

# 一种软件定义网络中的控制器热备份及选举算法

王文博,汪斌强,陈飞宇,王志明,宫阳阳

(国家数字交换系统工程技术研究中心,河南郑州 450002)

**摘要:** 多控制器软件定义网络中交换机迁移策略计算空间大、考虑因素单一且无法立刻迁移出故障域内交换机. 对此,本文将原交换机迁移问题优化成为控制器的热备份及选举问题(Controller Hot Backup and Election Problem, CHBE Problem),对控制器进行热备份,并设计相应的备份空间确定算法和主控制器选举算法,通过权衡信息交互、失联性、负载失衡和跨域通信四种代价实现合理的网络构建. 实验结果表明,与现有算法相比,负载均衡程度和跨域通信问题改善明显,备份空间平均缩小了65%;在OS3E拓扑中加权后的综合评价最高提升了71%.

**关键词:** 软件定义网络;多控制器;热备份;控制器选举;故障恢复

**中图分类号:** TP393 **文献标识码:** A **文章编号:** 0372-2112 (2016)04-0913-07

**电子学报 URL:** <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2016.04.023

## The Controller Hot Backup and Election Algorithms in Software Defined Networks

WANG Wen-bo, WANG Bin-qiang, CHEN Fei-yu, WANG Zhi-ming, GONG Yang-yang

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

**Abstract:** In multi-domain software defined networks, the existed switch migration strategies always have big computation space and only consider single migration factor. Once the controller gets broken, the switches cannot be migrated immediately. This paper presents the controller hot backup and election problem and designs the corresponding algorithms to find the backup space and complete the master controller election. To reasonably build the network, this paper considers the message exchange cost, the disconnectivity cost, load imbalance cost and cross domain cost. Results show that, compared with the existed algorithms, the proposed algorithms reduce the controller backup space by an average of 65%, and increase the weighted comprehensive evaluation up to 71% in OS3E topology.

**Key words:** software defined networks; multi-controller; hot backup; controller election; failover

### 1 引言

目前,新型网络体系结构,如智慧协同网络<sup>[1]</sup>,可重构网络<sup>[2]</sup>,软件定义网络,为解决传统网络的僵化问题提供了新的方案.其中,软件定义网络由于其易管理、可编程的特点,受到了广泛关注.近年来,随着业务需求和网络规模的不断增大,软件定义网络控制平面的扩展性受到巨大挑战<sup>[3,4]</sup>.其中,一个重要方面就是由于多控制器架构“命运共享”(fate sharing)所带来的控制平面的脆弱性.

针对该问题的解决方法大致能够分为两类.第1类方法通过静态布置控制器位置增强控制平面的可靠

性<sup>[5-9]</sup>.这类方法单纯改变节点分布,无法对网络动态调节,使网络处于一种“哑”的,“无法移动”的状态.第2类方法在控制器布局确定之后,通过迁移故障域或负载域内的交换机,对网络进行弹性管控<sup>[10]</sup>.这类方法虽然能动态调整网络,但往往考虑单一,仅关注片面的迁移代价<sup>[9-12]</sup>.导致迁移后某单一方面性能突出,其他方面性能较差.并且每一次完成迁移都需要对整个控制器集合进行重新计算,复杂度高、时延较大,极易造成通信的不一致性.

针对上述问题,本文在动态调整的基础上,提出一种对控制器进行热备份的迁移方法.先借助 openflow1.4 协议<sup>[13]</sup>定义了交换机的控制器备份空间,将原

交换机迁移问题优化成为控制器的热备份及其相应的 master 控制器选举问题 (Controller Hot Backup and Election Problem, CHBE Problem). 针对之前的迁移策略考虑因素单一, 对信息交互代价、失联性代价、负载失衡代价和跨域通信代价等四种代价进行了描述. 设计了两个算法, 分别确定控制器备份空间和完成主控制器选举, 使交换机每次迁移时的计算空间从整个控制器集合缩小到了控制器备份空间. 就目前的文献调查情况分析, 通过为控制器找寻备份空间来实现迁移是一种新的尝试, 为迁移策略的实现和设计提供了新的数学模型和思路.

## 2 相关研究

国内外许多学者都对第 1 类方法给予了关注. 文献 [5] 定义了控制平面的可靠性并提出基于贪心算法的控制器布局算法; 文献 [6] 在研究可靠性的基础上, 关注了控制器个数的约束条件. 此外, 还有针对负载均衡的 CCP 布局方法 [7]; 针对故障恢复的 FTCP 布局方法 [8] 和针对多种因素的帕累托优化布局方法 [9] 等等. 上述布局方式虽然对改善控制平面的脆弱性意义重大, 但是仍然无法从根本上解决“命运共享”带来的失效问题.

