处理能力无法满足网络事件处理请求时,使下线的控制器“X”重新上线,并重新计算控制器与交换机间映射关系;当控制器实际平均负载低于某个预设值时,选择使控制器“X”下线,并重新计算控制器的管理域。通过讨论个体的最优问题来完成对 CPP 的求解。非零和博弈将控制器放置问题的求解分散到 K 个控制器处理,在有限时间内求解出结果,但无法求得全局的最优解。

同时,讨价还价博弈也被引入到 CPP 的求解,通过“买方”时延、控制器间传输负载以及控制器负载三方不断博弈求解 CPP^[40]。

3.5 整数规划算法

整数规划模型^[14]在解决 CPP 时,通过增加新的约束参数来切割原问题解空间,不断缩小解空间,将 CPP 的整数最优解问题转化为解空间极点的问题^[41]。文献[14]选取特定半径 r 作为约束参数来切割原问题解空间,首先利用二分搜索求出半径 r ,再利用 r 约束参数调用整数规划求解方法。算法的时间复杂度与 SDN 拓扑结构相关,这是由于 SDN 拓扑结构直接影响到约束参数 r 值。

另外,文献[42]将 CPP 收敛为背包问题(knapsack problem),将交换机当作放入背包的对象,代价函数则是控制器与交换机间路径开销的反比。每次循环都利用 greedy-knapsack 算法依据代价函数选择出一个控制器位置,并确定控制器与交换机间映射关系,直到所有交换机都与控制器的映射确认才结束。

3.6 启发式搜索算法

启发式搜索算法被引入到 CPP 的求解中^[16,20,27,42,43],便于通过在线方式求解控制器与交换机的映射关系。启发式搜索算法求解包括模拟退火算法(simulated annealing algorithm)和基于 k -medioids clustering 的启发式算法这 2 个方面。

模拟退火算法是基于蒙特卡迭代求解法的一种启发式随机搜索算法,被引入到求解 CPP 中^[20,27,42]。利用 CPP 的最优解与最差解来确定初始温度,基于指数下降的温度计算函数,结合概率突变特性在解空间中随机寻找 CPP 的全局最优解。将模拟退火算法引入到 CPP 的求解时,并不一定可以找到全局的最优解。当模拟退火算法的参数设置恰当时,可在较短时间内求解出 CPP 的近似最优解。

基于 k -medioids clustering 的启发式算法^[16],利用平均时延和控制器负载均衡 2 个特殊性能尺度来实现对 CPP 的求解。在解空间中随机地利用 k -medioids clustering 算法多次搜索有效解来保证结果的精确性。同时,通过控制器额定负载限制其管理域内交换机数量来优化搜索性能。该算法能在较短的时间内和可接受的求解误差内求解出 CPP,但无法为 CPP 求解提供通用的解决方法。

3.7 基于最小割算法

最小割算法^[32]通过选取 SDN 拓扑图中所有割,

根据 SDN 网络应用场景选择合适的性能尺度来计算出最小割。最小割算法将拓扑图划分为 K 个子图，且每个子图内至少包含 2 个节点，则子图的数量即为 CPP 解的最小控制器数量，每个子图则为控制器的管理域。同时，基于最小割算法的时间复杂度与求解选择算法有关，通常采用 stoer-wagner 算法，但它不是一种求解 CPP 的通用算法。

3.8 多目标规划求解算法

SDN 应用场景要求同时选择多个性能尺度来求解 CPP。CPP 的求解被转化为多目标规划问题^[15,17~19,43]。CPP 的最优解取决于多个性能尺度的满足程度，在多个性能尺度之间和各种约束条件之间求得一个合理的折中值，如用 Pareto 最优解作为 CPP 的解^[17~19]。但 Pareto 最优解的解集大小与选取求解算法相关：传统优化技术一般得到的 Pareto 最优解的解集中仅包含一个有效解，而遗传算法求得 Pareto 最优解可能有多个。同时，利用多目标决策求解 CPP 还包括：1) 利用 Reference Level 算法求解 CPP^[15]；2) 将平均时延/最大时延最小化模型归纳为多目标规划求解，并利用模糊集理论求解^[43]；3) 利用改进的粒子群优化算法，通过控制器之间的协作和信息传递来达到对 CPP 的求解^[44,45]。因此，将 CPP 转化为多目标规划问题来求解 CPP 的时间复杂度和精确性取决于所选取的算法。

