

## 软件定义光网络中一种时延约束的控制器生存性部署方法

曾帅\* 盖绍聪 张毅 赵国锋 左理政

(重庆邮电大学未来网络研究中心 重庆 400065)

**摘要:** 控制时延、控制平面的生存性和控制平面的控制冗余程度是软件定义光网络中网络性能是否良好的重要判断依据。该文提出时延约束下的控制器生存性部署方法,该方法充分考虑时延、生存性和控制器冗余等网络性能因素,在用户指定时延的前提下,确保每个网络节点至少有两条控制链路,以提高控制平面的生存性。同时,保证使用尽可能少的部署节点完成整个网络的覆盖,以减少控制平面的控制冗余。仿真表明,该方法能够有效地减少控制时延,提高控制平面的生存性,并减少控制器的部署个数,降低控制冗余,有效地提高了软件定义光网络的整体网络性能。该方法保证至少两条控制链路与 C-MPC 算法起到了相同的保护作用,与 MCC 算法相比,使 SDON 网络控制平面可靠性提高了 20%。同时,在指定时延 10 ms 的约束条件下,在 NSF 和 COST239 网络中,与 C-MPC 算法相比所提算法分别减少了 88%和 75%的控制器部署。

**关键词:** 软件定义光网络;生存性;指定时延;控制冗余;控制平面

中图分类号: TN915.07

文献标识码: A

文章编号: 1009-5896(2017)07-1727-08

DOI: 10.11999/JEIT160820

## Survivability Deployment Method for Controller with Time-delay Constraint in Software Defined Optical Network

ZENG Shuai GAI Shaocong ZHANG Yi ZHAO Guofeng ZUO Lizheng

(Future Network Research Center, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

**Abstract:** Control delay, survivability of control plane and control redundancy of control plane are important criteria to judge whether the network performance is good in Software Defined Optical Network (SDON). A method that the survivability deployment for controller with time-delay constraint is put forward. This method takes full account of the network performance factors, such as time delay, survivability and controller redundancy. In the premise of a user specified delay, in order to improve the control plane survivability, it is insured that each network node has at least two control links. At the same time, the node deployments are kept to the minimum when complete of the entire network coverage, to reduce the redundancy of control plane. Simulation results show that the method can effectively reduce the control delay, improve the existence of control plane, and reduce the number of controller, decline the control redundancy, effectively improve the overall network performance of software defined optical network. The method ensures at least two control links have the same protection as C-MPC algorithm. Compared with the MCC algorithm, the reliability of SDON control plane is improved by 20%. Meanwhile, under the constraint conditions of 10 ms time delay, compared with C-MPC algorithm, the proposed algorithm can reduce the number of controller deployment by 88% and 75% in NSF and COST239 network.

**Key words:** Software Defined Optical Network (SDON); Survivability; Specified time delay; Control redundancy; Control plane

### 1 引言

计算机网络在当今社会中起着越来越重要的作

用,但是传统网络由于带宽容量、能源消耗和传输范围等技术的瓶颈不能满足现有网络应用的快速发展需求<sup>[1]</sup>。软件定义网络(Software Defined Networking, SDN)作为一种全新的网络架构实现了网络控制平面与数据平面分离,实现网络的统一灵活管控<sup>[2,3]</sup>。网络用户可以方便地对网络进行编程和设计,完成沿路径转发设备的统一部署,从而满足不同用户的任务需求<sup>[4]</sup>。软件定义光网络(Software Defined Optical Network, SDON)是对 SDN 技术的具体应用,是基于通用多协议标签交换(Generalized MultiProtocol Label Switching, GMPLS)和基于路径计算单元(Path Computation Element, PCE)光

收稿日期: 2016-08-03; 改回日期: 2017-04-27; 网络出版: 2017-05-18

\*通信作者: 曾帅 zengshuai@cqupt.edu.cn

基金项目: 重庆市教委科学技术研究项目(KJ1500429), 重庆市科委基础科学与前沿技术研究项目(cstc2017jcyjA0976, cstc2016jcyjA0560), 国家自然科学基金(6170011898, 61671092)

Foundation Items: The Scientific and Technological Research Program of Chongqing Municipal Education Commission (KJ1500429), The Scientific and Technological Research Program of Chongqing Science and Technology Commission (cstc2017jcyjA0976, cstc2016jcyjA0560), The National Natural Science Foundation of China (6170011898, 61671092)

网络之后的第3代智能光网络<sup>[5]</sup>。SDON通过专注于转发的哑(dumb)的转发设备和专注于控制的软件控制的驱动器实现大规模光网络海量业务的灵活调度<sup>[6]</sup>。SDON网络状态信息的收集,数据传输路由决策,光通路动态建立决策和拆除,控制流表下发决策等通过控制平面实现对数据平面的控制<sup>[6,7]</sup>。控制平面与数据平面的可靠连接关系到数据平面的转发能力,一旦控制平面与数据平面失联,数据平面将会大范围地失去数据转发能力。因此,控制器的部署位置对控制平面的生存性有着重要的作用。