针对第 1 类方法中存在的问题, openflow1.4 协议描述了一种动态调整的典型情况: 每个交换机除了连接一个 master 控制器, 还连接多个其余控制器作为

slave 控制器. 当该控制器发生故障或其域内交换机在某时刻流请求激增时, 需要对该域内交换机实现向外迁移. 文献 [9] 将距离和时延作为迁移的首要衡量因素; ElastiCon [10] 从弹性控制出发衡量了迁移前后负载均衡程度的改善; DCPP [11] 通过添加监视模块, 根据流量的实时变化对交换机进行动态调整. 这类动态调整策略极大的改善了第 1 类方法的固有缺陷, 保证了网络的通信畅通. 但是迁移代价衡量单一, 计算复杂度高、时延较大, 难以满足大型网络的需要.

## 3 CHBE 的建模与算法描述

本节对交换机迁移问题进行了重述和建模, 定义了备份空间和备份隶属矩阵, 将原问题建模成为一个考虑多种代价的整数规划问题, 继而给出了相应的 CHBE 算法并对其进行了描述.

### 3.1 问题的重述和建模

多控制器布局完成后, 本文将此场景描述为 openflow1.4 协议中提出的, 一个 master 控制器、多个 slave 控制器的应用场景.

如图 1(a) 所示, 假设网络被分为多个域, 每个域内均部署一个控制器. 以 s4 为例 (图 1(b)), 若 s4 的备份控制器空间为 (c1、c2、c3), 当 c0 发生不可逆故障或 s4 需要外迁时, 可直接在其控制器备份空间内完成 master 控制器的重新选举.

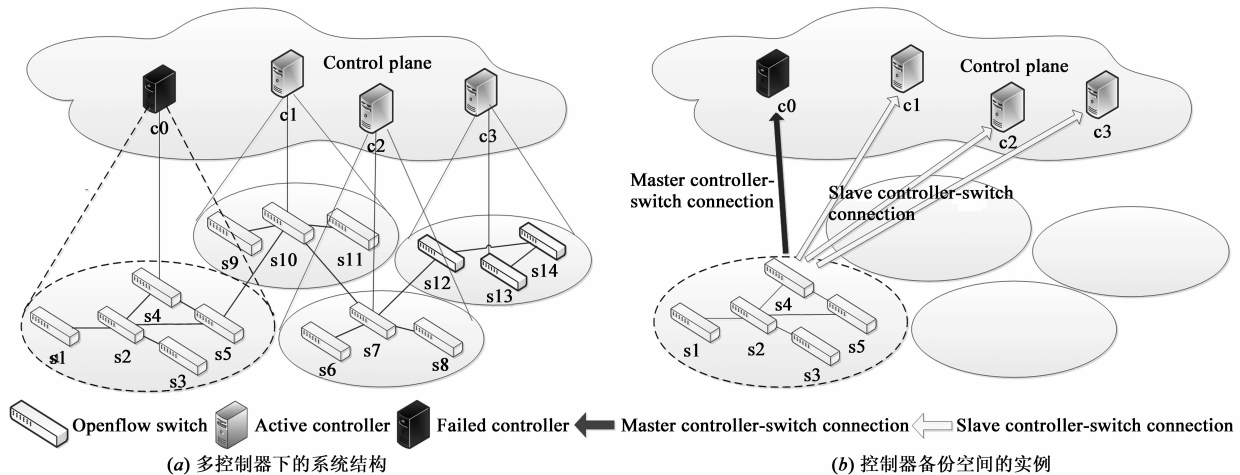


图1 系统结构图

本文所要研究的问题即是:

- (1) 为全网中的每一个 openflow 交换机确定备份控制器的个数及位置.
- (2) 控制器备份空间一旦确定, 迁移时完成 master 控制器的选举.

介绍模型之前, 完成两个假设并加以说明:

**假设(1)** 同一时刻, 最多只有一个网络组件发生

故障.

**假设(2)** 采用域内不同组件故障相互独立的研究模型.

本文采取了一种广泛应用的故障概率模型 [14], 对于网络内不同组件的故障发生在统计学上认为相互独立. 同时, 根据文献 [6], 网络中同时有两个组件发生故障的概率很小, 故假设同一时刻, 最多只有一个网络组

件发生故障.

