

## 基于时延的软件定义网络快速响应控制器部署

姚琳元\* 陈颖 宋飞 张宏科

(北京交通大学下一代互联网互联设备国家工程实验室 北京 100044)

**摘要:** 目前大多数针对软件定义网络(SDN)中控制器的部署方案均重点考虑传输时延(PD)对性能的影响,忽略了发送时延(TD)对于部署效果的影响。该文提出基于时延的网络快速响应控制器部署方案。首先,在合理考虑传输和发送两类时延的基础上,完善了已有的平均时延/最大时延最小化模型,并对两种模型是否存在最优解进行了理论证明;其次,利用模糊集理论得出了一种时延优化模型;第三,结合是否考虑发送时延提出了两种部署算法:传输算法和输送算法。为了测试方案的性能,选取实际网络拓扑及数据进行验证。结果表明输送算法在网络的响应速度及稳定性方面优于传输算法,时延优化模型在总时延方面较平均时延/最大时延最小化模型效果更优。

**关键词:** 软件定义网络; 部署; 控制器; 时延

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2014)12-2802-07

DOI: 10.3724/SP.J.1146.2014.00211

## Delay-aware Controller Placement for Fast Response in Software-defined Network

Yao Lin-yuan Chen Ying Song Fei Zhang Hong-ke

(National Engineering Laboratory for Next Generation Internet Interconnection Devices,  
Beijing Jiaotong University, Beijing 100044, China)

**Abstract:** Most of Controller placements take the Propagation Delay (PD) as the important consideration in Software-Defined Network (SDN), ignoring the influence of the Transmission Delay (TD) on the network performance. This paper provides a delay-aware controller placement for fast response. First, the Controller placement is formulated as an optimization problem based on PD and TD. Average delay and maximum delay minimization models are updated, of which the processes about the optimal solution are circumstantiated. Further, delay optimization model is deduced by fuzzy set theory. Finally, according to whether or not considering TD, two placement algorithms, Transmission and Propagation Algorithm (TPA) and Propagation Algorithm (PA), are presented. In order to measure the performance of the solutions, a factual network topology is chosen and the simulation result shows that TPA superiorities over PA in terms of response speed and network stability, the total delay of delay optimization model is less than the others.

**Key words:** Software-Defined Network (SDN); Placement; Controller; Delay

### 1 引言

软件定义网络 (Software-Defined Network, SDN)是由美国斯坦福大学 Clean Slate 研究组<sup>[1]</sup>提出的一种基于集中控制的新型网络架构<sup>[2]</sup>。在不改变传统 IP 数据包转发行为的基础上,SDN 将传统路由器/交换机的数据转发与逻辑控制进行分离,实现数据域与控制域的解耦<sup>[3,4]</sup>。在数据域中,SDN 定义为交换机(Switch)负责数据包的接收、存储和转发。在控制域中,SDN 定义为控制器(Controller)负责网络的顶层设计,与交换机连接;收集网络信息,实现

全局处理;通过控制交换机的流表项指导交换机对数据包的处理,实现网络的集中管理,为用户提供更好、更便捷的网络服务<sup>[5,6]</sup>。

良好的上网体验,要求网络畅达、快速,而合理、方便地部署控制器显得尤为重要。当今,核心网设备的部署已趋于固定,重新搭建网络拓扑不符合绿色、节能、便捷等现代网络理念<sup>[7]</sup>。在已铺设的网络架构基础上进行优化,与现有网络实现良好的硬件连接和逻辑连接是 SDN 发展的必然趋势。与此同时,SDN 在设计之初就确定了数据转发与控制相分离的基本思想,采用集中式管理,实现网络的一致性,势必要求网络的快速响应<sup>[8]</sup>。在已有网络的基础上,考虑控制器的部署位置是实现网络快速响应的有效方式之一。

文献[9]首次提出了 SDN 网络中控制器的部署

2014-02-19 收到, 2014-05-20 改回

国家自然科学基金(61301081), 高等学校博士学科点专项科研基金(20120009120005)和中央高校科研业务费专项基金(2014JBM010)资助课题

\*通信作者: 姚琳元 11111020@bjtu.edu.cn

问题：(1)网络中需要多少个控制器；(2)怎么部署控制器。为实现定量分析，作者首先定义了两个测试参数平均时延和最大时延；然后，利用真实拓扑结构进行分析，包括：控制器数量对时延的影响，平均时延与最优时延的对比等；最后，引入交换速度、环境保护和网络恢复 3 个网络时间参数，综合考虑网络对时延的需求，提出一个控制器即可满足当下网络的响应要求。在分析过程中，文献[9]只考虑了传输时延，忽略了发送时延对网络响应速率的影响。