3.9 聚类问题求解算法

聚类问题求解算法根据选择的性能尺度和约束条件，将 N 个交换机（或网络节点）划分为 K 个类别，使每个类别内的交换机与控制器间关系的相似度较大；交换机与其他类别的控制器间关系的相似度较小^[15,32,46~48]。 K 为控制器的数量，

每个类别表示控制器的管理域。文献[32]首次将 CPP 收敛为聚类问题求解，利用代价函数 g 表示管理域间连接链路和代价函数 f 表示管理域内控制器与交换机间平均传输时延，通过使 f 和 g 最小化来对网络中管理域进行划分。文献[46]将 CPP 收敛为 K -critical 问题，求解出控制器的最小数量，并利用选择出来的控制器构建高健壮性和负载均衡的控制网络。另外，spectral clustering placement^[47,49]和 K -means^[50]也被引入到 CPP 求解中。

CPP 求解算法的选择涉及控制器部署位置最优解的精确性和求解的时间复杂度。根据 SDN 网络应用场景来选择不同的求解算法。从求解结果、求解时间、离线/在线求解和网络规模、通用性等方面对 CPP 求解算法进行归纳，如表 1 所示。

4 归纳与展望

从网络类型和计算方式这 2 个方面归纳 CPP 研究现状，并展望未来 CPP 发展趋势。

4.1 总结归纳

4.1.1 网络类型

SDN 从以企业网、校园网等局域网部署实践，到下一代互联网 Internet2 中应用与部署，再到对移动网络的研究与探索。这些都说明了 SDN 不只局限于一个或几个应用场景。目前，SDN 应用的网络类型大致分为数据中心网络、WAN 网络以及移动网络。3 类网络的特性也决定 CPP 求解需考虑的性能尺度和搜索算法具有的一定的差异性。

1) 数据中心网络

CPP 在数据中心网络中研究相对较少，这与数据中心网络的特性有关^[32,35]。数据中心位于空间相对有限的机房内，控制器与交换机间距离相对要

表 1 求解控制器放置问题中常用算法的特性

算法	求解结果	求解时间	离线/在线求解	网络规模	通用性
随机搜索算法	最差解	受限于计算次数	离线	无限制	非通用
完全搜索算法	最优解	时间最长	离线	中小型	通用
贪婪搜索算法	较优解	受限于算法实现	离线/在线	无限制	通用
博弈算法	较优解	相对较短	离线/在线	无限制	通用
整数规划算法	近似最优解	相对较短	离线	无限制	通用
启发式搜索算法	较优解	受限于算法实现	离线/在线	无限制	通用
基于最小割算法	近似最优解	相对较长	离线	无限制	通用
多目标规划求解算法	近似最优解	受限于算法实现	离线	中小型	通用
聚类问题求解算法	较优解	受限于算法实现	离线	无限制	通用

短。为了避免数据中心配置管理复杂性,控制器与交换机间通常采取带外方式,即单独建立一套控制网络用于交换机与控制器间以及控制器与控制器间的数据传输。数据中心的网络拓扑结构相对固定,通过 SDN 实践经验能指导控制器的部署位置。

2) WAN 网络

多数的 CPP 通常围绕着 WAN 网络展开研究,这与 WAN 网络的属性相关:① WAN 网络的拓扑结构多样性,单纯通过 SDN 实践经验无法给出科学的部署方案;② WAN 网络的环境复杂性和网络业务类型多样性影响到选取的性能尺度和搜索算法;③ 带内连接方式实现控制器间、控制器与交换机间数据传输,增加 CPP 的复杂性。随着 SDN 在 WAN 网络中的应用,研究 CPP 更具有现实意义。

3) 移动网络

随着 SDN 在移动网络中的研究与探索,对控制器在移动网络的部署位置展开了研究^[51~53]。控制器与基站间采用带内连接方式,才能保证控制器与距离较远的基站连接。但控制器在移动网络中主要侧重移动终端的动态接入控制。因此,控制器在移动网络中的部署位置通常与网络运营商的部署策略相关。

4.1.2 计算方式

为了确保 SDN 具有良好的服务质量,针对控制器的框架及实现机制展开深入研究,避免控制器成为 SDN 性能瓶颈。一方面, CPP 求解为网络初期建设提供合理的规划,通过科学规划控制器位置来确保 SDN 能够提供更优的服务质量。另一方面,网络故障导致分组丢失、控制器与交换机间失联等现象。为了保证 SDN 的健壮性,通常采取复杂保护机制,根据网络状况实时地调整控制器与交换机间映射关系。因此, CPP 求解的计算方式划分为离线计算和在线计算。

