

# 基于网络时延的SDN逻辑一致性策略研究

宋海权<sup>1</sup>, 郭进<sup>1</sup>, 侯孟书<sup>2</sup>, 李赛飞<sup>1</sup>, 闫连山<sup>1</sup>, 王丹琛<sup>1</sup>

(1. 西南交通大学信息科学与技术学院 成都 610031; 2. 电子科技大学计算机科学与工程学院 成都 611731)

**【摘要】**对SDN网络更新过程中的暂时中间态进行了研究,阐述了因网络时延导致的控制逻辑不一致问题,并提出了一种解决方案。该方案通过在控制平面上调整安装控制规则的时序,使网络达到控制逻辑一致的目的,进一步建立了一种基于节点平均网络时延的目标优化模型,通过对模型求解得到SDN网络逻辑一致的时延最短路由路径。最后通过仿真实验验证了该方案的可行性和有效性。

**关键词** 控制逻辑一致性; 控制平面; 转发平面; 网络时延; 软件定义网络

中图分类号 TP309.3

文献标志码 A

doi:10.3969/j.issn.1001-0548.2014.05.017

## Research on the SDN Consistency Problems of the Control Logic Based on Network Delay

SONG Hai-quan<sup>1</sup>, GUO Jin<sup>1</sup>, HOU Meng-shu<sup>2</sup>, LI Sai-fei<sup>1</sup>, YAN Lian-shan<sup>1</sup>, and WANG Dan-chen<sup>1</sup>

(1. School of Information Science & Technology, Southwest Jiaotong University Chengdu 610031;

2. School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 611731)

**Abstract** This paper studies the transitional state in the software defined networking (SDN) networks during the SDN network update process. A route optimization scheme is proposed after the brief review of the logic inconsistency issues in SDN networks due to the delay in the network. In this scheme, the timing sequence of the control rules is tuned adaptively on the control plane so as to ensure that the control logic in the whole network is consistent. In addition, an objective optimization model based on the average network latency of nodes is constructed. With this model, we find the minimum-delay route while retaining the logic consistency in the SDN network. The simulation results demonstrate that the proposed scheme is feasible and effective.

**Key words** control logic consistency; control plane; forwarding plane; network delay; software-defined networking

在传统的网络体系结构中,作为核心设备的路由器包含报文转发和转发决策等功能。网络规模的急剧膨胀导致其承载的功能和结构变得模糊和臃肿不堪。SDN技术的核心思想是实现网络设备控制平面和数据转发平面解耦,这使得网络的处理层次更加扁平化<sup>[1-2]</sup>。网络设备以数据转发为主,网络的逻辑控制集中在控制器,控制器以标准化的接口对各种网络设备进行集中式的管理和配置。

因为网络事件有可能发生在任何一台交换机或者端主机上,所以本质上SDN网络还是分布式和异步操作的。传统网络因暂时中间态引发的一致性在集中式控制的SDN网络中仍然存在。文献[3]首先提出在SDN网络中,由于多交换机更新的非原子性,从而导致网络控制逻辑的不一致问题。文献[4-5]

从模型检查和符号执行的角度对SDN应用中存在的网络控制逻辑不一致性问题进行识别和分析。目前,针对该问题的解决方案研究主要集中在3个方面:

1) 增加交换节点的自治智能,降低控制器的负载,让控制平面响应速度更快,达到减少控制器与交换节点之间时延的目的。大多数都通过在交换节点安装缓存规则等方法实现(DIFANE系统<sup>[6]</sup>、DevoFlow设计方案<sup>[7]</sup>),但是这也违背了SDN中控制和转发分离的原则; 2) 让控制器通过状态管理模块对网络设备的暂时中间态进行管理。文献[8]通过全局状态集中管理能有效降低SDN控制平面的不一致性; 文献[9-10]针对网络控制逻辑更新一致性,提出每报文一致性和每流一致性的概念; 3) 通过优化控制器的部署位置来降低控制器与交换机的网络时延,保证网

收稿日期: 2013-08-05; 修回日期: 2014-01-14

基金项目: 铁道部重大项目(2012X004-A); 国家自然科学基金(61073177)