目前,已经存在很多关于控制器部署方法的研究。其中,文献[6]提出了基于最小点覆盖的控制平面跨层生存性控制器部署模型,该文献中采用最小点覆盖的原理,保证在节点度数不小于2的情况下每个节点至少有两个控制器与之相连。该部署方法很大程度地提升了控制平面的生存性,但是控制器部署节点相对较多。文献[8]提出了两个控制器部署问题:控制器部署个数、控制器部署位置,并通过256种网络拓扑控制平面进行分析,发现控制器只要部署位置合理,一个控制器就能够满足对控制时延的要求,该文献考虑了传输时延对控制器部署的影响,没有考虑控制平面生存性的问题。文献[9]考虑传输时延和发送时延,完善了平均时延/最大时延最小化模型,提出了考虑发送时延的两种部署算法。文献[10]提出了一种数学模型,该模型将控制器分为两类:一个是本地控制器(local controller),一个是根控制器(root controller),该文献中还提出一种控制器交互模型,网络设计人员利用这种数学模型实现控制器时延和缓存的计算。文献[11]提出了SVVR(SurViVoR)算法来确定控制器的部署方案,该算法使得网络节点与控制器之间满足链路分离的路径数的最大化,但是没有考虑控制链路的长度,造成较大的控制时延。文献[12]提出了一种多域软件定义网络架构(architecture for multi-domain software defined networks),这种架构将控制平面垂直地划分为两层,其中底层包含本地控制器,只负责本地信息的管理;上层控制器与底层控制器进行通信,协同本地控制器之间的工作。文献[12]中的部署方法提高了控制平面的兼容性,但是未对控制平面的时延和生存性作过多的考虑。文献[13]提出了基于最小割控制器分簇部署算法(Min Cut-Centroid, MCC),但是忽略了跨簇的控制平面生存性。文献[14]定义了控制器间的管控优先级,该方法有效地减少了网络节点对高控制中心的依赖,有效地提高了网络性能,但是该方法只适用于静态网络,面对复杂多变的网络效果不佳。文献[15]在一个给定所有通信

节点负载的网络中提出了一种算法,确定单一控制器的部署。该算法通过获取网络节点间的负载决定控制器的部署位置,从而减轻整体的网络复杂程度。该算法解决了网络整体的负载性能,但是未充分考虑SDN网络的控制时延、生存性等多方面网络性能因素。

在用户所允许最大控制时延的前提下,为了提高SDON控制平面生存性,并尽可能减少控制器的部署个数,减少控制平面的控制冗余,本文提出软件定义光网络中一种时延约束的控制器生存性部署方法。该方法保证控制器到网络节点间至少保持两条控制链路,有效地提高了控制平面的生存性,并根据控制时延为控制链路设定传输优先级,在保证生存性的同时减小了传输时延;此外,该方法根据控制器的部署个数确定部署模型,采用控制器覆盖范围并集的方式来实现对网络的整体覆盖,有效地减少了控制平面的控制冗余。

## 2 指定时延生存性部署模型

为了提高SDON控制平面的生存性和满足用户的控制时延需求,并且尽可能减少控制冗余,根据实际网络需求部署一个或多个控制器。根据控制器部署个数,部署模型分为单一控制器部署模型和多控制器部署模型。不同部署模型的划分是由于在SDON网络中部署多个控制器时,需找出合适的部署位置作为管控中心的部署位置,以协调控制器间的控制工作。算法至少为每个控制器到网络节点部署至少两条控制链路,即SDON网络拓扑每个节点度数至少为2,以提高控制平面的生存性。

(1)单一控制器部署模型: 根据实际网络传输需求,将一个控制器部署在网络节点中的某个合理位置可以满足传输时延需求,并保证目标节点到控制器间至少有两条控制链路。该模型根据指定时延获取控制器到每个节点的传输链路,根据控制链路时延大小,为每条链路设定不同的优先级。如图1(a)所示为单一控制器部署模型举例,图1(a)中虚线表示控制器满足时延和生存性条件的管控范围。在该管控模型下,SDON网络中部署一个控制器即可满足用户指定时延 $T_{cs}$ 需求。

(2)多控制器部署模型: 根据实际网络传输需求,一个控制器在控制实时延的约束下不能将网络拓扑中的节点全部覆盖(满足用户指定时延和每个节点至少有两条控制链路),因此需要部署多个控制器,多个控制器管控节点的并集可覆盖整个拓扑网络。为实现控制器间的协同管控,需要在多个控制器间选出管控中心。如图1(b)所示为多控制器部署

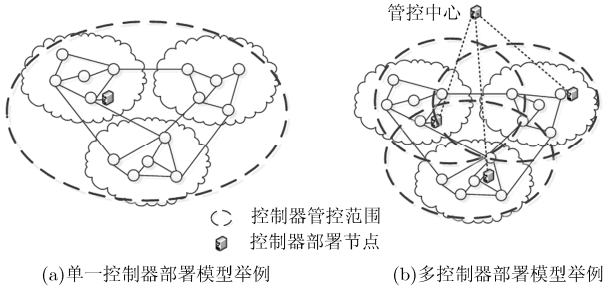


图 1 控制器部署模型

模型举例，图中虚线是控制器满足时延和生存性条件的管控范围，图 1(b)中 3 个控制器管控范围的并集完成整个 SDON 网络的覆盖。

### 3 指定时延的生存性部署算法

由前文调研分析可得，控制器的部署位置、部署个数直接关系 SDON 网络的整体性能。其中，若部署位置不当，SDON 网络中部分节点获得控制指令的时间就会过长；如果部署控制器的个数过多，就会加大控制器平面的控制冗余，反而降低控制平面的控制性能。本文提出一种用户指定时延约束下的控制器部署方法，在满足用户控制时延的需求之下，提高控制平面的生存性，减少控制器的部署个数。

