

低故障恢复开销的软件定义网络控制器布局算法

吴奇, 陈鸿昶

(信息工程大学, 河南 郑州 450001)

摘要: 控制器部署技术是软件定义网络的重要研究方向。现有控制器布局技术更多地关注节点故障时网络的鲁棒性布局方法, 缺乏对连边故障情况下的灾备分析。为了解决这个问题, 建立了一种低故障恢复开销软件定义网络控制器布局模型, 该模型针对连边故障进行设计, 并利用基于模拟退火思想的算法进行求解。仿真结果表明, 该模型及算法在尽量少地损失网络时延代价的同时, 有效降低了控制路径出现故障时网络的故障恢复开销。

关键词: 软件定义网络; 控制器部署; 鲁棒性; 连边故障; 故障恢复开销

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2020066

Low failure recovery cost controller placement strategy in software defined networks

WU Qi, CHEN Hongchang

Information Engineering University, Zhengzhou 450001, China

Abstract: The controller placement is an important problem in software defined networks. Existing researches pay more attention to deploying controller planning for node failures and ignore the fact that the link failure can also effect the network drastically. To solve the problem, a mathematical model for the controller placement that plans ahead for the link failures to avoid a drastic increase failure recovery cost was proposed. To solve the model, an algorithm based on simulated annealing was designed. The simulation results show that the proposed model and algorithm can improve the robustness of the network. The maximum failure recovery cost is significantly reduced when a link falls in failures.

Key words: software defined network, controller placement, robustness, link failure, failure recovery cost

收稿日期: 2019-12-13; 修回日期: 2020-02-04

通信作者: 吴奇, wqstudy@outlook.com

基金项目: 国家重点研发计划基金 (2018YFB0804004)

Foundation Item: The National key Research and Development Program of China (2018YFB0804004)

论文引用格式: 吴奇, 陈鸿昶. 低故障恢复开销的软件定义网络控制器布局算法[J]. 网络与信息安全学报, 2020, 6(6): 97-104.

WU Q, CHEN H C. Low failure recovery cost controller placement strategy in software defined networks[J]. Chinese Journal of Network and Information Security, 2020, 6(6): 97-104.

1 引言

随着网络技术的快速发展,传统网络已经难以适应未来网络发展的需求,为改变现有网络困局,软件定义网络(SDN, software defined network)应运而生^[1-2]。软件定义网络是一种新型网络技术,其通过分离网络中控制层和数据层,实现了对网络的灵活管理,可以有效适应未来网络灵活、按需、弹性的设计需求。

在SDN的设计中,多控制器部署架构^[3-6]被设计出来以满足各项业务的需求。通常在多控制器部署架构中,网络会被分为多个域,每一台控制器仅控制自身域内的交换机,多个控制器互相协作、信息交互,并实现全局性的网络视图。在多控制器部署架构中,控制路径的鲁棒性研究一直是研究者的研究重点,目前控制路径的研究思路主要分为两个方面^[7],分别为无保护机制控制路径研究和有保护机制控制路径研究,在无保护机制控制路径研究中,研究者对控制路径的故障概率进行建模,通过最小化故障概率模型,进而实现控制路径的可靠性部署;而在有保护机制控制路径研究中,研究者为每一个交换机设置多条不相交的控制路径,进而保证单条控制路径出现故障时,交换机仍然可以和控制器维持正常的连接。

虽然上述研究一定程度上提高了控制路径的可靠性,但这些研究大多将整个控制路径看作整

体,缺少对路径上连边发生故障的分析,事实上,控制路径中不同连边故障对于网络的影响是不同的,连边故障导致的交换机迁移数量越多,网络的故障恢复成本(带宽消耗、迁移成本等)越大。图1所示网络就是一个典型的SDN,其中包含3个控制域,以控制域2中的交换机布局为例,在图1(a)中控制器被部署在交换机B上,网络中其他的交换机可以通过路径“A-B”“D-B”及“C-B”3条控制路径连接到控制器B,任何一条连边的故障最多影响一个控制器;但若控制器按照图1(b)所示部署在交换机D上时,网络中其他交换机的控制路径分别为“A-B-D”“C-B-D”及“B-D”3条路径连接到控制器D,显然地,在该种情况下连边“B-D”作为控制路径被经过的次数远远大于连边“A-B”和“C-B”,一旦连边“B-D”发生故障,交换机A、B、C都将无法连接到控制器D,此时网络在进行故障恢复时,需要对3个交换机进行故障迁移,此时网络付出的迁移成本会明显大于图1(a)的布局情况。