模型把网络建模成为一个无向图  $G=(S,E)$ , 其中  $S$  是无向图中的节点的集合,  $E$  是边的集合.  $G$  分成不同的域, 每一个域部属一个控制器. 即  $G=\{D_1, D_2, \dots, D_k, \dots, D_n\}$ , 且任意两个域没有重叠. 控制器集合为  $V=\{v_1, v_2, \dots, v_j, \dots, v_n\}$ , 域  $D_k$  内的控制器为  $v_k$ . 对于域  $D_k$ , 域内的 openflow 交换机集合为  $\{s_{k1}, s_{k2}, \dots, s_{km_k}\}$ , 数量为  $m_k$  个. 同时定义了备份隶属矩阵  $X=[x_{sv}]_{|S| \times |V|}$ , 当  $x_{sv}$  等于 1 时, 说明控制器  $v$  为交换机  $s$  进行备份, 反之则为 0.

为了解决备份控制器的布置位置和数量问题, 本文考虑了以下四种代价:

(1) 交换机与多控制器间的信息交互. 以域  $D_k$  内的控制器  $v_k$  发生故障为例 (描述其他三个代价时, 此条件不变), 剩余控制器集合为  $V^*=V \setminus \{v_k\}$ . 当用  $d(s,v)$  描述节点  $s$  和  $v$  之间的最短距离时, 该域内多控制器之间的信息交互的平均代价  $C_{kr}$  可以表示为:

$$C_{kr} = \frac{1}{|D_k|} \sum_{s_u \in D_k} \sum_{v_j \in V^*} d(s_{ki}, v_j) \cdot x_{s_u v_j} \quad (1)$$

(2) 失联性. 本文认为, 当 openflow 交换机再没有可供选择的备份控制器时, 该 openflow 交换机失联, 即它将无法得到来自控制器的任何指示. 文献[12]中已经做出了关于网络稳定性的研究工作. 在这里, 假设每条链路的故障概率为  $l$ , 定义域  $D_k$  内的 openflow 交换机的平均失联性  $C_{kd}$  为:

$$C_{kd} = \frac{1}{|D_k|} \sum_{s_u \in D_k} \prod_{v_j \in V^*} (1 - l^{d(s_u, v_j)}) \cdot x_{s_u v_j} \quad (2)$$

(3) 负载均衡. 虽然网络流量时刻变化, 但由于备份空间提供的是一族控制器, 在考虑负载均衡时, 可以用空间确定时的流量平均值, 经验性的圈定这一族控制器的大概分布范围.

本模型使用等概率的分配方法描述了交换机迁移为控制器带来的负载贡献,  $r_{v_j}$  表示网络无故障时控制器  $v_j$  所要处理的负载. 本文使用其所需处理的 PACKET\_IN 消息的数量来近似的描述控制器负载<sup>[7]</sup>.  $r_{s_u}$  表示域  $D_k$  中的 openflow 交换机  $s_{ki}$  向 master 控制器提出的平均流请求数, 由于 openflow 交换机的迁移, 控制器  $v_j$  所需要处理的新的流请求数  $R_{v_j}$  为:

$$R_{v_j} = r_{v_j} + \sum_{s_u \in D_k} \frac{1}{\sum_{v_j \in V^*} x_{s_u v_j}} \cdot r_{s_u} \cdot x_{s_u v_j} \quad (3)$$

用  $L_{v_j}$  来表示控制器  $v_j$  所能处理的流请求数的最大值, 也即控制器  $v_j$  的容量. 此时用迁移完成之后剩余控制器容量使用率的方差来描述负载均衡代价  $C_{kb}$ :

$$C_{kb} = \text{Var}(R_{v_j}/L_{v_j})_{v_j \in V^*} \quad (4)$$

其中, 式(4)中的  $\text{Var}(R_{v_j}/L_{v_j})_{v_j \in V^*}$  表示对  $V^*$  中的  $|V^*|$  个元

素构成的向量  $(\frac{R_{v_1}}{L_{v_1}}, \frac{R_{v_2}}{L_{v_2}}, \dots, \frac{R_{v_{|V^*|}}}{L_{v_{|V^*|}}})$  取方差.