本文提出基于指定时延的生存性部署方法。设 SDON 网络拓扑为  $G = (V, E)$ ，其中  $V$  代表网络节点集， $E$  代表网络中的链路集。部署节点的控制器用  $v_{pc}$  表示，控制器管控中心用  $v_{cc}$  表示，多控制器部署情况下管控中心部署节点用  $v_{cc}$  表示，控制器控制指令到第  $i$  个节点的传输时延用  $T_{pci}$  表示，SDON 网络所允许的最大控制时延用  $T_{cs}$ 。以上数学符号及意义参见表 1。

#### 3.1 部署目标

为了满足 SDON 网络指定控制时延，保证控制

表 1 数学符号及其意义

符号	含义
$v_i$	网络节点中的第 $i$ 个节点
$v_{pc}$	控制器部署节点
$v_{cc}$	多控制器部署情况下管控中心部署节点
$T_{pci}$	控制器控制指令到第 $i$ 个节点的传输时延
$T_{cs}$	SDON 网络所允许的最大控制时延
$NUM_{v_n(T_{n1}^1, T_{n1}^2, \dots, T_{n1}^k)}$	部署节点 $v_n$ 到节点 $v_i$ 时延小于约束时延 $T_{cs}$ 的数量
$ v_{i1} \cup v_{i2} \dots \cup v_{in} $	部署节点覆盖整个网络所需控制器个数
$T(v_{pci}, v_{pcj})$	多控制器部署条件下控制部署节点 $v_{pci}$ 到部署节点 $v_{pcj}$ 的传输时延

器到指定网络节点有尽可能多的控制通道，并且保证 SDON 网络中的控制器尽可能少。本文在满足用户指定时延  $T_{cs}$  下，保证每一个网络节点至少有 2 条控制链路，即  $NUM_{v_n(T_{n1}^1, T_{n1}^2, \dots, T_{n1}^k)} \geq 2$ ，本文算法约束条件具体如表 1 及式(1)-式(3)所示。

$$\max \left[ \text{avg} \sum_1^k NUM_{v_{i1}(T_{i11}^1, T_{i11}^2, \dots, T_{i11}^k)}, \dots, \text{avg} \sum_1^k NUM_{v_{iz}(T_{iz1}^1, T_{iz1}^2, \dots, T_{iz1}^k)} \right] \quad (1)$$

$$\min(|v_{i1} \cup v_{i2} \dots \cup v_{in}|) \quad (2)$$

$$\min \left\{ \text{avg} \left[ \sum_{i=1, j=1}^{j=n} T(v_{pci}, v_{pcj}) \right], \text{avg} \left[ \sum_{i=2, j=1}^{j=n} T(v_{pci}, v_{pcj}) \right], \dots, \text{avg} \left[ \sum_{i=n, j=1}^{j=n} T(v_{pci}, v_{pcj}) \right] \right\} \quad (3)$$

约束条件说明：式(1)表示若将控制器部署在  $v_{i1} \sim v_{iz}$  处，控制链路条数的最大平均值；式(2)表示多控制器部署条件下，部署节点的并集将整个 SDON 网络覆盖所需控制器个数的最小值。式(3)中表示多控制器部署条件下，部署节点间平均交互时延的最小值。本文算法为提高控制平面生存性，保证控制器到网络节点至少有 2 条控制链路，控制链路的条数将决定控制平面的生存性，因此式(1)就是为了保证控制链路条数最大化，以最大程度地提高控制平面生存性；此外，控制器冗余也是制约网络性能的重要因素，本文旨在减少控制平面的控制冗余，以尽可能减少控制器部署个数，式(2)是为了保证网络节点包含在控制器的管控范围内，且所需控制器的个数最少。

#### 3.2 部署步骤

指定时延生存性部署算法为离线部署算法，控制器的部署将事先根据本文算法确定部署位置。此外，部署控制器中预留控制链路由表，根据控制链路的联通性、控制链路的优先级动态地指定控制指令的传输路由。

步骤 1 求  $v_1$  到  $v_i$  的所有无环路径。采用回溯法/穷举法寻找  $v_1$  到  $v_i$  的所有路径，如果边有任意一条重复，记为同一条链路，并记录每条链路的传输时延，并根据传输时延为控制链路制定优先级(注：重复边的路径只记录时延最短的路径)。将结果记录为：

$v_1(T_{11}^1, T_{11}^2, \dots, T_{11}^k)$ ，其中  $T_{11}^k$  表示  $v_1$  到  $v_i$  的第  $k$  条路径的传输时延。

步骤2 遍历所有的目标节点  $v_i (1 < i \leq n)$ 。

采用步骤1中的方法和记录方式, 遍历所有的目标节点  $v_1$  到  $v_2$ ,  $v_1$  到  $v_3, \dots, v_1$  到  $v_n$ 。并依次记录结果:  $v_1(T_{12}^1, T_{12}^2, \dots, T_{12}^k), v_1(T_{13}^1, T_{13}^2, \dots, T_{13}^k), \dots, v_1(T_{1i}^1, T_{1i}^2, \dots, T_{1i}^k), \dots, v_1(T_{1n}^1, T_{1n}^2, \dots, T_{1n}^k)$ 。

步骤3 遍历所有的起始节点  $v_j (1 < j \leq n)$ 。

采用步骤1和步骤2中的方法和记录方式, 依次记录起始节点到SDON网络中所有目标节点的链路和对应的链路传输时延。并记录结果如下:

- (1)  $v_1(T_{12}^1, T_{12}^2, \dots, T_{12}^k), v_1(T_{13}^1, T_{13}^2, \dots, T_{13}^k), \dots, v_1(T_{1i}^1, T_{1i}^2, \dots, T_{1i}^k), \dots, v_1(T_{1n}^1, T_{1n}^2, \dots, T_{1n}^k)$ ;
- (2)  $v_2(T_{21}^1, T_{21}^2, \dots, T_{21}^k), v_2(T_{23}^1, T_{23}^2, \dots, T_{23}^k), \dots, v_2(T_{2i}^1, T_{2i}^2, \dots, T_{2i}^k), \dots, v_2(T_{2n}^1, T_{2n}^2, \dots, T_{2n}^k)$ ;
- (3)  $v_3(T_{31}^1, T_{31}^2, \dots, T_{31}^k), v_3(T_{32}^1, T_{32}^2, \dots, T_{32}^k), \dots, v_3(T_{3i}^1, T_{3i}^2, \dots, T_{3i}^k), \dots, v_3(T_{3n}^1, T_{3n}^2, \dots, T_{3n}^k)$ ;
- (n)  $v_n(T_{n1}^1, T_{n1}^2, \dots, T_{n1}^k), v_n(T_{n2}^1, T_{n2}^2, \dots, T_{n2}^k), \dots, v_n(T_{ni}^1, T_{ni}^2, \dots, T_{ni}^k), \dots, v_n(T_{n(n-1)}^1, T_{n(n-1)}^2, \dots, T_{n(n-1)}^k)$ 。

从遍历结果可以看出, 第  $i$  个节点作为起始节点时, 第  $i$  到前  $(i-1)$  个节点无需遍历(注: 上述结果只是为记录遍历结果, 代码实现过程中不对第  $i$  到前  $(i-1)$  个节点进行遍历)。

步骤4 给定控制时延约束参数以此确定满足条件的控制链路数量。指定时延约束参数由SDON网络的自身特性以及用户的网络体验确定。假设根据实际需求给出时延参数为  $T_{cs}$ , 因此保证每条链路的控制实验小于  $T_{cs}$ 。

根据步骤3中确定的每条链路的时延记录结果, 依次与  $T_{cs}$  进行比较, 根据比较结果记录起始节点所用于满足条件的控制链路数。记录结果如下:

- (1)  $NUM_{v_1(T_{12}^1, T_{12}^2, \dots, T_{12}^k)}, NUM_{v_1(T_{13}^1, T_{13}^2, \dots, T_{13}^k)}, \dots, NUM_{v_1(T_{1i}^1, T_{1i}^2, \dots, T_{1i}^k)}, NUM_{v_1(T_{1n}^1, T_{1n}^2, \dots, T_{1n}^k)}$ ;
- (2)  $NUM_{v_2(T_{21}^1, T_{21}^2, \dots, T_{21}^k)}, NUM_{v_2(T_{23}^1, T_{23}^2, \dots, T_{23}^k)}, \dots, NUM_{v_2(T_{2i}^1, T_{2i}^2, \dots, T_{2i}^k)}, NUM_{v_2(T_{2n}^1, T_{2n}^2, \dots, T_{2n}^k)}$ ;
- (n)  $NUM_{v_n(T_{n1}^1, T_{n1}^2, \dots, T_{n1}^k)}, NUM_{v_n(T_{n2}^1, T_{n2}^2, \dots, T_{n2}^k)}, \dots, NUM_{v_n(T_{ni}^1, T_{ni}^2, \dots, T_{ni}^k)}, \dots, NUM_{v_n(T_{n(n-1)}^1, T_{n(n-1)}^2, \dots, T_{n(n-1)}^k)}$ 。

注:  $NUM_{v_n(T_{ni}^1, T_{ni}^2, \dots, T_{ni}^k)}$  表示起始节点为  $v_n$  时,  $v_n$  到其它节点时延小于指定时延  $T_{cs}$  的数量。

步骤5 比较判断  $NUM_{v_n(T_{ni}^1, T_{ni}^2, \dots, T_{ni}^k)}$ 。为保证控制平面的生存性需保证  $NUM_{v_n(T_{ni}^1, T_{ni}^2, \dots, T_{ni}^k)} \geq 2$ ,

$NUM_{v_n(T_{n2}^1, T_{n2}^2, \dots, T_{n2}^k)} \geq 2, \dots, NUM_{v_n(T_{ni}^1, T_{ni}^2, \dots, T_{ni}^k)} \geq 2$ , 即起始节点到目标节点至少应该有两条控制通道。

(1) 如果只存一个  $v_i$  (当  $i = n$  时) 满足上述条件, 则节点  $v_i$  就为控制器  $v_{pc}$  部署节点。

(2) 如果存在多个节点  $v_{i1}, v_{i2}, \dots, v_{iz}$  满足上述条件时, 则此时求所有控制链路数目的平均值, 以保证SDON网络中每个节点的平均生存性最大。

$$\max \left[ \text{avg} \sum_1^k NUM_{v_{i1}(T_{i11}^1, T_{i11}^2, \dots, T_{i11}^k)}, \dots, \text{avg} \sum_1^k NUM_{v_{iz}(T_{iz1}^1, T_{iz1}^2, \dots, T_{iz1}^k)} \right]$$

满足上述公式中的最大控制链路条数的节点作为控制器  $v_{pc}$  部署节点。

(3) 如果不存在节点  $v_i$  (当  $v_i = v_n$  时) 保证

$NUM_{v_n(T_{n1}^1, T_{n1}^2, \dots, T_{n1}^k)} \geq 2, NUM_{v_n(T_{n2}^1, T_{n2}^2, \dots, T_{n2}^k)} \geq 2, \dots, NUM_{v_n(T_{ni}^1, T_{ni}^2, \dots, T_{ni}^k)} \geq 2$ , 但是  $v_i$  节点作为起始节点能够