为了降低连边故障在网络故障过程中增加的恢复成本,本文设计了一种低恢复开销的控制平面布局算法,该方法首先由每一条连边作为控制路径的次数,构造出控制路径的故障恢复开销,然后通过平衡控制路径的故障恢复开销(failure recovery cost)和故障恶化时延(latency deterioration under link failures),构造出控制平面模型,为对模型进行求解,设计了一种基于模拟退火算

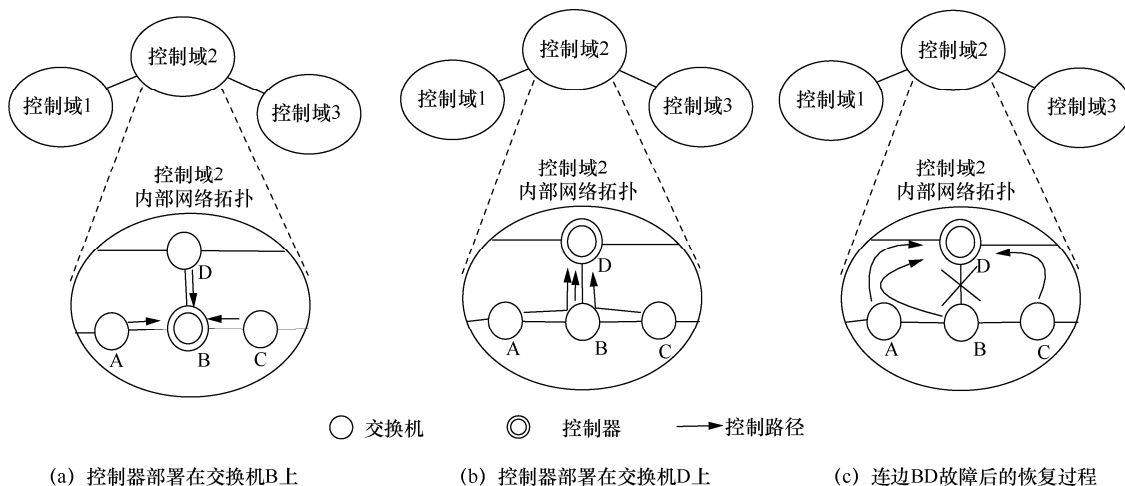


图1 不同控制器部署位置对交换机控制路径的影响

Figure 1 Different controller placements and their impact on switch control paths

法的优化算法。仿真实验表明,所提算法相对于其他算法,其部署出的控制平面布局可以在尽量少地损失网络的时延代价的同时,有效降低控制路径出现故障时网络的故障恢复开销。

2 相关研究

自从 SDN 被提出以来,控制器部署问题(CPP, controller placement problem)^[8]是一个核心问题。传统的控制器部署问题大多针对时延指标、负载均衡、故障容忍等多种指标进行设计,主要以提高 SDN 的可用性为目标^[9-12]。但近年来,随着 SDN 的大规模应用,控制器部署的可靠性问题引起越来越多的研究者关注。

控制器部署的可靠性研究是指控制器和交换机间中断时(可能是控制器故障、交换机故障或是控制路径故障)如何保证 SDN 正常工作能力的一些研究工作。目前,SDN 中针对控制路径可靠性的研究工作主要分为无保护机制的控制路径研究和有保护机制的控制路径研究^[7]。