(4) 多域间通信. 当流经过不同域时, 交换机需要向所属的控制器询问路径, 将带来由于多域产生的域间通信开销<sup>[11]</sup>. 用  $P_{s_u v_j}$  表示 openflow 交换机  $s_{ki}$  到控制器  $v_j$  的最短路径中经过的不同域所对应控制器的集合. 由此, 多域间通信的平均开销  $C_i$  可以表达为:

$$C_{ki} = \frac{1}{|D_k|} \sum_{s_u \in D_k} \sum_{v_j \in V^*} r_{s_u} \cdot x_{s_u v_j} \cdot \sum_{s_m \in D_n} \frac{d(s_{mi}, v_m)}{|D_m|} \quad (5)$$

至此, 所考虑的四种种代价已经描述完毕, 式(6)中的综合代价  $\text{cost}_k$  用于描述算法的性能. 本模型的目标即是使式(6)达到最小值. 其中,  $\alpha, \beta, \gamma, \mu$  是用于权衡四种代价权重的常数.

$$\min \text{cost}_k = \alpha C_{kr} + \beta C_{kd} + \gamma C_{kb} + \mu C_{ki} \quad (6)$$

$$\forall i, j \quad x_{s_i v_j} \in \{0, 1\} \quad (7)$$

$$\forall v_j \in V^*, s_i \in \{s_{k1}, s_{k2}, \dots, s_{km_k}\} \quad \sum_j x_{s_i v_j} \geq 1 \quad (8)$$

式(7)和式(8)是模型的约束条件. 式(7)保证了备份隶属矩阵中的元素是二进制变量. 式(8)保证了每个 openflow 交换机至少存在一个备份控制器. 此时, 即得到域  $D_k$  内的 openflow 交换机的备份情况  $[x_{sv}]_{|D_k| \times |V^*|}$ . 通过对整个网络遍历, 将得到全网的备份隶属矩阵.

### 3.2 算法描述

本节介绍 CHBE 算法 (CHBEA I 和 CBHEA II). 算法分为两个部分: CHBEA I 完成控制器备份空间确定, CHBEA II 完成确定后的 master 控制器选举.

CHBEA I 是基于模拟退火算法 (Simulated Annealing) 的一种启发式算法. 算法的流程主要是, 每次计算一个域内的备份隶属矩阵, 然后遍历全网内的其他域, 最终将所得到的子备份隶属矩阵进行整合, 得到优化后的最终解 (行 14). 算法首先需要有一个初始值作为计算的起点, 在这里选择了全 1 矩阵. 通过随机改变备份隶属矩阵中的元素向最优解进行靠近. 每次迭代时, 如果新获得的  $\text{cost}_k$  值向优化目标靠近, 就接受新的备份隶属矩阵 (行 7); 否则, 仅以一定概率接受新的备份隶属矩阵 (行 9). 当计算完成域  $D_k$  的子备份隶属矩阵  $\text{now}_k$  后, 再计算其他域, 直到得到子备份隶属矩阵的向量  $\{\text{now}_1, \text{now}_2, \dots, \text{now}_n\}$ , 进行整合 (行 14).

#### CHBEA I : 控制器备份空间确定算法

输入: 网络拓扑图,  $G(S, E)$   
 控制器集合,  $V$   
 控制器容量向量,  $L$

每台交换机的流请求,  $r_{s_{ki}}$   
 带入算法的初始解,  $\chi$   
 输出: 备份隶属矩阵,  $X$

```

1 foreach  $D_k \in \{D_1, D_2, \dots, D_n\}$  do
2    $now_k \leftarrow \chi$ 
3   while(  $T > T_{\min}$  )
4      $new_k \leftarrow$  返回一个满足约束的可行解  $now_k$ 
5      $\Delta \leftarrow cost_k(now_k) - cost_k(new_k)$ 
6     If (  $\Delta \geq 0$  )
7        $now_k \leftarrow new_k$ 
8     else
9        $now_k \leftarrow$  以  $e^{-\frac{\Delta}{T}}$  的概率将  $new_k$  赋值
10    endif
11     $T = T \cdot \lambda$ 
12  endwhile
13 end
14 通过整合子矩阵集合  $\{now_1, now_2, \dots, now_n\}$  得到  $X$ 
```

CHBEA II 中, 首先定义了 master 控制器分配矩阵  $Y = [y_{sv}]_{|S| \times |V|}$ . 当  $y_{s,v_j} = 1$  时, 意为将控制器  $v_j$  作为交换机  $s_{ki}$  的 master 控制器. 算法二输入算法一的结果, 从交换机  $s_{ki}$  的备份控制器空间中选举 master 控制器, 即  $v_j \in \{\bar{V} | \forall v_m \in \bar{V}, x_{s_{ki}v_m} = 1\}$  (行 4). 在选举过程中, 由于之前确定备份空间时已经考虑了信息交互代价和失联性代价, 这两个指标对于已经选定备份空间的同一个交换机来讲, 在控制器的选举上并不构成影响因素. 这里, 按照式(4)和式(5)计算控制器  $v_i$  作为交换机  $s_{ki}$  的 master 控制器的代价(行 6), 此时式中计算的流量值是当时流量的真实值, 并非确定备份空间时的流量数据. 直到找到产生最小代价的控制器节点  $v_j$  作为 master 控制器. 迭代的顺序取决于故障域内的交换机节点距非故障域边缘的距离(行 3). 这样做的目的是为了尽量杜绝交换机分配时的跨域问题. 每次迭代结束后, 需要更新非故障域和控制器已接受的流请求数集合(行 10, 行 11), 以便进入下一次迭代. 直到将故障域中的所有交换机都分配到新的控制器上, 得到最终的 master 控制器分配矩阵  $[y_{sv}]_{|S| \times |V|}$ .