文献[10, 11]分别从网络可靠性和网络弹性两个角度对控制器的部署进行了分析、建模。文献[10]提出了一个新的参量，即可预期控制链路亏损百分比(Expected Percentage of Control Path Loss)，并从可靠性的角度对 SDN 中控制器的部署进行了分析。文献提出了 4 种不同的部署算法，利用真实拓扑对算法进行了验证，其中模拟退火法(Simulated Annealing)优于其他算法；认为控制器的数量应当适当选择，过多或过少都会影响网络的可靠性。文献[11]从网络弹性和容错的角度对 SDN 中控制器的部署进行了分析，应用帕累托优化理论，实现网络弹性和网络容错的权衡；通过对 OS3E 拓扑及拓扑动物园(Topology Zoo)中的大量拓扑进行分析，认为保证网络的连续，网络中至少要有 20%的节点设置为控制器。

文献[12]充分利用固定式网络微积分结构，建立数学模型，解决了 SDN 在部署过程中的可扩展性问题。文中将控制器分为本地控制器(Local Controller)和超级控制器(Root Controller)，利用数学模型从本地控制器中获取固定格式的事件时延和缓存长度，利用经典的排队理论解决了边界和最差条件下的两种部署情况。另外，文献[12]建立了一个控制器互动分析模型，该模型给出了控制器的进程、流量控制函数，利于网络设计人员计算出控制器的时延和缓存需求。对于下一步工作，文章提出了两个方向的目标：(1)采用随机网络微积分的方法对 SDN 中的部署问题进行研究，完成相关仿真和实验工作；(2)在参照边界条件的同时，考虑均衡条件下的平均值。

本文旨在通过合理部署控制器，实现网络的快速响应。影响网络响应时延的因素有很多，主要包括传输链路的距离、链路上的数据传输速度、网络中传输的数据量、网络设备的数据收发能力等<sup>[9]</sup>。文献[13]定义了网络时延的 4 个变量：处理时延、排队时延、传输时延(Propagation Delay, PD)和发送时延(Transmission Delay, TD)。当网络处于通畅状态时，排队时延可忽略不记，而处理时延一般是  $\mu\text{s}$  数量级甚至更低，也可忽略不记。传输时延受链路距

离的影响，发送时延受数据包大小及端口速率的影响，二者对网络响应时延的影响不可忽略。文献[9]在部署控制器时仅考虑传输时延而忽略了发送时延。本文拟从传输时延和发送时延对控制器的部署进行重新规划。

通过对相关控制器部署方案的研究，本文提出了一种基于传输时延和发送时延的网络快速响应控制器部署方案。首先，完善平均时延和最大时延最小化模型，在此基础上，推导出时延优化模型；其次，提出了解决上述模型的基于传输时延和发送时延的输送算法(Transmission and Propagation Algorithm, TPA)以及基于传输时延的传输算法(Propagation Algorithm, PA)；最后，利用实际拓扑数据，完成数据仿真，得出不同模型最佳部署方案以及对应的平均时延、最大时延和总时延，并与相关部署方案进行了对比、分析。

## 2 模型假设及理论证明

网络时延的主要衡量标准分为平均时延和最大时延<sup>[9]</sup>，本节从传输时延和发送时延两个角度，研究平均时延和最大时延，分析网络响应速度。具体工作如下：首先设定基本假设，定义基本的参考变量，完成数学化描述，推导出参数间的数学关系；在此基础上，对现有的平均时延模型和最大时延模型进行完善，添加发送时延变量，得出平均时延最小化模型和最大时延最小化模型，并利用优化理论对上述模型是否存在最优解进行理论证明<sup>[14]</sup>；最后从上述模型中提炼出时延优化模型，引入模糊集理论<sup>[15]</sup>，给出时延优化模型的解决方案。下文的公式表示同时考虑传输时延和发送时间的数学分析模型，与第 3 节中 TPA 对应；去除所述模型中发送时延变量的数学分析模型，与第 3 节中 PA 对应，限于篇幅，本节不做说明。