1) 离线计算

离线计算指在已知网络拓扑结构情况下,根据 SDN 网络部署场景的需求选择合适的性能尺度和搜索算法,求解出控制器最优的部署方案,即控制器数量、控制器与交换机间映射关系。在离线计算中,控制器不因 SDN 环境变化(如网络拓扑结构改变、网络设备的单点故障以及控制器的负载过大等)来重新计算控制器与交换机间映射关系。离线计算仅单次求解 CPP,根据求解的结果部署控制器。

为了避免网络故障出现分组丢失、交换机与控

制器间失联等现象,一系列保护机制被提出。如在控制器与交换机间传输路径建立备份机制、控制器冗余机制,当控制器出现故障,交换机将尝试连接冗余控制器。但并不需要对 CPP 进行重新计算,属于离线计算范畴。

离线计算通常对 CPP 求解的时间要求不高,允许花费几天甚至几周的时间^[13],同时,离线计算通常对 CPP 求解的精确性相对较高,期望求解出控制器最优位置方案。因此,离线计算用于在网络规划阶段求解 CPP。

2) 在线计算

控制器的负载影响到控制器响应网络事件时间:控制器负载过大导致网络事件响应时间相对较长;控制器负载过小致使控制器的资源利用率较低。同时,网络故障也是影响网络性能的因素之一,会导致分组丢失、交换机与控制器间失联等现象。为解决上述问题,现有 SDN 保护机制支持动态地调整控制器与交换机间映射关系,根据网络中控制器负载或网络故障发生后的拓扑结构,及时调整控制器的管理域^[54~57]。

当 SDN 动态调整控制器与交换机间映射关系的机制时,离线计算时间长的特征无法满足在线求解 CPP 的需求。用于根据网络状况来实时求解 CPP 的在线计算方式被提出。一旦满足触发条件,在线计算将重新求解 CPP,利用求解的结果来实现动态地调整网络中控制器与交换机间映射关系,确保网络性能的最大化。

在线计算除了要求控制器具有动态调整控制器管理域的机制外,还要求 CPP 求解算法能够在较短时间内计算出控制器最优的调整方案,防止求解 CPP 时间过长导致无法实时调整控制器与交换机的映射关系^[58]。在线计算对 CPP 求解的精确性相对不高,在一定误差范围内接收近似最优解。

4.2 未来展望

随着 SDN 在 WAN 网络中逐步的部署与应用, CPP 已成为现实而有意义的研究热点^[59]。通过分析并总结求解 CPP 的相关研究,对 CPP 研究未来趋势进行展望。

1) 由于 WAN 网络具有跨越地域广、拥有节点数量多的特点,多控制器架构已成为 SDN 网络未来发展趋势^[60,61]。控制器间通过交互信息来确保网络状态一致性,但 CPP 的前期研究没有考虑到控制器间协同的因素。因此, CPP 研究应紧密结合控制

器采取的架构,并非提出通用的求解方法。如控制器分别选取分布式架构和层次式架构时,由于控制器间交互信息的控制路径的重要性不同(分布式架构中 2 控制器间控制路径重要性相同;层次式架构中 master 控制器与 slaver 控制器间控制路径重要性要高),设计求解控制器放置问题的思路也不尽相同。现有的 CPP 研究侧重于通用的解决思路,在 SDN 规划时很难有较好的指导意义。未来 CPP 研究应结合控制器架构,根据机制原理来选取性能尺度。

2) SDN 技术应用场景呈现多样化发展趋势,网络业务需求自然存在差异性^[62]。如企业网侧重于访问控制,应尽可能确保控制器与交换机间控制路径的可靠性,优先选择可靠性作为求解 CPP 的性能尺度之一;WAN 网络要求控制器响应网络事件的时间短,应选择传输时延作为求解 CPP 的性能尺度之一。目前,现有研究通常不指明应用场景、性能尺度的选择缺乏合理依据、实验验证数据同质化。未来研究应从实用性角度出发,结合 SDN 应用场景的需求来研究 CPP,以便指导同类或相似的网络场景中 CPP 的求解。