满足部分节点所述条件, 设  $v_{i1}, v_{i2}, \dots, v_{im}$  中每个节点都能满足部分节点的生存性控制链路数目需求, 即单个节点  $v_{im}$  不能将整个SDON网络节点全部覆盖, 因此需求多个节点  $v_i$  部署控制器, 共同完成SDON网络中节点的覆盖。同时, 为减少SDON网络中控制平面的冗余, 应使得最少的节点完成SDON网络节点的覆盖, 使得  $|v_{i1} \cup v_{i2} \dots \cup v_{im}|$  数值最小, 即  $\min(|v_{i1} \cup v_{i2} \dots \cup v_{im}|)$ 。

以上步骤1至步骤5即可完成控制器部署位置的选取, 其中上述步骤5(1)和步骤5(2)的部署位置节点对应单一控制器部署模型, 步骤5(3)中的部署节点对应多控制器部署模型。

步骤6 管控中心  $v_{cc}$  节点位置确定。对于多控制器的管控平台, 多个控制器之间需要协调工作, 因此本文所有控制器部署中需要确定一个管控中心, 以协调控制器之间的工作。本文部署算法考虑控制器之间的协调信令传输时延, 设控制器部署节点位置为  $v_{pc1}, v_{pc2}, \dots, v_{pcn}$ 。  $v_{cc}$  节点的部署位置为到其控制器部署节点平均交互时延最小的节点, 即所求节点满足

$$\min \left\{ \text{avg} \left[ \sum_{i=1, j=1}^{j=n} T(v_{pci}, v_{pcj}) \right], \text{avg} \left[ \sum_{i=2, j=1}^{j=n} T(v_{pci}, v_{pcj}) \right], \dots, \text{avg} \left[ \sum_{i=n, j=1}^{j=n} T(v_{pci}, v_{pcj}) \right] \right\}$$

### 3.3 控制指令传输决策

根据3.2节中基于指定时延的控制器生存性部

署方法，可以计算出控制器的部署位置。控制器的部署位置确定后，如何有效地实现控制器对网络节点的管控是待解决的另一问题。文献[6]根据基于最小点覆盖的部署模型，设计了 3 级管控下的传输策略；文献[9]以控制指令传输时延作为控制器部署条件。本文针对所提部署算法，提出本文的控制指令传输策略。本文部署算法在计算起始节点到目标节点传输路由时，根据传输时延为每条控制链路指定优先级，并生成控制链路表供控制器查询使用。在控制器下发控制指令时，控制器将根据控制链路的优先级决定由哪条链路传输控制指令，若当前最大优先级的控制链路发生故障，优先级次之的链路作为当前控制器的指令传输链路。该策略充分利用部署算法计算结果，一方面，提高部署算法结果的利用率；另一方面，控制器下发控制指令时，只需对控制链路表做查询操作无需重新计算控制链路，减少指令传输决策带来的额外的开销。

### 3.4 指定时延约束的生存性部署算法复杂性分析和优势分析

(1)算法复杂性分析：同文献[6]所提出的基于最小点覆盖的部署算法一样，本文部署算法为离线部署算法，在 SDN 控制器接入 SDON 网络之前，用户根据拓扑结构与节点间的链路信息，依据本文算法计算部署节点位置，并将 SDN 控制器部署于所计算位置。本文算法需遍历单个源节点到其余所有目标节点的路径，并依次遍历所有源节点，复杂度为  $O(n^3)$ ；记录拓扑路径并对其传输时延标识后，算法需对每个节点的遍历结果做满足指定时延约束的比较，复杂度为  $O(n^2)$ 。所提算法通过步骤拆分，分担各节点计算量，且各节点间并行计算提高算法运行效率。虽然总体复杂度相对较高，但指定时延约束的生存性部署算法采用离线部署的方式，有效弥补了算法中可能存在的复杂度较高的不足。该部署算法在网络建设前期一次性静态完成，不占用网络实际计算资源和运行时间，不降低网络部署和建成后的实时运行效率和整体传输性能。

#### (2)指定时延约束的生存性部署算法优势分析：

本文算法以用户需求出发，结合网络拓扑结构，在满足用户需求前提下完成 SDN 控制器生存性的部署目标。所提算法具有以下优势：(1)可根据 SDON 网络的实际需求和用户的网络要求指定传输时延，能够更好地满足 SDON 网络中的控制平面对数据平面的管控；(2)在满足用户指定控制时延的基础之上，本文所提控制器部署算法理论上能够使得控制时延 100%满足用户需求；(3)控制器的部署位置到网络节点，根据控制时延指定至少 2 条的控制

链路，因此大大提高了控制平面的生存性；(4)本文采用控制器管控范围并集的方式实现 SDON 网络中网络节点的有效覆盖，减少了控制器的部署个数，从而减少控制平面的控制冗余。

综上所述，本文算法从控制时延、生存性和控制平面管控复杂度 3 个方面综合考虑控制器的部署问题，从而有效地提高网络的整体性能。本文仿真实验部分将对所提部署算法的优势进行仿真验证。

## 4 仿真实验与结果分析

本文将使用 NSF<sup>[16]</sup>和 COST239 拓扑结构作为仿真测试环境，与文献[6]所提的 C-MPC 算法、文献[11]所提的 SVVR 算法、文献[13]所提的 MCC 算法进行仿真对比。对于网络的整体性能，生存性和时延是两个重要的参数指标。本文算法在用户指定时延的条件之下提高控制平面的生存性。本文将采用与文献[6]相同的验证指标，具体测试指标定义如下：