### 2.1 基本假设和参数定义

假设  $G(V, E)$  表示网络拓扑， $V = \{v_1, v_2, \dots, v_q\}$  代表拓扑中网络节点集合， $q$  是拓扑中节点数量， $E = \{(v_s, v_t) | v_s, v_t \in V\}$  代表网络节点之间链路的集合；每一个交换机由唯一网络节点标识，控制器部署位置从拓扑网络节点中进行筛选，也唯一标识；交换机与控制器可重复使用同一网络节点，此种情况下，认为二者的传输时延为 0； $C = \{c_1, c_2, \dots, c_k\}$  表示控制器集合，其中  $C \subseteq V$ ， $\{c_m^v\}$  代表受控制器  $c_m$  控制的交换机集合， $|c_m^v|$  代表受控制器  $c_m$  控制的交换机数量； $L_{v_t}^{v_s}$  代表节点  $v_t$  与  $v_s$  之间的最短链路长度； $I_{v_t}^{v_s}(t)$  代表节点  $v_s$  与  $v_t$  之间需要传输的数据量； $y = (y_c, c \in C)$  为控制器的部署方案； $D_P, \bar{D}_P, D_T, \bar{D}_T$

分别代表传输时延、平均传输时延、发送时延、平均发送时延；用  $S$  表示链路的传输速率， $B$  表示网络节点的网口收发速率，本文认为拓扑中所有链路的  $S$  相同，所有网络节点  $B$  相同，每一个交换机由唯一的控制器控制。

通过上述参数的数学化描述，传输时延、发送时延可通过带宽、信息量、链路长度等基本参数进行表示。式(1)表示部署方案  $y_c$  下数据从  $v_n$  移动到  $c_m$  在链路上的传输时延；式(2)表示部署方案  $y_c$  下数据在  $v_n$  与  $c_m$  之间的最短链路上传输时的发送时延， $h_{c_m}^{v_n}$  为数据在  $v_n$  与  $c_m$  之间的最短链路上传输经过的节点数量，不包括  $v_n$  和  $c_m$  自身，当  $v_n$  与  $c_m$  的最短链路为二者之间的直连链路时， $h_{c_m}^{v_n}$  为 0；式(3)表示部署方案  $y_c$  下交换机到控制器的平均传输时延；式(4)表示部署方案  $y_c$  下交换机到控制器的最大传输时延；式(5)表示部署方案  $y_c$  下交换机到控制器的平均发送时延；式(6)表示部署方案  $y_c$  下交换机到控制器的最大发送时延。

$$D_P^{v_n, c_m}(y_c) = (y_c) L_{c_m}^{v_n} / S \quad (1)$$

$$D_T^{v_n, c_m}(y_c) = (y_c) [(h_{c_m}^{v_n} + 1) I_{c_m}^{v_n}(t)] / B \quad (2)$$

$$\overline{D_P^C}(y_c) = \frac{1}{q} \sum_{c_m: c_m \in C} \sum_{v_n: v_n \in \{c_m^v\}} (y_c) L_{c_m}^{v_n} / S \quad (3)$$

$$D_{wP}^C(y_c) = \max_{y_c \in y} \left( (y_c) L_{c_m}^{v_n} / S; c_m: c_m \in C, v_n: v_n \in \{c_m^v\} \right) \quad (4)$$

$$\overline{D_T^C}(y_c) = \frac{1}{q} \sum_{c_m: c_m \in C} \sum_{v_n: v_n \in \{c_m^v\}} (y_c) [(h_{c_m}^{v_n} + 1) I_{c_m}^{v_n}(t)] / B \quad (5)$$

$$D_{wT}^C(y_c) = \max_{y_c \in y} \left( (y_c) [(h_{c_m}^{v_n} + 1) I_{c_m}^{v_n}(t)] / B; c_m: c_m \in C, v_n: v_n \in \{c_m^v\} \right) \quad (6)$$

## 2.2 平均时延最小化模型

$$(P1) \left. \begin{array}{l} \min_{y_c \in y} \overline{D_P^C}(y_c) + \overline{D_T^C}(y_c) \\ \text{s.t. } \{c_m^v\} \subseteq V, |c_m| \leq q, C \subseteq V \end{array} \right\} \quad (7)$$

通过对上述参数的整理，选取传输时延与发送时延二者和的平均值作为目标函数，得出平均时延最小化模型(P1)。限制条件包括控制器的数量，控制器的部署方式。该模型可用于分析、讨论网络控制器部署中平均时延的最小值，研究网络时延的总体走势，反应网络时延的平均趋势。定理 1 是对该模型最优解的说明。

**定理 1** 平均时延最小化模型(P1)是凸优化问题，当控制器数量一定时，控制器的最优部署方案存在，即存在最优解。

**证明**  $q, S, B, h$  为已知常数，式(7)中  $\{c_m^v\}$ ,

$|c_m|, C, L_{c_m}^{v_n}, I_{c_m}^{v_n}$  为变量，因此目标函数可调整为：

$f(L, I) = \overline{D_P^C}(y_c) + \overline{D_T^C}(y_c)$ 。因为  $\nabla^2 f(L, I) \succeq 0$ ，因此目标函数为凸函数；约束条件为线性、离散，因此可行域有界；所以该模型有最优解<sup>[14]</sup>。证毕