作者简介: 宋海权(1981-),男,博士生,主要从事形式化验证、SDN网络技术等方面的研究。

络的一致性。文献[11]提出采用平均时延和最坏时延两个指标分析Openflow控制器的位置部署问题, 利用时延作为关键约束条件优化控制器的个数和部署位置。

本文在研究SDN网络控制不一致问题的基础上, 提出一种基于平均时延的SDN控制逻辑一致性解决方案(control logic consistency method based on the average delay in SDN, CLCBAD)。该方法属于单数据包一致性的解决方案。首先, SDN控制器上发现和维持网络拓扑及相邻节点间的传输平均时延; 其次, 当数据包到达区域网络的源节点时, SDN控制器根据网络拓扑对数据包可能经过的所有路径进行计算, 找出一条同时满足逻辑控制一致和平均时延总和最短的路径; 再次, SDN控制器精确控制路径中各交换节点的控制规则的安装策略; 最后, 数据包根据控制规则在最短时间内通过该条路径。CLCBAD方法的优点在于通过网络的集中控制使得网络路由的收敛性更快, 不仅能加强网络的控制逻辑一致性, 而且能从网络中选择出平均时延最短的数据包路由方案。

### 1 SDN网络时延影响的不一致问题

Openflow<sup>[12]</sup>初步实现了SDN原型设计思想, 控制器根据需要增加、删除、修改交换节点中的流表, 从而达到控制网络的目的(如: 链路负载均衡、动态组网等)。Openflow交换节点中都有一张或者多张流表, 每张流表由包含着匹配项、计数器以及动作集合的数据项组成。交换节点依据流表数据项中的匹配项以及动作集合, 决定数据包的处理方式(转发、丢弃、上传等), 所以Openflow就是基于流表让整个网络实现控制层面和转发层面分离。由于SDN网络本质上仍然具有分布式和异步的特点, SDN控制器对各个交换节点控制规则安装不具备原子性。如图1所示, 主机1发到主机2的报文在交换机1的流表中未匹配到相应的匹配项, 则上传到控制器。控制器分别对交换机1和交换机2安装新规则, 控制数据包转发。但是由于时延的存在, 若从交换机1转发的报文在新规则安装之前到达交换机2, 交换机2仍然按照旧的流表项对报文进行处理, 结果可能会造成交换机2对报文的处理具有不确定性(如被错误丢弃或者错误转发等)。因此控制器与交换机之间的网络延迟也是影响SDN网络一致的一个重要因素。

本文基于SDN技术架构, 提出一个解决逻辑控制不一致问题的技术方案, 即在控制平面建立一个

负责处理保证SDN网络一致性的控制模块, 如图2所示。当某个应用需要对交换机转发规则更新时, 该模块负责从应用平面提取关键信息, 并结合全局网络拓扑视图等数据结构进行运算, 最后给出合理的一致性策略, 指导网络运行。下面对技术方案中一致性控制模块的控制策略进行讨论。

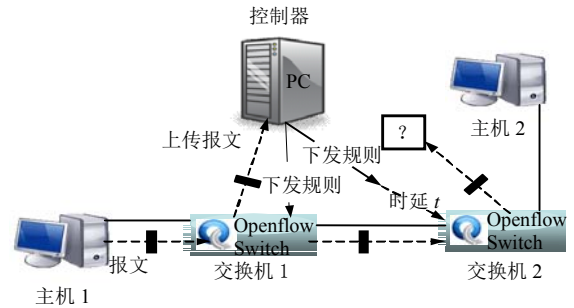


图1 基于延迟的逻辑控制一致性问题

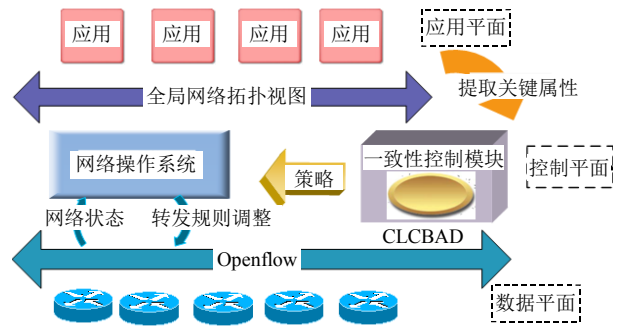


图2 基于SDN技术架构的逻辑控制一致性方案

## 2 基于时延约束的SDN路由优化方法

### 2.1 方法描述与设计

CLCBAD方法是了解决SDN网络因时延而导致的控制逻辑不一致问题。方法的推导实现过程主要有以下5个步骤: 1) 确定SDN网络拓扑及交换节点间的时延、交换节点与控制器的时延; 2) 确定源交换节点与宿交换节点之间的可达路径集; 3) 判断可达路径是否存在控制逻辑不一致问题; 4) 提出基于控制逻辑一致的最短时延路由模型; 5) 控制器规则安装策略。