#### CHBEA II: 主控制器选举算法

输入: 网络拓扑图,  $G(S, E)$   
 控制器集合,  $V$   
 控制器容量向量,  $L$   
 备份隶属矩阵,  $X$   
 控制器负载向量,  $r_{v_j}$   
 输出: 控制器分配矩阵,  $Y$

```

1 foreach  $D_k \in \{D_1, D_2, \dots, D_n\}$  do
2    $M = G \setminus D_k$ 
3   将  $D_k$  域中的交换机按照距  $M$  边缘升序排列
4   foreach  $s_{ki} \in D_k$  do
```

```

5   foreach  $v_j \in \{\bar{V} | \forall v_m \in \bar{V}, x_{s_{ki}v_m} = 1\}$  do
6      $c_j \leftarrow$  按照式(4)和(5)计算将  $s_{ki}$  分配给  $v_j$  的代价
7   end
8   将  $s_{ki}$  分配给  $v_j = \arg \min c_j$ 
9    $y_{s_{ki}v_j} \leftarrow 1$ 
10   $M \leftarrow M + \{s_{ki}\}$ 
11   $r \leftarrow$  更新控制器的负载向量
12 End
13 End
```

## 4 仿真分析

### 4.1 仿真环境搭建

关于环境搭建和参数设置, 本文做出几点说明:

(1) 关于拓扑的选择和控制器放置位置. 本文首先使用 OS3E<sup>[15]</sup> 完成算法验证, 然后在 topology zoo<sup>[16]</sup> 中提供的公共拓扑上重新做了实验. 在控制器的放置上, 均采取文献[3]中提到的平均通信开销最小的方法. 使用 topology zoo 进行试验时, 本文一律按照控制器数量占交换机节点总数的 17% 的比例进行布置. 这样做的原因是, 当布置过少控制器时, 由于备份空间缩小带来的计算增益将不再明显, 文献[6]中提出, 当布置控制器所占交换机节点总数比例为 20% 左右时, 带来的控制路径损失(control path loss)最小.

(2) 交换机负载和控制器容量的设置. 为了更好的搭建仿真环境, 本文借鉴了文献[7]中的部分设置, 但是作了一些修改. 平均流产生速率设为 300K/s. 由于控制器所搭载服务器存在单个和集群的差异, 本文为其容量作了差异化设置, 也即将 PACKET\_IN 消息的平均流量设为 8M ~ 11M. 同时, 本模型还为控制器设计了负载上限, 当控制器的负载超过这个上限时, 认为此刻已经发生故障. 在仿真的初刻, 保证负载均衡和控制器完好, 仿真过程中通过调节负载上限, 能够使控制器处理更多的流和消息.

(3) 在综合考虑各方面代价时, 关于权重系数的说明. 该权重值只是为了平衡四方面因素的影响, 改变之后, 各因素的相对关系变化不大. 在这里, 模型在对各因素进行归一化之后, 选取了 1:1:1:1 的比重, 实际工程中, 可以按照需要再作调整. 同时, 本文使用式(6)中的加权后的综合代价作为算法性能的综合评价.

### 4.2 仿真分析

为了说明算法的性能, 本文将 CHBE 算法与若干算法进行了比较说明. 由于该方向文献较少, 本文借鉴了以往研究中的控制器布置算法, 将某些布置算法的思想作为控制器备份空间确定的一个因素. 其中, 距控制器跳数最小算法(Shortest Path Algorithm, SPA)和距离域最短算法(Nearest Domain Algorithm, NDA)参看文

献[9],考虑控制器布置的负载均衡算法(Load Balancing Algorithm, LBA)在文献[7]中有所提及. 上述三种算法均只考虑了交换机迁移或控制器布置的单一因素,其中,SP算法是将故障域内的交换机分配给离它最近的控制器,ND算法是将故障域内的交换机分配给距离非故障域边缘最近的对应控制器,而LB算法是每次分配都选择剩余控制器容量使用率最低的控制器. 本文在具体仿真中还做了细微调整,保证了实验场景相

同. 例如,将备份空间引入上述三种算法并设置为整个控制器集合.