无保护机制的控制路径研究通过分析控制路径的故障率来分析网络的可靠性。控制路径故障率是指控制区与交换机间失去连接概率的均值。文献[13]通过最小化控制路径失联概率(expected percentage of control path loss)的期望,实现了最小故障概率的控制器布局。文献[14]通过最大化控制路径有效性的期望(expected percentage of valid control paths),确保了网络中整体控制路径的鲁棒性。文献[15]通过假定网络连边和节点的故障率相同,设计了整合连边和节点故障情况的布局方法。

有保护机制的控制路径研究则通过预先为控制路径分配多条控制路径的策略,进而降低单条控制路径故障对网络的影响。文献[11]利用控制器与交换机间路径的数量来反映网络控制路径的可靠性,控制器与交换机间的路径数量越多,网络路径故障率越低。文献[16]在控制路径失联期望的基础上,提出了有保护机制下的控制路径故障率,进一步分析了控制路径的安全性。文献[17]利用网络中每一个交换机距离控制器间不相交控制路径的数量,用以评估控制路径的可靠性,可以保证单条路径上多条连边故障时,交换机仍有

路径可以连接控制器。文献[18]设计了一种多阶段路径重规划机制,可以在控制路径出现故障后使交换机快速重新定向到控制器。

3 模型建立

本文首先分析网络中每一条连边的故障恢复开销,然后分析连边故障时交换机的故障恶化时延,最后综合故障恢复开销和故障恶化时延,构造出控制平面部署模型。

3.1 主要参数

表 1 给出了模型中的主要符号。

表 1 模型中的主要符号
Table 1 Symbols in problem

符号	定义
V	网络中全部的交换机集合
E	网络中全部的连边集合
g_{ij}	连边经过度
h_k^{ij}	连边 $\langle i, j \rangle$ 是否为节点 k 控制路径上连边
c_{ij}	连边 $\langle i, j \rangle$ 导致的交换机故障恢复开销
d_{ij}	连边 $\langle i, j \rangle$ 之间的距离
p_k	交换机 k 的控制路径的故障恢复开销
q_k	交换机 k 的控制路径时延
U_x	控制器 x 的最大负载
u_k	交换机 k 生成的最大负载
w_{kx}	交换机 k 是否被控制器 x 控制
y_x	交换机 x 上是否部署有控制器
M	网络中部署的控制器数量

3.2 控制路径故障恢复开销

在 SDN 中,每一个交换机到控制器之间都存在着控制路径,控制路径由多个连边组成,显然对于一个连边而言,其作为控制路径的次数越多,该连边故障时,受到影响的交换机数量越多。本文利用连边经过度 g_{ij} 表示网络中全部交换机将连边 $\langle i, j \rangle$ 作为控制路径的次数。

$$g_{ij} = \sum_k h_k^{ij} \quad (1)$$

其中, h_k^{ij} 表示连边 $\langle i, j \rangle$ 是否为交换机 k 控制路径上连边,若是, h_k^{ij} 的取值则为 1, 否则为 0。

交换机的故障恢复过程主要和交换机的备份控制路径相关,一旦主控制路径上的连边出现故障,交换机将会尝试利用备份控制路径连接控制

器,图 1(c)展示了连边“B-D”故障后,交换机 A、B、C 的备份控制路径,虽然这些备份控制路径可以使交换机重新连接到控制器,但显然这些备份控制路径的长度远远大于图 1(b)中的主要控制路径长度。这些备份控制路径的长度是交换机在连边故障下的恢复开销,若备份控制路径长度为 b_k ,则连边 $\langle i, j \rangle$ 导致的故障恢复开销 c_{ij} 为