3) SDN 是典型的集中控制架构,系统健壮性是其与生俱来的局限性。面对上述挑战,一系列的保护机制被提出并应用到 SDN 中。当交换机与控制器失联时,交换机主动搜寻其他在线控制器,确保网络正常运转。CPP 求解应指导交换机搜寻到最优的控制器与交换机映射关系。在网络故障时交换机不具备求解 CPP 的能力,通常利用备份机制使交换机与其他预置的控制器尝试连接。目前,在线求解方式适用于根据控制器负载调整控制器与交换机映射关系的场景。未来的研究要结合 SDN 保护机制的内在机理来研究 CPP 的在线求解方式。

4) 完全搜索算法依旧是网络前期规划求解 CPP 的优先选择,也是其他求解 CPP 搜索算法评估的重要依据。尽管完全搜索算法的时间复杂度相对较大,但将完全搜索算法求解的计算集划分为若干不相交的子集,通过分布式时间来提升完全搜索算法的求解性能。未来对搜索算法的研究主要侧重于在线求解算法,研究热点是如何在较短的时间内求解出近似最优的控制器放置方案。

5 结束语

针对 SDN 研究领域控制器放置问题,本文全面阐述了 CPP 在性能尺度和搜索算法 2 个方向的

研究现状。在此基础上,从网络类型和计算方式 2 个维度归纳了 CPP 的研究内容,并展望了 CPP 未来的发展趋势,以期为后续的进一步研究提供参考。

参考文献:

- [1] YANG L, DANTU R, ANDERSON T, et al. Forwarding and control element separation (ForCES) framework[S]. RFC 3746, 2004.
- [2] KIM B G, WANG P. ATM network: goals and challenges[J]. Communications of the ACM, 1995, 38(2):39-44.
- [3] VAN D M J E, ROONEY S, LESLIE I, et al. The tempest-a practical framework for network programmability[J]. IEEE Network, 1998, 12(3): 20-28.
- [4] BISWAS J, LAZAR A A, HUARD J F, et al. The IEEE P1520 standards initiative for programmable network interfaces[J]. IEEE Communications Magazine, 1998, 36(10): 64-70.
- [5] SMITH J M, CALVERT K L, MURPHY S L, et al. Activating networks: a progress report[J]. Computer, 1999, 32(4): 32-41.
- [6] 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.
- [7] KOPONEN T. Software is the future of networking[C]//The 8th ACM/IEEE Symposium on Architectures for Networking and Communications Systems. 2012:135-136.
- [8] GUDE N, KOPONEN T, PETTIT J, et al. NOX: towards an operating system for networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(3): 105-110.
- [9] KOPONEN T, CASADO M, GUDE N, et al. Onix: a distributed control platform for large-scale production networks[C]//The 9th USENIX Conference on Operating Systems Design and Implementation (OSDI'10). 2010: 1-6.
- [10] BHOWMIK S, TARIQ M A, KOLDEHOFE B, et al. Distributed control plane for software-defined networks: a case study using event-based middleware[C]//2015 ACM International Conference on Distributed Event-Based Systems. 2015:92-103.
- [11] LIN P P, BI J, HU H Y. ASIC: an architecture for scalable intra-domain control in OpenFlow[C]//The 7th International Conference on Future Internet Technologies (CFI'12). 2012: 21-26.
- [12] YEGANEHS H, GANJALI Y. Kandoo: a framework for efficient and scalable offloading of control applications[C]//The 1st Workshop on Hot Topics in Software Defined Networks (HotSDN'12). 2012: 19-24.
- [13] HELLER B, SHERWOOD R, MCKEOWN N. The controller placement problem[C]//The First Workshop on Hot Topics in Software Defined Networks. 2012: 7-12.
- [14] 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.
- [15] VOCHIN M, BORCOCI E, AMBARUS T. On multi-controller placement optimization in software defined networking based WANs[C]//The 4th International Conference on Networks. 2015:261-266.
- [16] LANGE S, GEBERT S, SPOERHASE J, et al. Specialized heuristics for the controller placement problem in large scale SDN networks[C]//International Teletraffic Congress. 2015:210-218.