**定义 1** 故障出现概率  $P_1$  为控制信道工作路由发生故障的概率，如式(4)所示。

$$P_1 = \text{ave}_{i,j} \left[ 1 - (1-p)^{W(s_i, c_j)} \right] \quad (4)$$

**定义 2** 故障恢复概率  $P_2$  为依靠控制信道保护路由使控制信道快速恢复的概率，如式(5)所示。

$$P_2 = \text{ave}_{i,j} \left\{ 1 - \left[ 1 - (1-p)^{W(s_i, c_j)} \right] \cdot \left[ 1 - (1-p)^{B(s_i, c_j)} \right] \right\} \quad (5)$$

其中， $p$  表示光纤百公里故障概率； $s_i$  表示第  $i$  个 OpenFlow 交换机节点； $c_j$  表示第  $j$  个控制器节点； $W(s_i, c_j)$  表示控制信道工作路由的长度， $B(s_i, c_j)$  表示保护路由的长度； $\text{ave}(\bullet)$  为取均值。

如图 2 所示为 4 种部署算法模型在 NSF 网络下故障出现概率随光纤百里故障概率的变化趋势图。从图中可以看出，在 NSF 网络与 COST 网络中，4 种管控模型部署算法故障出现概率具有相同的走势。本文所提指定时延管控模型告警出现概率比 MCC 管控模型有明显改善，与 SVVR 管控模型故障出现概率基本持平。这是由于指定时延管控部署模型根据控制器部署个数，将部署模型分为单控制器部署模型和多控制器部署模型，该算法根据指定时延优先级指定控制路由，有效地减少了控制路由的长度，因此，比 MCC 模型有明显的改善。本文部署模型在多控制器部署模型下与 SVVR 所提管控模型同属于 2 级管控模型，二者的故障出现概率基本持平。从图中可以看出 C-MPC 部署模型要优于本文部署模型，且在 COST239 网络中优势较大，这是由于 C-MPC 采用了 3 级管控模型，在 SDON

网络中投入了大量的 SDN 控制器，且 COST239 网络中连通性大，网络节点将与更多的 SDN 控制器相连接，在两种网络中比本文管控模型小幅提升了 9% 和 13% 左右，但是给控制平面带来了大量的控制冗余。

如图 3 所示为在 SDON 网络出现故障后，通过控制平面的路由决策来实现网络中的故障恢复。从图中可以看出，在 NSF 网络与 COST239 网络中呈现相似趋势，指定时延生存性部署模型明显好于 MCC 与 SVVR 管控模型，与 C-MPC 故障恢复概率基本保持持平。这是由于本文部署模型根据控制时延为每个网络节点创建了至少两条的控制通道，根据控制信道的优先级决定当前使用的控制链路，若当前控制链路出现故障，优先级次之的链路将作为新的控制链路，这与 C-MPC 的基于最小点覆盖模型起到了相同的保护作用。如图 3 所示，相同光纤百里故障概率下，指定时延管控模型在 COST239 网络中比在 NSF 网络中能够获得更好的保护，这是因为 COST239 网络中网络节点连通度更大，使得在控制器到网络节点的控制链路增多，从而具有更好的保护作用。本文所用控制器数量明显少于 C-MPC 部署的控制器数量，下文将给出与 C-MPC 部署模型控制器部署个数的比较。

文献[8]中指出，网状网恢复机制的触发时延为 100 ms，环形保护机制的触发时延为 50 ms，控制平面的高效运行的容忍时延为 10 ms。因此，在保

证上述网络条件的基础之上，实现本文部署算法与 C-MPC 算法控制平面部署控制器个数的对比。根据文献[8]中时延参数的说明，本文指定时延参数在 10 ms 以下与文献[6]中的 C-MPC 算法中所需控制器的部署个数进行比较。图 4 所示为在不同的指定时延约束条件下控制器的部署个数。从图 4 中可以看出，在 NSF 网络中指定控制时延小于等于 5 ms 时，本文部署算法只需 4 个部署节点就可保证每个节点至少有 2 条的控制链路；而 C-MPC 算法部署节点的个数始终需要 9 个部署节点。在指定时延为 10 ms 时，在 NSF 网络中本文所提算法只需部署 1 个控制器即可完成整个网络的覆盖，并保证控制器到网络节点之间至少有 2 条控制链路。在 COST239 网络中，在小于等于 5 ms 时，本文算法需要 3 个部署节点，在时延约束大于 6 ms 时，本文算法只需 1 个控制器部署节点即可完成整个网络的覆盖，C-MPC 算法在时延约束大于 5 ms 时始终需要 4 个控制器部署节点。在指定时延小于等于 4 ms 时，NSF 网络和 COST239 网络中 C-MPC 算法均无法满足 4 ms 的时延要求，虽然本文部署节点个数增多，但可任意满足用户的指定时延需求，能够更好的提高网络的整体性能。

为验证本文算法控制平面的时延特性，本文在满足文献[8]指标要求下，采用 Floodlight 和 Mininet 仿真实验平台创建拟真实验环境，在测试环境中采用 iperf 发包工具产生 TCP 和 UDP 数据包，采用