### 2.2 SDN网络拓扑及时延的确定

任意给定的网络 $G(V,E)$ , 其中,  $V$ 为网络中所有交换节点的集合;  $E \subseteq V \times V$  表示交换节点连接的集合; 令 $N=|V|$ 为网络中交换机节点的个数。CLCBAD方法在控制器上动态维护区域网络的时延邻接矩阵, 如果网络拓扑更新时, 则对矩阵进行修改。在SDN的事实标准Openflow协议中, 通过controller-to-switch消息可以让交换节点从各个端口发送LLDP协议来发现网络的逻辑拓扑结构和网络

时延。

由于网络时延受网络负载、传输数据的动态特性等因素影响,对网络时延的精确测量和控制比较困难。为了尽可能的准确,随机间隔一段时间对时延进行若干次测量,最后取平均时延代表节点之间的网络时延。

$$d_{ij} = \frac{1}{n} \sum_{k=1}^n d_{ijk} \quad (1)$$

式中,  $d_{ij}$ 表示交换节点*i*与*j*之间的网络时延,如果*i*与*j*不相邻,则令 $d_{ij}=0$ ;  $n$ 表示总测量次数,  $d_{ijk}$ 表示对 $d_{ijk}$ 的第*k*次测量值,从而得到区域网络的时延邻接矩阵为:

$$S = (d_{ij})_{N \times N} \quad (2)$$

同样,控制器上也动态维护二元组集合  $sc = \{(s, c_s) | s \in V\}$ , 在二元组 $(s, c_s)$ 中,  $s$ 为交换节点编号,  $c_s$ 表示交换节点*s*到控制器的平均网络时延。

### 2.3 交换节点可达路径

SDN区域网络中通过无向图邻接矩阵的搜索算法得到两个交换节点间的可达路径集合记为 PathSet。令  $Ps_{ij} \in \text{PathSet}$ ,  $Ps_{ij} = \langle v_i, v_k, \dots, v_j \rangle$  序关系表示交换节点*i*(源交换节点)与*j*(宿交换节点)之间的一条有序路径;令  $L = |\text{PathSet}|$  表示可达路径集合元素的个数;令  $p(i, j) = d_{ij} \in S$  表示邻接矩阵中交换节点*i*与*j*间的时延。

由于在集合PathSet中,可能某些路径经过的交换节点存在逻辑控制不一致的问题,故需要对集合中的所有路径经过的交换节点进行计算和判定。

### 2.4 交换节点逻辑控制不一致性问题的判断规则

控制逻辑不一致的判断思路是遍历可达路径中的每一个交换节点,如果有一个交换节点的时延满足如下条件:控制器到该交换节点的网络时延大于控制器到路径源节点*i*的时延再加上源交换节点到这个交换节点的时延之和,则认为该路径下存在逻辑控制不一致问题。

**定义 1**  $\forall v \in ps_{ij}, \exists (s, c_s) \in sc, d_{im} \in S$ , 满足:

$$c_k > c_i + \sum_{v_i, v_m \in ps_{ij}} d_{im} \quad (3)$$

则表示该可达路径下存在因为延迟而引起的控制逻辑不一致问题,延迟为:

$$\Delta t = c_k - c_i - \sum_{v_i, v_m \in ps_{ij}} d_{im} \quad (4)$$

### 2.5 基于控制逻辑一致的路由优化

序关系  $P_{ij} = \langle v_i, v_k, \dots, v_j \rangle$  为交换节点*i*与*j*之间

的一条有序的可达路径,且令 $\Delta t_{\max}$ 表示路径 $p_{ij}$ 的所有交换节点中,由式(4)求出的最大时延。该可达路径下数据包在到达交换节点*i*并转发到*j*的网络总延迟为:

$$t = 2 * d'_i + \sum_{v_i, v_m \in ps_{ij}} d_{im} + \Delta t_{\max} \quad (5)$$