本文首先使用了 OS3E 网络对 CHBE 算法进行了验证. 具体仿真所使用的网络拓扑情况如图 2(a)所示情况. 图 2(b)、图 2(c)是分别在图 2(a)的基础上运行 CHBE 算法和 SP 算法的情况示意. 可以明显看出,CHBE 算法能较好的重新布置交换机,而 SP 算法造成了跨域问题(图 2(c)),这将严重阻碍网络的正常通信.

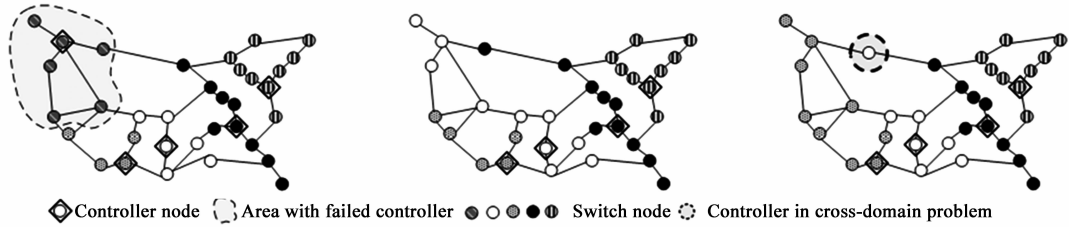


图2 OS3E网络分域示意图(控制器的个数:  $k=5$ )

接着,研究拓展了 OS3E 中控制器的数量,仿真算法对各个因素的影响. 实验关注点设置为单域内控制器发生故障(域的划分按照距离控制器节点的远近进行,开始时保证网络负载均衡<sup>[11]</sup>),完成 master 控制器选举之后的时刻.

算法为交换机找寻的备份个数的平均值和多控制器信息交互代价也相应增加. 但是相较于其他三种算法,CHBE 算法选取了较小的备份空间并花费了较少的通信代价. 根本原因在于 CHBE 算法找到的备份空间只是整个控制器集合的一个子集.