- [17] HOCK D, HARTMANN M, GEBERT S, et al. Pareto-optimal resilient controller placement in SDN-based core networks[C]//The 25th International Teletraffic Congress (ITC). 2013: 1-9.
- [18] HOCK D, GEBERT S, HARTMANN M, et al. POCO-framework for Pareto-optimal resilient controller placement in SDN-based core networks[C]//IEEE Network Operations and Management Symposium (NOMS). 2014: 1-2.
- [19] HOCK D, HARTMANN M, GEBERT S, et al. POCO-PLC: enabling dynamic Pareto-optimal resilient controller placement in SDN networks[C]//2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). 2014: 115-116.
- [20] 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.
- [21] TOOTOONCHIAN A, GORBUNOV S, GANJALI Y, et al. On controller performance in software-defined networks[C]//2012 Usenix Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services. 2012:7-10.
- [22] YAO G, BI J, GUO L. On the cascading failures of multi-controllers in software defined networks[C]//The IEEE International Conference on Network Protocols. 2013: 1-2.
- [23] 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]//The 15th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM). 2014: 1-6.
- [24] BEHESHTI N, ZHANG Y. Fast failover for control traffic in software-defined networks[C]//2012 IEEE Global Communications Conference (GLOBECOM). 2012: 2665-2670.
- [25] ZHANG T, BIANCO A, GIACCONE P. The role of inter-controller traffic in SDN controllers placement[C]// 2016 IEEE Network Function Virtualization and Software Defined Networks. 2016: 1-6.
- [26] LIU J, XIE R. Reliability-based controller placement algorithm in software defined networking[J]. Computer Science and Information Systems, 2016, 13(2):547-560.
- [27] HU Y, WANG W, GONG X, et al. Reliability-aware controller placement for software-defined networks[C]//IFIP/IEEE International Symposium on Integrated Network Management. 2013:672-675.
- [28] HU Y, WANG W, GONG X, et al. On the placement of controllers in software-defined networks[J]. The Journal of China Universities of Posts and Telecommunications, 2012, 19(19): 92-97.
- [29] XU J T, QU H, ZHAO J H. KMOBPSO-based high reliability controller placement algorithm in SDN[J]. Telecommunications Science, 2015, 31(6):71-77.
- [30] GUO M, BHATTACHARYA P. Controller placement for improving resilience of software-defined networks[C]//The 4th International Conference on Networking and Distributed Computing (ICNDC). 2013:23-27.
- [31] FU Y, BI J, CHEN Z, et al. A hybrid hierarchical control plane for flow-based large-scale software-defined networks[J]. IEEE Transactions on Network & Service Management, 2015, 12(2):117-131.
- [32] ZHANG Y, BEHESHTI N, TATIPAMULA M. On resilience of split-architecture networks[C]//IEEE Global Telecommunications Conference (GLOBECOM 2011). 2011: 1-6.
- [33] MULER L F, OLIVERIA R R, LUIZELLI M C, et al. Survivor: an enhanced controller placement strategy for improving SDN survivability[C]//2014 IEEE Global Communications Conference (GLOBECOM). 2014: 1909-1915.
- [34] 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.
- [35] SALLAHI A, STHILAIRE M. Optimal model for the controller placement problem in software defined networks[J]. IEEE Communications Letters, 2015, 19(1): 30-33.
- [36] ZENG D, TENG C, GU L, et al. Flow setup time aware minimum cost switch-controller association in software-defined networks[C]//The 11th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE). 2015: 259-264.
- [37] YAO L, HONG P, ZHANG W, et al. Controller placement and flow based dynamic management problem towards SDN[C]//2015 IEEE International Conference on Communication Workshop (ICCW). 2015:363-368.
- [38] SAHOO K S, SAHOO B, DASH R, et al. Optimal controller selection in software defined network using a greedy-SA algorithm[C]//2016 IEEE Conference IndiaCom. 2016:1-5.
- [39] OBADIA M, BOUET M, ROUGIER J L, et al. A greedy approach for minimizing SDN control overhead[C]//The 1st IEEE Conference on Network Softwarization (NetSoft). 2015: 1-5.
- [40] KSENTINI A, BAGGA M, TALEB T, et al. On using bargaining game for optimal placement of SDN controllers[C]//ICC 2016-2016 IEEE International Conference on Communications. 2016:1-6.
- [41] QI H, LI K. Software-defined networking controller placement in distributed datacenters[M]//Software Defined Networking Applications in Distributed Datacenters. Springer International Publishing, 2016:39-55.
- [42] BARI M F, ROY A R, CHOWDHURY S R, et al. Dynamic controller provisioning in software defined networks[C]//The 9th International Conference on Network and Service Management (CNSM). 2013: 18-25.
- [43] ROS F J, RUIZ P M. On reliable controller placements in software-defined networks[J]. Computer Communications, 2016, 77: 41-51.
- [44] GAO C, WANG H, ZHU F, et al. A particle swarm optimization algorithm for controller placement problem in software defined network[C]//The 15th International Conference on Algorithms and Architectures for Parallel Processing. 2015: 44-54.
- [45] LIU S, WANG H, YI S, et al. NCPSO: a solution of the controller placement problem in software defined networks[C]//The 15th International Conference on Algorithms and Architectures for Parallel Processing. 2015:213-225.
- [46] CERVELLO-PASTOR C, GARCIA A J. On the controller placement for designing a distributed SDN control layer[C]//2014 IFIP Networking Conference. 2014: 1-9.
- [47] XIAO P, QU W, QI H, et al. The SDN controller placement problem for WAN[C]//2014 IEEE/CIC International Conference on Communications in China (ICCC). 2014:220-224.
- [48] AOKI H, SHINOMIYA N. Controller placement problem to enhance performance in multi-domain SDN networks[C]//The Fifteenth International Conference on Networks. 2016: 108-109.
- [49] XIAO P, ZHI-YANG L I, GUO S, et al. A K self-adaptive SDN controller placement for wide area networks[J]. Frontiers of Information Technology