另外,若 $p_{ij}$ 没有控制逻辑不一致问题时,令 $\Delta t_{\max}=0$ 。基于网络时延的控制逻辑一致性的路由优化实际上可以归结为一个时延约束的目标规划问题,即在所有的可达路径集中寻找最优解。为了方便求解,本文将可达路径集转换为可达矩阵:如路径序关系 $\langle 5, 2, 7, 9 \rangle$ 表示从交换节点5~9的可达路径,在可达矩阵中的表示路径为 $(5, 2, 7, 9, 0, \dots, 0) \in R^{|V|}$ ,故可达路径矩阵为:

$$\text{PathMatrix} = (Ps_{ij})_{L \times N} \quad (6)$$

由前面分析可得出如下的目标规划数学模型:

$$\text{delay\_min} = \min \left( 2 * d'_i + \sum_{j=1}^N (P(x_{j+1}, x_j)) + \max(\Delta t) \right) \quad (7)$$

$$\text{s.t.} \begin{cases} \Delta t = \max(0, c_k - c_i - \sum_{m=1}^{k-1} p(x_{m+1}, x_m)) & (\text{st1}) \\ 0 < x_j < N \quad c_k, c_i \in sc & (\text{st2}) \\ \mathbf{x} \in \text{PathMatrix}, p(m, n) = d_{mn} \in S & (\text{st3}) \end{cases}$$

这个目标优化模型在控制逻辑一致性的基础上求出从交换节点*i*到*j*的最短延迟。约束(st1)判断控制器到每个交换节点是否存在控制逻辑不一致问题,如果不存在不一致问题则 $d'_k - d'_i - \sum_{m=1}^{k-1} p(x_{m+1}, x_m) < 0$ ,此时max函数使得 $\Delta t$ 取值为0。约束(st2)到约束(st3)为取值的约束。函数 $p(m, n)$ 表示在时延邻接矩阵中交换节点*m*与*n*的时延。求出的解向量 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \text{PathMatrix}$ 即为在所有的可行性解空间(可达矩阵)中寻找到的最优解。

### 2.6 控制规则安装策略

通过分析发现,控制器如果能准确控制新规则在交换节点的安装时间,就可以避免出现控制逻辑不一致的情况。如图1所示,控制器先于*t*时间下发规则到交换机2, *t*时间之后再下发规则到交换机1,这可以使交换机2对数据包的处理具有确定性。

如果路径集的最短时延路径中存在控制逻辑不一致问题,控制器需要对交换节点控制规则的安装策略进行调整,使之满足控制逻辑一致性要求。

1) 利用上述分析和规则判断出从交换节点*i*到*j*

的最短时延可达路径中所有存在控制逻辑不一致问题的交换节点, 再求出相对于控制器时钟周期延迟  $\Delta t$ , 并与交换节点  $k$  组成二元组  $(k, \Delta t)$ , 最后把所有的二元组按  $\Delta t$  从大到小顺序存入队列  $Q$  中。

2) 控制器从队列  $Q$  中取出第一个二元组  $(k, \Delta t)$ , 设置一个初始定时器  $Timer = \Delta t$ , 并让控制器对交换节点  $k$  下发控制规则, 从队列  $Q$  中删除第一个二元组元素,  $Timer$  递减1个时钟周期。

3) 当  $Timer$  的值和队列  $Q$  第一个二元组  $(k, \Delta t)$  元素中  $\Delta t$  值相同时, 让控制器对交换节点  $k$  安装控制规则, 从队列  $Q$  中删除第一个二元组元素,  $Timer$  递减1个时钟周期。重复此步骤直到  $Timer$  为0。