$$c_{ij} = \sum_k h_k^{ij} b_k \quad (2)$$

故障恢复开销 c_{ij} 的物理意义是受连边故障影响交换机的备份控制路径长度之和。显然,故障恢复开销 c_{ij} 的值越大,连边故障对于网络的影响越大。

由连边 $\langle i, j \rangle$ 导致的故障恢复开销 c_{ij} ,可以构造出每一个交换机控制路径的故障恢复开销 p_k 。

$$p_k = \sum_{ij} c_{ij} h_k^{ij} \quad (3)$$

此时,全部控制路径的故障恢复开销可以计算为

$$p^G = \sum_k p_k \quad (4)$$

3.3 连边故障恶化时延

若网络中连边 $\langle i, j \rangle$ 的时延为 d_{ij} ,则交换机 k 的控制路径时延可以表示为

$$q_k = \sum_{ij} d_{ij} h_k^{ij} \quad (5)$$

此时,全部控制路径的正常平均时延 q_{normal}^G 可以计算为

$$q_{\text{normal}}^G = \frac{1}{V} \sum_k q_k \quad (6)$$

若连边故障 $\langle x, y \rangle$ 出现故障,则交换机 k 的控制路径时延可以表示为

$$q_k^{xy} = \sum_{ij} d_{ij} h_k^{ij} (1 - h_k^{xy}) + b_k h_k^{xy} \quad (7)$$

q_k^{xy} 前一项表示交换机 k 的控制路径上不包含故障连边 $\langle x, y \rangle$ 时的时延,后一项表示交换机 k 的控制路径上包含故障连边 $\langle x, y \rangle$ 时的时延。由 q_k^{xy} 可以计算出连边故障 $\langle x, y \rangle$ 出现故障时,

网络全部交换机的平均时延 q_{xy}^G 。

$$q_{xy}^G = \frac{1}{V} \sum_k q_k^{xy} \quad (8)$$

任意一条连边出现故障时,网络的全部交换机的时延为

$$q_{\text{fault}}^G = \frac{1}{EV} \sum_{xy} \sum_k q_k^{xy} \quad (9)$$

故网络的连边故障恶化时延可以表示为

$$q^G = |q_{\text{normal}}^G - q_{\text{fault}}^G| \quad (10)$$

3.4 模型建立

本文的研究目的是如何在网络中部署控制器,即控制路径规划,在尽量少地损失网络时延代价的同时,有效降低控制路径出现故障时网络的故障恢复开销,故优化目标可以表示为

$$\min v_G \quad (11)$$

优化目标的约束为

$$v_G = \lambda p^G + (1 - \lambda) q^G \quad (12)$$

$$U_x y_x \geq \sum_k u_k w_{kx} \quad (13)$$

$$\sum_x w_{kx} = 1 \quad (14)$$

$$\sum_x y_x = M \quad (15)$$

$$w_{kx}, y_x, h_k^{ij} \in \{0, 1\} \quad (16)$$

其中, λ 为平衡因子, U_x 表示控制器 x 的最大负载, u_k 表示网络中每一个交换机产生的负载, w_{kx} 表示控制器 x 是否为交换机 k 的控制器, y_x 表示控制器是否部署在交换机 x 上,在优化目标的约束中,式(13)表示网络中每一个控制器控制的交换机负载需要小于控制器最大能接受的最大负载,式(14)表示网络中每一交换机都拥有一个控制器,式(15)表示网络中的控制器数量为 M ,式(16)表示参数取值范围。

4 问题求解

基于建立的控制平面部署模型,本文针对目标函数进行求解,设计了一种基于模拟退火的求解方法,模拟退火算法^[19]是一种启发式算法,其源于固体退火原理,综合了统计物理学知识,通过“降温”的方式寻求系统平衡状态,从而得到

近似最优解。它的渐近收敛性（理论上已被证明以概率 1 收敛于全局最优解）和简单通用性适合于本文所要求解的优化目标。

算法首先随机从网络的全部可能的控制器部署位置上选择一种布局方式，按照最短时延原则和负载约束计算出每一个交换机的控制器归属，然后根据交换机的控制器归属情况，计算出交换机到控制器的最短控制路径和控制路径上的连边经过度，并由式(12)计算出此时控制平面布局的合理性。

大部分情况下，初始控制平面部署并不是最优的控制平面部署情况，故需要采用更新策略在原始控制平面部署的基础上计算出新的控制器布局方式，更新策略可以表示为

$$\sum_x \sum_{x'} y_x y_{x'} = M - 1 \quad (17)$$

其中， y_x 表示第 n 次迭代时的控制器部署位置的集合， $y_{x'}$ 表示第 $n+1$ 次迭代时的控制器部署位置的集合。通过上述公式可以保证网络在第 n 次迭代和第 $n+1$ 次迭代时，仅有一个控制器的部署位置发生变化。