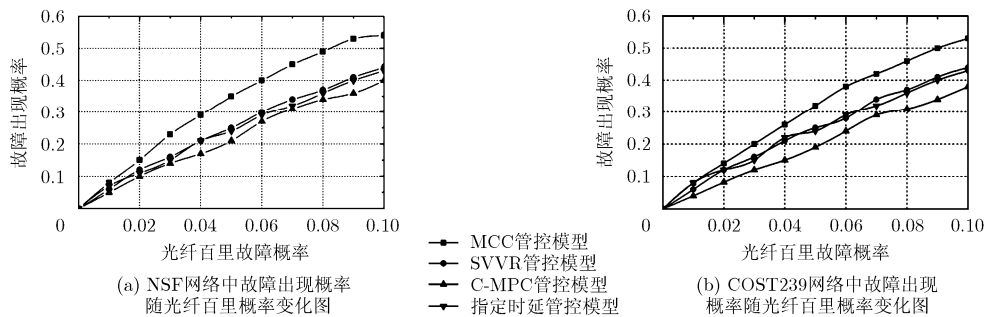


图 2 4 种部署模型可靠性比较 1

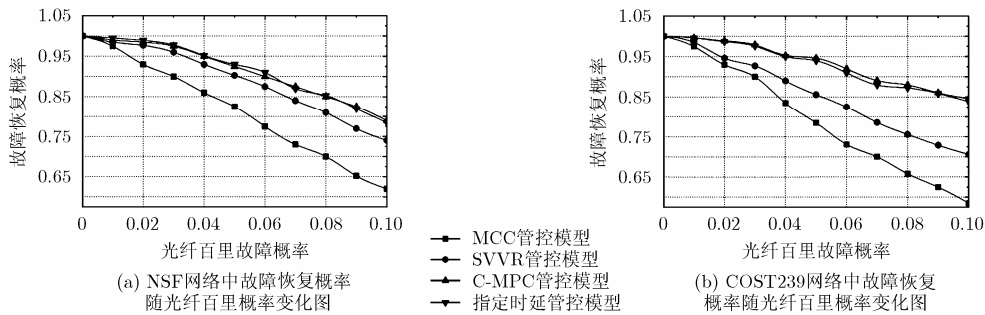


图 3 4 种部署模型可靠性比较 2

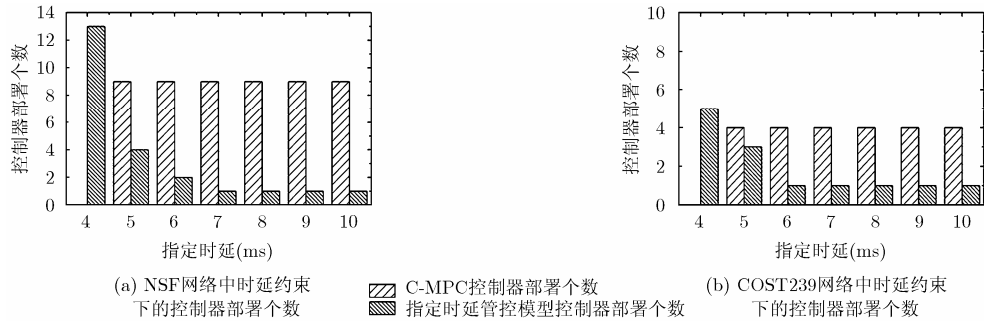


图 4 不同指定时延约束条件下的控制器部署个数

用户流量发生模型模拟真实的网络数据包，并将模拟产生的数据包注入到 Mimirnet 创建的网络拓扑当中，以使得仿真环境更加逼真。如图 5 所示为 4 种部署算法控制时延对比。仿真结果表明，在 NSF 和 COST239 两种网络环境下，本文算法控制时延最小时延明显小于其余 3 种算法的时延，最大时延明显小于 SVVR 算法，比 MCC 算法控制时延稍有改观，比 C-MPC 算法控制平均时延上有所提高。本文算法在用户指定时延参数的前提下，能够满足文献[8]所提控制时延的指标要求，能以尽可能少的控制器部署个数满足更好的网络性能需求。本文部署算法能达到这样的性能需求，这是因为控制器与网络节点之间选择至少 2 条链路作为控制链路，并且控制器根据控制链路列表选择优先级高的链路作为指令传输链路，这样有效地减少了控制时延和提高了控制平面的生存性。

### 5 结束语

本文提出软件定义网络中基于指定时延的控制

器生存性部署方法，该部署方法在用户指定控制实时延的基础之上，确保控制器到每个网络节点之间至少有两条控制链路，每条控制链路根据控制时延拥有不同的优先级，控制器在下发控制指令时根据控制链路的联通性和优先级确定由哪条链路向目标节点传递控制指令。根据部署算法确定控制器的部署位置后，本文对部署算法的部署结果进行了有效的仿真实验验证，结果表明本文算法起到了 C-MPC 相同的保护作用，与 MCC 算法相比，使 SDON 网络控制平面可靠性提高了 20%。用户在指定 10 ms 的约束条件下，在 NSF 和 COST239 网络中控制器的需求数目比 C-MPC 算法分别减少了 8 个和 3 个，即可减少 88%和 75%的部署成本。本文所提软件定义光网络中时延约束的控制器生存性部署方法能够有效地满足用户的控制时延需求，提高控制平面的生存性，并有效地减少了控制器的部署个数，减少了网络的投资成本，提高了网络的整体性能。