4) 控制器同时安装可达路径中剩余交换节点的控制规则。

5) 数据包按照交换节点上的安装规则通过SDN网络。

通过控制器时钟周期的精确控制, 基本可以避免SDN网络出现因网络延迟而引起的控制逻辑不一致情况。

### 3 仿真分析

根据上述的理论研究和分析, 建立基于网络时延的控制逻辑一致性路由最优化模型。在仿真模型中, 时延都是以毫秒为基本单位。本文的仿真实验采用文献[15]的一个网络通信随机拓扑生成模型(salama模型), 随机生成10个交换节点, 如图3所示。salama模型是生成随机网络拓扑比较常用的模型。仿真实验对随机网络拓扑中交换节点到控制器的时延参数进行了设置, 使得某些路径具备控制逻辑不一致问题的发生条件, 然后使用本文提出的CLCBAD方法进行验证, 证明其具有保证SDN网络一致性和寻找最短平均时延路由的能力。节点上的二元组表示节点序号  $i$  和控制器到该节点的时延  $C_i$ 。拓扑网中节点之间连线上的数字表示交换节点之间的时延  $d_{ij}$ 。

根据仿真模型中给出的网络拓扑和延迟, 利用本文阐述的CLCBAD方法选择出一条基于网络时延的控制逻辑一致的最优路径。1) 构造出网络模型的延迟邻接矩阵  $S$  及交换节点与控制器的延迟集合  $sc$ ; 2) 利用深度优先算法得到从任意节点间的路径集合; 3) 对每条路径所经过的交换节点是否存在控制逻辑一致性问题进行判定; 4) 求解路径集中控制逻辑一致的并且网络时延最短的路径; 5) 控制器控制对这条路径中所有交换节点控制规则的安装

策略。

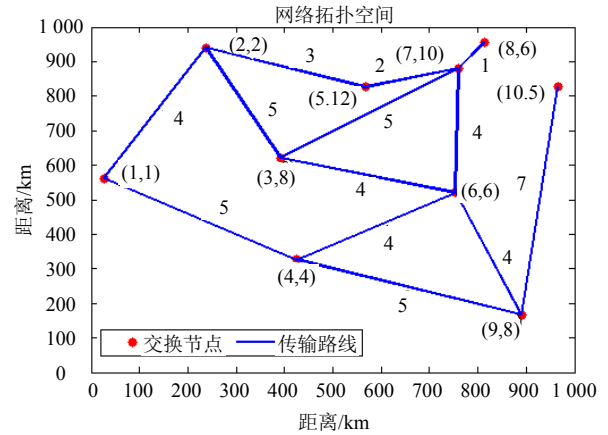


图3 SDN网络拓扑结构

仿真模型中的邻接矩阵为  $S = [(0,4,0,5,0,0,0,0,0,0,0)^T, (4,0,5,0,3,0,0,0,0,0)^T, (0,5,0,0,0,4,5,0,0,0)^T, (5,0,0,0,4,0,0,5,0)^T, (0,3,0,0,0,0,2,0,0,0)^T, (0,0,4,4,0,0,4,0,4,0)^T, (0,0,5,0,2,4,0,1,0,0)^T, (0,0,0,0,0,0,1,0,0,0)^T, (0,0,0,5,0,4,0,0,0,7)^T, (0,0,0,0,0,0,0,0,7,0)^T]$ 。控制器到交换节点的时延集合  $sc = \{(1,1), (2,2), (3,8), (4,4), (5,12), (6,6), (7,10), (8,6), (9,8), (10,5)\}$ 。

若在仿真模型中求交换节点1~7之间逻辑控制一致的路由路径, 则由邻接矩阵及深度优先算法可得出路径集合  $PathSet$ , 转化为可达路径矩阵  $PathMatrix = [(1,2,3,6,7,0,0,0,0,0)^T, (1,2,3,7,0,0,0,0,0,0)^T, (1,2,5,7,0,0,0,0,0,0)^T, (1,4,6,3,2,5,7,0,0,0)^T, (1,4,6,3,7,0,0,0,0,0)^T, (1,4,6,7,0,0,0,0,0,0)^T, (1,4,9,6,3,2,5,7,0,0)^T, (1,4,9,6,3,7,0,0,0,0)^T, (1,4,9,6,7,0,0,0,0,0)^T]$ 。