图 3(a)和图 3(b)说明了随着控制器数量的增加,

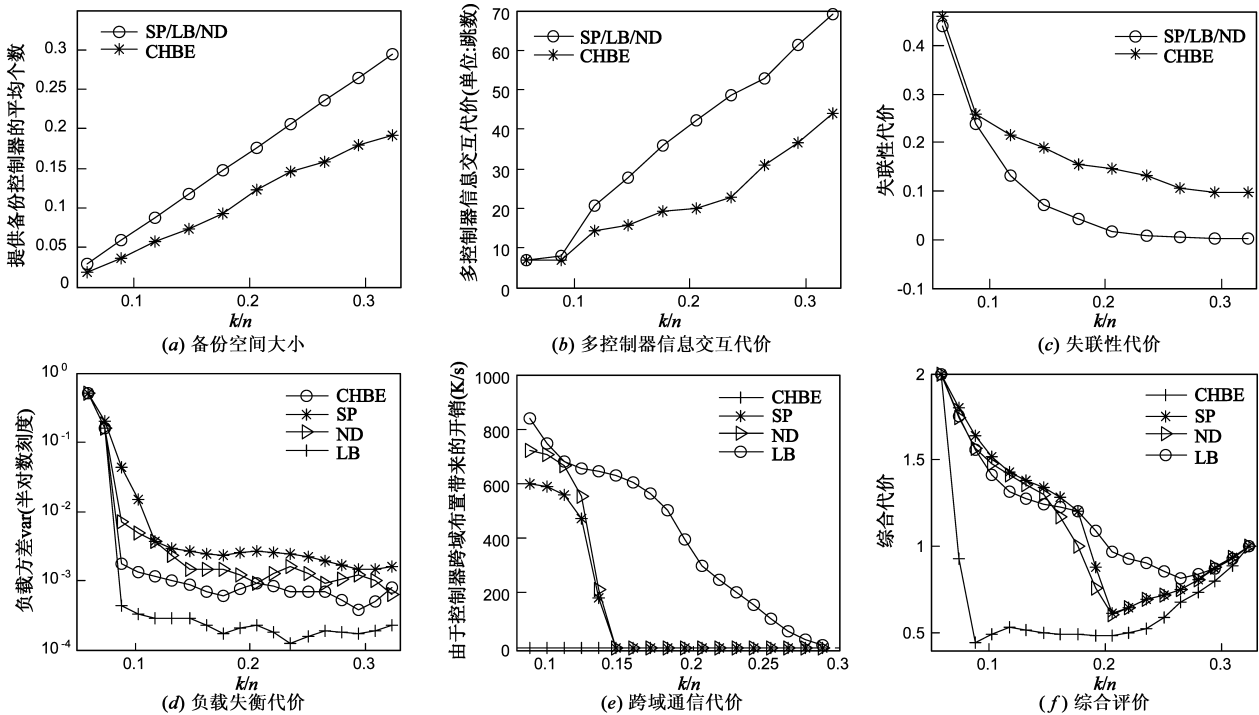


图3 算法性能仿真图

图 3(c) ~ (f) 分别对比了算法在失联性、负载方差、跨域通信和综合代价上的不同表现:

(1) 失联性代价和信息交互代价相互矛盾,即控制

器数量越多,失联性越小. 这是由于备份空间越大,失联的可能性越低,但是多控制器造成的通信开销也相应增加.

(2)在负载方差方面,控制器数量越多,负载越均衡.对比不同算法,CHBE 算法仅劣于 LB 算法,这是由于 LB 算法仅考虑负载均衡,没有其他因素的约束.

(3)由于考虑了跨域通信开销,四种算法中仅有 CHBE 算法始终没有出现跨域问题.整体看来,控制器布置数量越多,故障域内的交换机个数越少,跨域问题越不明显.LB 算法的性能最差,这是由于当流请求数占控制器容量比例不大时,LB 算法为交换机提供的可选控制器空间增大.

(4)CHBE 算法在个别单项因素上稍有欠缺,但是综合评价优于其他算法.当  $k/n = 0.12$  时(表 1),CHBE 算法较其他三种算法,在信息交互和跨域通信两个方面有明显优势,负载均衡性能仅次于 LB 算法,综合评价最大提高了 68%.当  $k/n = 0.088$  时,综合评价提升最大,达到 71%.

表 1 四种算法的性能比较

考量因素 算法名称	信息交互代价 (跳数)	失联性 代价	负载失 衡代价 ( $\times 10^{-4}$ )	跨域 通信代价 (K/s)	综合 评价
CHBE	9.6	0.24	13	0	0.49
SP	13.2	0.19	147	590	1.52
ND	13.2	0.19	49	708	1.48
LB	13.2	0.19	3	750	1.41

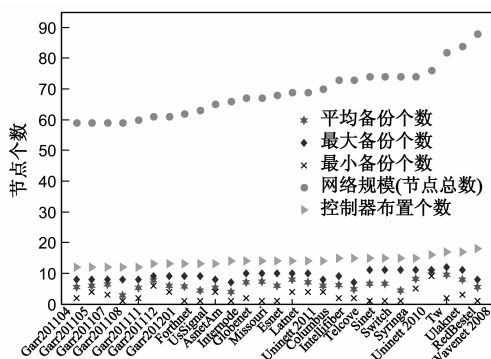


图 4 算法在不同拓扑中的备份空间大小

最后,本文在 topology zoo 提供的 261 个公共拓扑中挑选了 27 个拓扑重做实验.如图 4,横轴是拓扑名称,纵轴是节点个数.这些拓扑的节点数在 60~90 之间.本文在这些拓扑中使用了 CHBE 算法进行仿真分析,说明了算法在不同拓扑上的适用性.实验发现,CHBE 算法将交换机的备份控制器数量平均缩小了 65%,大大缩小了备份空间,达到了四种代价权衡下的综合结果.

### 5 结论

针对多控制器软件定义网络中交换机迁移策略计

算空间大、考虑因素单一且迁移时延较大等问题,本文优化并回答了 CHBE 问题:(1)为全网中的每一个 openflow 交换机确定备份控制器的个数及位置;(2)控制器备份空间一旦确定,迁移时完成 master 控制器的重新选举.

本文首次采用热备份的方法完成了对交换机迁移策略的设计.将上述问题建模成一个考虑信息交互,失联性,负载失衡和跨域通信等多种代价的整数规划问题.通过基于模拟退火算法的 CHBEA I 确定了控制器备份空间,通过 CHBEA II 完成了 master 控制器选举.与 SP、ND、LB 算法相比,所提算法虽然在个别单项因素的考量上稍有欠缺,但在负载均衡和跨域通信两个方面改善明显,综合评价最大提升了 71%,并将控制器备份空间平均减小了 65%.通过引入热备份的方法,为迁移策略的实现和设计提供了新的数学模型和努力方向.本文关注了单个控制器故障的一般场景,多点同时故障时通过调整控制器的流请求情况、更新非故障域,CHBE 算法依然能够给出此应用场景下的交换机迁移方法.对于更复杂的多点故障和级联故障情景,将在下一步的工作中继续研究.