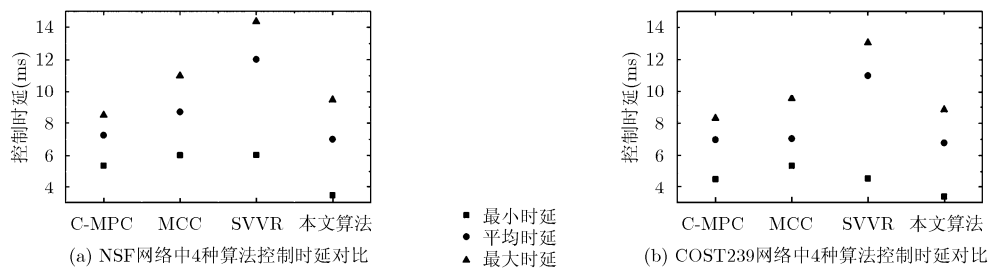


图 5 控制时延对比

### 参 考 文 献

[1] YANG Hui, ZHANG Jie, ZHAO Yongli, et al. SUDO: Software defined networking for ubiquitous data center optical interconnection[J]. *IEEE Communications Magazine*, 2016, 54(2): 86-95. doi: 10.1109/MCOM.2016.7402266.

[2] MCKEOWN N, ADNRSON T, BALAKRISHNAN H, et al. OpenFlow: Enabling innovation in campus networks[J]. *ACM SIGCOMM Computer Communication Review*, 2008, 38(2): 69-74. doi: 10.1145/1355734.1355746.

[3] JAIN S, KUMAR A, MANDAL S, et al. B4: Experience with a globally-deployed software defined WAN[J]. *ACM SIGCOMM Computer Communication Review*, 2013, 43(4): 3-14. doi: 10.1145/2534169.2486019.

[4] 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. doi: 10.13328/j.cnki.jos.004701.

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. doi: 10.13328/j.cnki.

- jos.004701.
- [5] CHOI J S. Design and implementation of a PCE-based software-defined provisioning framework for carrier-grade MPLS-TP networks[J]. *Photonic Network Communications*, 2015, 29(1): 96–105. doi: 10.1007/s11107-014-0472-0.
- [6] 熊余, 董先存, 李圆圆, 等. 软件定义光网络中基于最小点覆盖的控制平面跨层生存性设计[J]. 电子与信息学报, 2016, 38(5): 1211–1218. doi: 10.11999/JEIT150815.
- XIONG Yu, DONG Xiancun, LI Yuanyuan, *et al.* The cross-layer survivable design of control plane based on minimum point covering in software defined optical network [J]. *Journal of Electronics & Information Technology*, 2016, 38(5): 1211–1218. doi: 10.11999/JEIT150815.
- [7] PAKZAD F, PORTMAN M, TAN W L, *et al.* Efficient topology discovery in OpenFlow-based Software Defined Networks[J]. *Computer Communications*, 2016, 77(1): 52–61. doi: 10.1016/j.comcom.2015.09.013.
- [8] HELLER B, SHERWOOD R, MCKEOWN N, *et al.* The controller placement problem[C]. The First ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, New York, USA, 2012: 7–12.
- [9] 姚琳元, 陈颖, 宋飞, 等. 基于时延的软件定义网络快速响应控制器部署[J]. 电子与信息学报, 2014, 36(12): 2802–2808. doi: 10.3724/SP.J.1146.2014.00211.
- YAO Linyuan, CHEN Ying, SONG Fei, *et al.* Delay-aware controller placement for fast response in software-defined network[J]. *Journal of Electronics & Information Technology*, 2014, 36(12): 2802–2808. doi: 10.3724/SP.J.1146.2014.00211.
- [10] AZODOLMOLKY S, WIEDER P, and YAHYAPOUR R. Performance evaluation of a scalable software-defined networking deployment[C]. IEEE 2013 Second European Workshop on Software Defined Networks, Berlin, GER, 2013: 68–74.
- [11] MÜLLER L F and OLIVEIRA R R. Survivor: An enhanced controller placement strategy for improving SDN survivability[C]. IEEE Global Communications Conference, Austin, USA, 2014: 1909–1915.
- [12] GUO Z, HU Y, SHOU G, *et al.* An implementation of multi-domain software defined networking[C]. International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015), Shanghai, China, 2015: 1–5.
- [13] ZHANG Y, BEHESHTI N, TATIPAMULA M, *et al.* On resilience of split-architecture networks [C]. IEEE Global Telecommunications Conference (GLOBECOM), Houston, USA, 2011: 1–6.
- [14] YU M, REXFORD J, FREEDMAN M J, *et al.* Scalable flow-based networking with DIFANE[J]. *ACM SIGCOMM Computer Communication Review*, 2010, 40(4): 351–362. doi: 10.1145/1851182.1851224.
- [15] GENYA Ishigaki and NORIHIKO Shinomiya. Controller placement algorithm to alleviate burdens on communication nodes[C]. International Conference on Computing, Networking and Communications (ICNC), Kauai, Hawaii, USA, 2016: 1–5.
- [16] LONG L and KAMAL A E. Tree-based protection of multicast services in WDM mesh networks[C]. Global Telecommunications Conference, Honolulu, USA, 2009: 1–6.
- 曾 帅: 男, 1982 年生, 博士, 讲师, 研究方向为未来网络、软件定义网络、移动互联网.
- 盖绍聪: 男, 1991 年生, 硕士生, 研究方向为软件定义光网络抗毁技术.
- 张 毅: 男, 1970 年生, 硕士, 教授, 研究方向为未来互联网、网络测量及物联网应用.
- 赵国锋: 男, 1972 年生, 博士, 教授, 研究方向为未来网络、软件定义网络、网络测量.
- 左理政: 女, 1994 年生, 硕士生, 研究方向为软件定义光网络抗毁技术.