该仿真模型的最优路由路径就转换成在可行性域  $PathMatrix$  中寻找最优解。目标优化数学模型的求解过程为: 依次根据从  $PathMatrix$  中取出行向量作为目标优化的可行解, 求出控制逻辑一致的从交换节点1~7在网络时延; 然后找出最短时延所对应的行向量即为最优可行解。如图4所示,  $PathMatrix$  中每行向量的序号分别对应路径集中的  $PathID$ 。对可达路径矩阵中的路径分别运用2.4节中介绍的规则进行判定, 并计算出各条路径在控制逻辑一致情况下的网络时延。图4中表示了路径集中9条路径的策略调整前后的网络延迟变化, 若两者不一致则表明该路径在调整前存在逻辑不一致问题。从图4中得知: 路径  $2(1,2,5,7,0,0,0,0,0,0)^T$  中存在控制逻辑不一致的情况。在调整前节点间的网络延迟为11ms, 经过控制规则安装策略调整之后延迟变成17ms。同时也可以看出, 在控制逻辑一致性的条件下, 交换节点1和7之间的最短延迟路径为路径  $5(1,4,6,7,0,0,0,0,0,0)^T$ , 而不是调整之前的路径



$2(1,2,5,7,0,0,0,0,0)^T$ 。

路径2因存在控制逻辑不一致性问题,所以在计算路径2的延迟时必须加上调整的时钟。根据CLCBAD方法,若路径2仍然是路径集中的最短延迟路径,则需要控制器对路径2中交换节点的控制规则

的安装策略进行调整来保证控制逻辑一致性。根据2.6节的步骤,可得队列 $Q$ 中的元素一次为2元组: $(6,5), (0,1), (0,2), (0,7)$ 。控制规则的安装顺序为:1) 安装交换节点5的控制规则;2) 6 ms之后同时安装剩余交换节点1,2,7的控制规则。

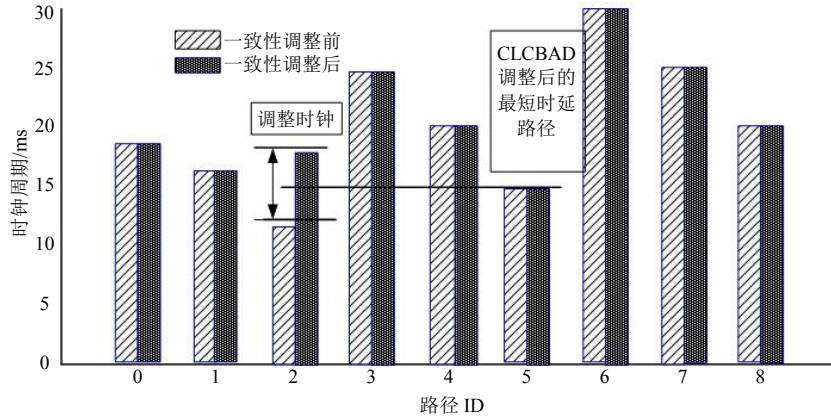


图4 节点1到7路径集一致性调整对比图

本文在mininet实验平台上使用CLCBAD方法和其他通用路由协议就网络控制逻辑一致性进行了对比分析。OSPF是一种典型的链路状态路由协议,路由器通过交换并保存链路信息,从而掌握全局拓扑结构,独立计算路由。OSPF作为一种内部网关协议(interior gateway protocol, IGP),用于在同一个自治域(AS)中的路由器之间发布路由信息。区别于距离矢量协议(RIP),OSPF具有支持大型网络、路由收敛快、占用网络资源少等优点,在目前应用的路由协议中占有相当重要的地位。故本文采用OSPF协议作为和CLCBAD方法对比的协议。在计算路由时采用网络时延代表链路状态。对比实验步骤如下:1) 按照图3所示的网络拓扑结构,在mininet中设置了10个主机节点分别与10个交换节点相连接,并设置相关时延参数;2) 每次随机选择两个主机进行通信;每次测试时清空所有交换节点的流表;3) 其他交换节点转发到的数据包,若在流表中找不到相应的匹配项则选择丢弃或者随机转发到其他端口(模拟控制逻辑不一致);4) 图2中的一致性控制模块分别基于最短路径优先(OSPF)路由协议及CLCBAD方式对交换节点进行路径选择;5) 利用SDN控制器对交换节点的信息进行统计。表1列出了100 000次随机选取通信主机进行UDP通信的情况下,OSPF协议和CLCBAD方法分别出现逻辑控制不一致(丢弃、错误转发)的统计数据。表1的统计信息表明,OSPF只是数据链路状态的路由协议,其计算出的路由路径只是选取时延最短的路由路径,未考虑控制器对控制