然后，考虑到算法可能陷入局部最优解，本文利用 Metropolis 准则接受新解。

$$P(\Delta, T) = \begin{cases} \exp\left(-\frac{\Delta}{T}\right), \Delta \geq 0 \\ 1, \Delta < 0 \end{cases} \quad (18)$$

其中， T 表示模拟退火网络中的温度参数，其更新机制为 $T_{n+1} = \alpha T_n$ ， n 表示网络中的迭代次数， $\Delta = (v_G)_{n+1} - (v_G)_n$ 表示两次迭代过程的目标值之差。式(18)表示若解 $(v_G)_{n+1} < (v_G)_n$ ，算法将以 1 的概率接受新解，否则将以 $P(\Delta, T)$ 的概率接受新解。

综上所述，基于模拟退火的优化算法可以表示如下。

算法 1 基于模拟退火的优化算法

输入 网络拓扑 G ，初始温度 T_0 ，终止温度 T_e ，温度下降系数 α ，最大迭代次数 $n_{\max}$

输出 最优控制平面布局 $\{y\}^*$

- 1) $n = 0$
- 2) 随机选择控制器的部署位置 y_0
- 3) 按照最短时延原则和负载约束计算出每

一个交换机的控制器归属，然后计算每一个交换机到控制器的最短控制路径和控制路径上的连边经过度

4) 计算出 $(v_G)_0$

5) while $T \geq T_e$

6) 按照更新策略计算出控制器的部署位置 y_{n+1}

7) 按照最短时延原则和负载约束计算出每一个交换机的控制器归属，然后计算每一个节点到控制器的最短路径和经过度

8) 计算出 $(v_G)_{n+1}$

9) if $\Delta < 0$

10) $\{y\}^* = \{y\}_{n+1}$

11) else if $P(\Delta, T) \geq \text{random}(1)$

12) $\{y\}^* = \{y\}_{n+1}$

13) end if

14) end if

15) if $n > n_{\max}$

16) $T_{n+1} = \alpha T_n, x = 1$

17) else

18) $n = n + 1$

19) end if

20) end while

21) return $\{y\}^*$

算法的复杂度分析：首先计算网络全部节点到控制器的最短路径的复杂度为 $O(VE)$ ，计算网络中每一条连边的经过度的复杂度为 $O(E)$ ，故计算出控制器布局所需要的复杂度可以表示为 $O(VE^2)$ ，考虑到模拟退火算法的复杂度和初始温度 T_0 、终止温度 T_e 、温度下降系数 α 及最大迭代次数 $n_{\max}$ 有关，其复杂度用 $O(I)$ 表示，故算法的总复杂度可以表示为 $O(IVE^2)$ 。

5 实验

5.1 实验环境及参数说明

实验的软硬件环境及相关参数设定表示如下。

1) 本文的仿真平台为 Matlab 2012b，仿真环境为 Win10 64 位系统，Intel Core i5 4200 处理器，8 GB 内存。

2) 本文在不同规模的随机网络进行仿真实验, 随机网络采用 Gephi^[20]生成, 生成的随机网络的规模分别为 100 个交换机、200 个交换机和 300 个交换机, 相邻交换机之间的时延范围为 2 ms 至 6 ms。随机网络中本文设定每一个交换机产生的负载都是相同的, 即 $u = u_0$, 且和控制器的最大负载 U_c 的关系表示为

$$MU_c = (1 + \gamma)Vu_0 \quad (19)$$

其中, V 表示网络中的交换机数量, M 表示网络中的控制器数量, γ 表示冗余系数, 冗余系数表明每一个控制器都拥有冗余负载以应对网络中的异常流量。

在上述网络中, 本文假定节点间信号的传输速度为光速 (2×10^8 m/s) 的 $\frac{2}{3}$, 故节点间的时延可以利用如下公式进行计算。

$$l_{\max} = \frac{d_{ij}}{2 \times 10^8} \quad (20)$$

3) 本文分别选择基于最短路径规划的控制平面布局算法 (Latency)^[21]、基于不相交路径的控制平面布局算法 (Disjoint)^[11] 和所提 (Proposed) 算法对比。基于最短路径规划的控制平面布局算法的目标是选择出正常控制路径和故障控制路径差值最小的控制平面布局算法; 基于不相交路径的控制平面布局方法则在基于最短路径规划的控制平面布局算法的基础上, 为备份路径增加不相交的要求, 进而保证备份控制路径的可靠性。