- & Electronic Engineering, 2016, 17(7):620-633.
- [50] WANG G, ZHAO Y, HUANG J, et al. A *K*-means-based network partition algorithm for controller placement in software defined network[C]//IEEE International Conference on Communications. 2016:1-6.
- [51] AUROUX S, KARL H. Efficient flow processing-aware controller placement in future wireless networks[C]//Wireless Communications and Networking Conference (WCNC). 2015:1787-1792.
- [52] LIYANAGE M, GURTOV A, YLIANTTILA M. The controller placement problem in software defined mobile networks (SDMN)[M]// Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture. 2015:129-147.
- [53] AUROUX S, KARL H. Flow processing-aware controller placement in wireless DenseNets[C]//The 25th International Symposium on Personal, Indoor, and Mobile Radio Communication. 2014:1294-1299.
- [54] KILLI B P R, RAO S V. Optimal model for failure foresight capacitated controller placement in software-defined networks[J]. IEEE Communications Letters, 2016, 20(6): 1108-1111.
- [55] UI HUQUE M T I, JOURJON G, GRAMOLI V. Revisiting the controller placement problem[C]//Local Computer Networks. 2015:450-453.
- [56] YAO L Y, CHEN Y, SONG F, et al. Delay-aware controller placement for fast response in software-defined network[J]. Journal of Electronics & Information Technology, 2014,36(12):2802-2808.
- [57] JIMENEZ Y, CORDERO J A, CERVELLO-PASTOR C. Measuring robustness of SDN control layers[C]//2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). 2015:774-777.
- [58] HU Y, WANG W, GONG X, et al. Balanceflow: controller load balancing for openflow networks[C]//The 2nd International Conference on Cloud Computing and Intelligent Systems (CCIS). 2012: 780-785.
- [59] NGUYEN K, MINH Q T, YAMADA S. A software-defined networking approach for disaster-resilient WANs[C]//International Conference on Computer Communications and Networks. 2013:1-5.
- [60] DIXIT A, HAO F, MUKHERJEE S, et al. Towards an elastic distributed SDN controller[J]. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 7-12.
- [61] MUQADDAS A S, BIANCO A, GIACCONE P, et al. Inter-controller traffic in ONOS clusters for SDN networks[C]//IEEE International Conference on Communications. 2016: 1-6.

- [62] FENG T, BI J, WANG K. Joint allocation and scheduling of network resource for multiple control applications in SDN[C]//Network Operations and Management Symposium. 2014:1-7.

作者简介:



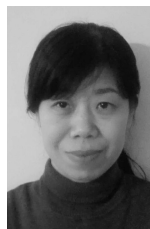
高先明 (1988-), 男, 山东济南人, 国防科学技术大学博士生, 主要研究方向为下一代网络体系架构、路由体系架构和网络功能虚拟化等。



王宝生 (1970-), 男, 河北沧州人, 国防科学技术大学教授, 主要研究方向为下一代网络体系架构、路由体系架构和路由协议和安全等。



邓文平 (1981-), 男, 湖南耒阳人, 国防科学技术大学讲师, 主要研究方向为下一代网络体系架构、BGP 和计算机网络等。



陶静 (1971-), 女, 山东昌邑人, 国防科学技术大学副教授, 主要研究方向为网络应用、网络安全和分布式仿真等。