规则下发的时延,故可能就存在着控制逻辑不一致的问题,如交换节点3、5、7。在CLCBAD方法和OSPF协议的时间比较方面:本文随机1 000次选择两个交换节点分别按照两种方式选择路径进行通信,发现CLCBAD方法在耗时上较OSPF略高6.52%(主要耗时包括控制器端有一致性判断耗时及一致性策略调整耗时)。故CLCBAD方法是重点在保证控制一致的基础上,再选取最短时延的路由路径。实验也验证了它能很好解决控制逻辑不一致的问题。

表1 OSPF与CLCBAD统计信息比较

交换节点	OSPF		CLCBAD	
	丢弃/%	错误转发/%	丢弃/%	错误转发/%
1	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00
3	1.47	1.06	0.00	0.00
4	0.00	0.00	0.00	0.00
5	7.20	5.41	0.00	0.00
6	0.00	0.00	0.00	0.00
7	5.56	3.20	0.00	0.00
8	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00

## 4 结论

SDN网络中,控制器是整个网络的核心,也是实现网络精细化控制的关键。本文提出的CLCBAD方法着重解决SDN网络中因网络时延而导致控制逻辑不一致的问题。CLCBAD方法的中心思想是首先对网络节点间、节点与SDN控制器之间的网络时延进行平均测量,然后通过改变下发控制规则的次序

和时间而使SDN网络到达逻辑控制一致。仿真实验表明, CLCBAD方法在保证SDN网络逻辑一致的基础上, 能够选出交换节点间的网络时延最短的最优路径。

### 参 考 文 献

- [1] MARTIN C, MICHAEL J, SCOTT S. Ethane: Taking control of the enterprise[C]//SIGCOMM'07. Kyoto, Japan: [s.n.], 2007.
- [2] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: Enabling innovation in campus networks[C]//ACM SIGCOMM Computer Communication Review. Seattle: [s.n.], 2008, 38(2): 69-74.
- [3] CANINI M, KOSTIĆ D, REXFORD J, et al. Automating the testing of OpenFlow applications[C]//The 1st International Workshop on Rigorous Protocol Engineering (WRiPE). [S.l.]: [s.n.], 2011.
- [4] CANINI M, VENZANO D, PERESINI P, et al. A NICE way to test Openflow applications[C]//Proc of the 9th USENIX Symp on Networked Systems Design and Implementation (NSDI). San Jose: USENIX Association, 2012.
- [5] 左青云, 陈鸣, 赵广松, 等. 基于 Openflow 的 SDN 技术[J]. 软件学报, 2013, 24(5): 1078-1097.  
ZUO Qing-yun, CHEN Ming, ZHAO Guang-song, et al. Openflow-based SDN technologies[J]. Journal of Software, 2013, 24(5): 1078-1097.
- [6] MINLAN Y, MICHAEL J, JIA Wang. Scalable flow-based networking with DIFANE[C]//SIGCOMM'10. New Delhi, India: [s.n.], 2010.
- [7] CURTIS A R, MOGUL J C, TOURRILHES J, et al. DevoFlow: Scaling flow management for high-performance networks[C]//Proc of the SIGCOMM. Toronto: ACM Press, 2011.
- [8] DAN L, ANDREAS W, BRANDON H. Logically centralized? State distribution trade-offs in software defined networks[C]//HotSDN'12. Helsinki, Finland: [s.n.], 2012.
- [9] MARK R, NATE F, JENNIFER R. Consistent updates for software-defined networks: Change you can believe in[C]//Hotnets' 11. Cambridge, MA, USA: [s.n.], 2011.
- [10] MARK R, NATE F, JENNIFER R. Abstractions for network update[C]//SIGCOMM'12. Helsinki, Finland: [s.n.], 2012.
- [11] HELLER B, SHERWOOD R, MCKEOWN N. The controller placement problem[C]//Proc of the SIGCOMM 2012 Workshop on Hot Topics in Software Defined Networking (HotSDN). Helsinki: ACM Press, 2012.
- [12] Open Networking Foundation(ONF). Openflow switch specification, version 1.1.0[EB/OL]. [2013-8-25]. <http://archive.openflow.org/wp/documents/>.
- [13] SALAMA H F. Multicast routing for real-time communication on high-speed networks[D]. Raleigh: North Carolina State University, 1996.

编辑 蒋晓