4) 仿真在不同网络规模网络中, 分别验证连边故障时最多受影响的交换机数 (maximum affected switches)、最大平均连边故障恢复开销 (maximum average link failure recovery cost) 以及

故障恶化时延 (latency deterioration under link failures)。其中, 平均连边故障恢复开销表示受到连边故障影响的交换机的平均故障恢复开销, 其公式表示为

$$\bar{c}_{ij} = \frac{\sum_k h_k^{ij} b_k}{\sum_k h_k^{ij}} \quad (21)$$

5) 本文设定模拟退火算法的初始温度为 99 °C, 终止温度为 90 °C, 温度下降系数为 0.999 5, 最大迭代次数为 1 000。

5.2 仿真分析

5.2.1 连边故障时受影响的交换机数

首先, 分析不同规模网络下, 3 种方法计算出的控制平面布局中, 一条连边出现故障时, 最多受影响的交换机数, 如图 2 所示。

图 2 展示了不同网络规模下随着控制器数量的增加, 一条连边发生故障时, 最多受到连边故障影响交换机数量的变化曲线。在实验中明显可以注意到, 随着网络规模不断变大, 连边故障影响的交换机数量在不断增加, Proposed 算法和 Disjoint 算法的性能明显优于 Latency 算法。虽然 Proposed 方法的性能略差于 Disjoint 算法, 但值得注意的是, 无论如何变化网络规模及控制器的部署数量, Proposed 算法和 Disjoint 算法都十分接近, 这表明 Proposed 算法部署出的控制平面的控制路径也大多是不相交的。

5.2.2 最大平均连边故障恢复开销

然后, 分析不同规模网络下, 3 种算法的最大平均连边故障恢复开销, 如图 3 所示。

图 3 展示了不同网络规模下随着控制器数量变化, 一条连边发生故障时, 平均连边故障恢复

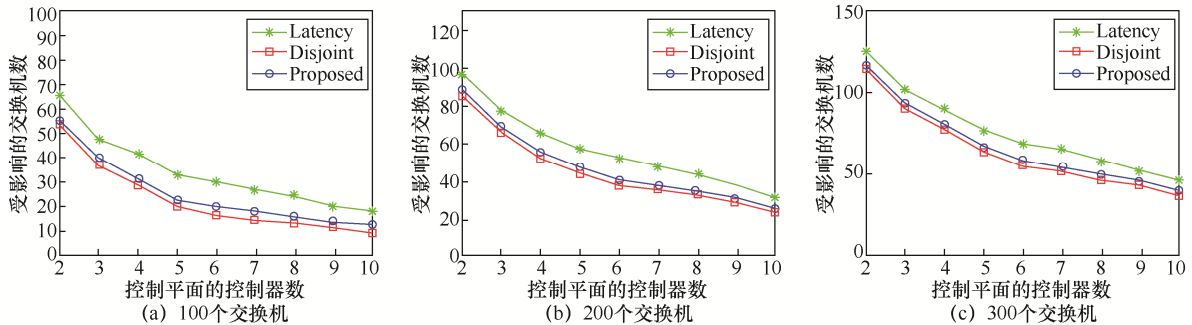


图 2 3 种算法下连边故障时受影响的交换机数
Figure 2 Maximum affected switches under different algorithms