### 参考文献

- [1] 张宏科,罗洪斌.智慧协同网络体系基础研究[J].电子学报,2013,41(7):1249-1254.  
Zhang Hong-ke, Luo Hong-bin. Fundamental research on theories of smart and cooperative networks[J]. Acta Electronica Sinica, 2013, 41(7): 1249-1254. (in Chinese)
- [2] 兰巨龙,程东年,胡宇翔.可重构信息通信基础网络体系研究[J].通信学报,2014,35(1):128-139.  
Lan Ju-long, Cheng Dong-nian, Hu Yu-xiang. Research on reconfigurable information communication basal network architecture[J]. Journal on Communications, 2014, 35(1): 128-139. (in Chinese)
- [3] B Heller, R Sherwood, N McKeown. The controller placement problem[A]. Proceedings of the First Workshop on Hot Topics in Software Defined Networks[C]. Helsinki, Finland: IEEE, 2012. 7-12.
- [4] D Levin, A Wundsam, B Heller, N Handigol, A Feldmann. Logically centralized?: State distribution trade-offs in software defined networks[A]. Proceedings of the First Workshop on Hot Topics in Software Defined Networks[C]. Helsinki, Finland: IEEE, 2012. 1-6.
- [5] Y Hu, W Wendong, X Gong, X Que, S Cheng. On the placement of controllers in software-defined networks[J]. The Journal of China Universities of Posts and Telecommunications, 2012, 19(2): 92-171.
- [6] Hu Y, Wang W, Gong X, et al. On reliability-optimized

- controller placement for software-defined networks [ J ]. Communications China, 2014, 11 ( 2 ) : 38 - 54.
- [ 7 ] G Yao, J Bi, Y Li, L Guo. On the capacitated controller placement problem in software defined networks [ J ]. IEEE Communications Letters, 2014, 18 ( 8 ) : 1339-1342.
- [ 8 ] Ros F J, Ruiz P M. Five nines of southbound reliability in software-defined networks [ A ]. ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking ( HotSDN ) [ C ]. Chicago, USA : IEEE, 2014. 1 - 6.
- [ 9 ] D Hock, S Gebert, M Hartmann, T Zinner, Tran-Gia. PO-CO-framework for pareto-optimal resilient controller placement in SDN-based core networks [ A ]. Network Operations and Management Symposium [ C ]. Krakow, Poland : IEEE, 2014. 1 - 2.
- [ 10 ] A Dixit, F Hao, S Mukherjee, T V Lakshman, R Kompella. Towards an elastic distributed SDN controller [ A ]. Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking [ C ]. Hong Kong, China : IEEE, 2013. 7 - 12.
- [ 11 ] M F Bari, A R Roy, S R Chowdhury, Q Zhang, M F Zhan, R Ahmed, R Boutaba. Dynamic controller provisioning in software defined networks [ A ]. Proceedings of 9th International Conference on Network and Service Management [ C ]. Zürich, Switzerland : IEEE, 2013. 18 - 25.
- [ 12 ] Y Zhang, N Beheshti, M Tatipamula. On resilience of split-architecture networks [ A ]. Global Telecommunications Conference ( GLOBECOM ) [ C ]. Houston, USA : IEEE, 2011. 1 - 6.
- [ 13 ] Open Flow Switch Specification, Version 1. 4. 0 [ OL ]. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>.
- [ 14 ] K Le, V Li. Modeling and analysis of systems with multi-mode components and dependent failures [ J ]. IEEE Transactions on Reliability, 1989, 38 ( 1 ) : 68 - 75.
- [ 15 ] Internet2 Open Science, Scholarship and Services Exchange [ OL ]. <http://www.internet2.edu/network/ose/>.
- [ 16 ] 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.

#### 作者简介



王文博 男, 1991 年 1 月出生, 河南信阳人. 国家数字交换系统工程技术研究中心硕士研究生, 主要研究方向为软件定义网络.  
E-mail : jameschenmu@ outlook. com



汪斌强 男, 1963 年 2 月生, 安徽安庆人. 国家数字交换系统工程技术研究中心教授、博士生导师, 主要研究方向为宽带信息网络, 网络安全.  
E-mail : w bq6301@ 163. com



陈飞宇 男, 1990 年 8 月出生, 河南周口人. 国家数字交换系统工程技术研究中心硕士研究生, 主要研究方向为宽带信息网络.