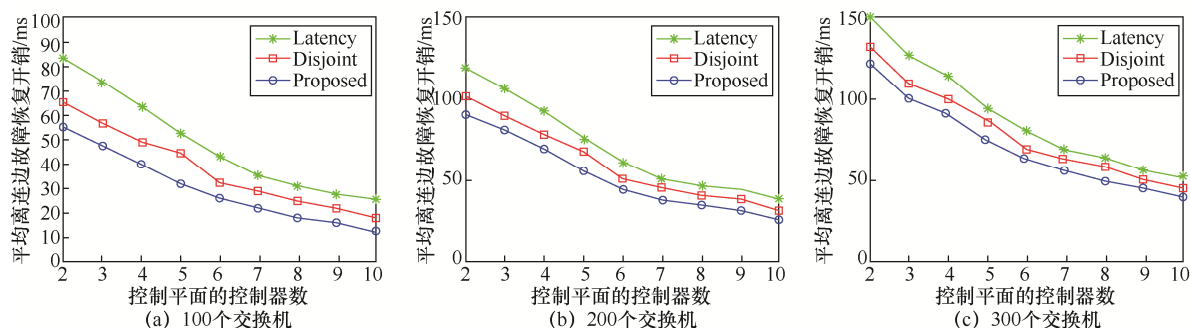


图3 3种算法下的最大平均连边故障恢复开销
Figure 3 Maximum average link failure recovery cost under different algorithms

开销的变化曲线。在实验中明显可以注意到,随着网络规模变大,3种算法的平均连边故障恢复开销都明显上升,在不同规模的网络中,Proposed算法都优于Disjoint算法和Latency算法。随着控制器部署数量的增加,虽然3种算法的平均连边故障恢复开销都会下降,但Proposed方法仍明显优于Disjoint算法和Latency算法。

5.2.3 故障恶化时延

最后,分析不同规模网络下,3种算法计算出的控制平面布局下,网络的故障恶化时延,如图4所示。

图4展示了不同网络规模下随着控制器数量增加时,3种布局算法下的故障恶化时延。Proposed算法的时延性能明显优于Disjoint算法,差于Latency算法,但是随着控制器部署数量的增加,所提算法的性能不断接近Latency算法,这表明在控制器部署数量较多的大型网络中,Proposed算法完全可以在尽量少损失网络通信性能的同时,有效提高网络的可靠性。

6 结束语

本文着重对网络中连边出现故障时,网络的故障恢复开销进行研究,针对性地设计了控制平面部署模型,并提出了一种基于模拟退火思想的控制平面布局算法。实验结果表明,本文提出的控制器部署策略有效降低网络在连边出现故障时的故障恢复成本,可以在少损失网络时延代价的同时,有效提高网络的鲁棒性。

参考文献:

- [1] KILLI B P R, RAO S V. Capacitated next controller placement in software defined networks[J]. IEEE Transactions on Network and Service Management, 2017, 14(3): 514 - 527.
- [2] 施江勇, 杨岳湘, 李文华. 基于SDN的云安全应用研究综述[J]. 网络与信息安全学报, 2017, 3(5): 10-25.
SHI J Y, YANG Y X, LI W H. Research on SDN-based cloud security application[J]. Chinese Journal of Network and Information Security, 2017, 3(5): 10-25.
- [3] KOPONEN T, CASADO M, GUDE N, et al. Onix: a distributed

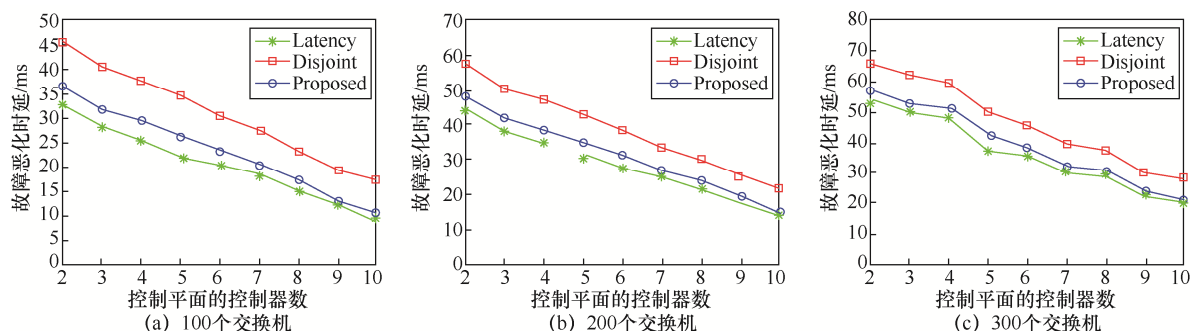


图4 3种算法下故障恶化时延
Figure 4 Different algorithms' latency deterioration under link failures

- control platform for large-scale production networks[C]//Usenix Conference. 2010: 351-364.
- [4] YEGANEH S H, GANJALI Y. Kandoo: a framework for efficient and scalable offloading of control applications[C]//Workshop on Hot Topics in Software Defined Networks. 2012: 19-24.
- [5] BERDE P, HART J, et al. ONOS: towards an open distributed SDN OS[C]//Workshop on Hot Topics in Software Defined Networking. 2014: 1-6.
- [6] LYU Y Y, GUO Y F, QI C, et al. Flow consistency in an intensive SDN security architecture with multiple controllers[J]. Chinese Journal of Network and Information Security, 2017, 3(12): 62-78.
- [7] 高先明, 王宝生, 邓文平. SDN 网络中控制器放置问题综述[J]. 通信学报, 2017, 38(7): 155-164.
- GAO X M, WANG B S, DENG W P. Survey of controller placement problem in software defined network[J]. Journal on Communications, 2017, 38(7): 155-164.
- [8] HELLER B, SHERWOOD R, MCKEOWN N. The controller placement problem[J]. ACM Sigcomm Computer Communication Review, 2012, 42(4): 473-478.
- [9] HOCK D, HARTMANN M, GEBERT S, et al. Pareto-optimal resilient controller placement in SDN-based core networks[C]//The International Teletraffic Congress. 2013: 1-9.
- [10] LIU J, XIE R. Reliability-based controller placement algorithm in software defined networking[J]. Computer Science and Information Systems, 2016, 13(2): 547-560.
- [11] LUCAS F Müller, OLIVEIRA R R, LUIZELLI M C, et al. Survivor: an enhanced controller placement strategy for improving SDN survivability[C]//IEEE Global Communications Conference. 2014: 1909-1915.
- [12] 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.
- [13] HU Y N, WANG W D, GONG X Y, et al. Reliability-aware controller placement for Software-Defined Networks[C]//International Symposium on Integrated Network Management. 2013: 672-675.
- [14] HU Y N, WANG W D, GONG X Y, 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.
- [15] ZHANG Y, BEHESHTI N, TATIPAMULA M. On resilience of split-architecture networks[C]//Global Telecommunications Conference. 2012: 1-6.
- [16] HU Y N, WANG W D, GONG X Y, et al. On reliability-optimized controller placement for software-defined networks[J]. China Communication, 2014, 11(2): 38-54.
- [17] VIZARRETA P, MACHUCA C M, KELLERER W. Controller placement strategies for a resilient SDN control plane[C]//International Workshop on Resilient Networks Design & Modeling. 2016: 253-259.
- [18] SAVAS S S, TORNATORE M, DIKBIYIK F, et al. RASCAR: recovery-aware switch-controller assignment and routing in SDN[J]. IEEE Transactions on Network and Service Management, 2018, 15(4): 1222-1234.
- [19] AI J, GUO Z, CHEN H. Thwarting worm spread in heterogeneous networks with diverse variant placement[J]. IEEE Communications Letters, 2018, 22(7): 1346-1349.
- [20] BASTIAN M, HEYMANN S, JACOMY M. Gephi: an open source software for exploring and manipulating networks[C]//International AAAI Conference on Weblogs and Social Media. 2009.
- [21] ADRICHEM N L M V, ASTEN B J V, KUIPERS F A. Fast recovery in software-defined networks[C]//Third European Workshop on Software Defined Networks. 2014.

[作者简介]



吴奇 (1991-), 男, 江苏徐州人, 信息工程大学博士生, 主要研究方向为网络安全。



陈鸿昶 (1964-), 男, 河南郑州人, 信息工程大学研究员、博士生导师, 主要研究方向为网络安全、大数据。