## 2.3 最大时延最小化模型

$$(P2) \left. \begin{array}{l} \min_{y_c \in y} \max_{c_m: c_m \in C, v_n: v_n \in \{c_m^v\}} (D_P^{v_n, c_m}(y_c) + D_T^{v_n, c_m}(y_c)) \\ \text{s.t. } \{c_m^v\} \subseteq V, |c_m| \leq q, C \subseteq V \end{array} \right\} \quad (8)$$

(P2) 为最大时延最小化模型，将传输时延与发送时延二者和的最大值作为参考点，选取最小值一组，作为该模型的目标函数，通过等价变化，得到

$$(P2) \left. \begin{array}{l} \min_{y_c \in y} \max_{c_m: c_m \in C, v_n: v_n \in \{c_m^v\}} (D_P^{v_n, c_m}(y_c) + D_T^{v_n, c_m}(y_c)) \\ \Leftrightarrow \left\{ \begin{array}{l} \min_{y_c \in y, t \in R} t \\ \text{s.t. } \max_{c_m: c_m \in C, v_n: v_n \in \{c_m^v\}} (D_P^{v_n, c_m}(y_c) + D_T^{v_n, c_m}(y_c)) = t \end{array} \right. \\ \Leftrightarrow \left\{ \begin{array}{l} \min_{y_c \in y, t \in R} t \\ \text{s.t. } (D_P^{v_n, c_m}(y_c) + D_T^{v_n, c_m}(y_c)) \leq t, \\ c_m: c_m \in C, v_n: v_n \in \{c_m^v\} \end{array} \right. \end{array} \right\} \quad (9)$$

最终数学模型为

$$(P2) \left. \begin{array}{l} \min_{y_c \in y, t \in R} t \\ \text{s.t. } (D_P^{v_n, c_m}(y_c) + D_T^{v_n, c_m}(y_c)) \leq t, c_m: c_m \in C, \\ v_n: v_n \in \{c_m^v\}, \{c_m^v\} \subseteq V, |c_m| \leq q, C \subseteq V \end{array} \right\} \quad (10)$$

该模型可以反应出当前部署方案下的网络最大响应时延，同时，以最大响应时延作为限制条件，找到最优部署方案，定理 2 是对该模型最优解的说明。

**定理 2** 最大时延最小化模型(P2)是凸优化问题，当控制器数量一定时，控制器的最优部署方案存在，即存在最优解。

**证明**  $q, S, B, h$  为已知常数，式(10)中  $\{c_m^v\}, |c_m|, C, L_{c_m}^{v_n}, I_{c_m}^{v_n}$  为变量，目标函数可调整为： $f(L, I, h, t) = t$ ，因为  $\nabla^2 f(L, I, h, t) \succeq 0$ ，因此目标函数为凸函数；约束条件为线性、离散，因此可行域有界；所以该模型有最优解<sup>[14]</sup>。证毕

## 2.4 时延优化模型

$$(P3) \left. \begin{array}{l} \min_{y_c \in y} \overline{D_P^C}(y_c) + \overline{D_T^C}(y_c) \text{ and } \min_{y_c \in y, t \in R} t \\ \text{s.t. } (D_P^{v_n, c_m}(y_c) + D_T^{v_n, c_m}(y_c)) \leq t, \\ c_m: c_m \in C, v_n: v_n \in \{c_m^v\}, \{c_m^v\}, |c_m|, C \end{array} \right\} \quad (11)$$

$$F_1(L, I) = g_1(-\overline{D_P^C}(y_c) - \overline{D_T^C}(y_c))$$

$$= \begin{cases} 0, & -\overline{D_P^C}(y_c) - \overline{D_T^C}(y_c) < {}^*D_{P1} - \theta_1 \\ 1 - \frac{1}{\theta_1} \left( {}^*D_{P1} + \left( \overline{D_P^C}(y_c) + \overline{D_T^C}(y_c) \right) \right), & {}^*D_{P1} - \theta_1 \leq -\overline{D_P^C}(y_c) - \overline{D_T^C}(y_c) < {}^*D_{P1} \\ 1, & -\overline{D_P^C}(y_c) - \overline{D_T^C}(y_c) \geq {}^*D_{P1} \end{cases} \quad (12)$$

$$F_2(L, I) = g_2(-t)$$

$$= \begin{cases} 0, & -t < {}^*D_{P2} - \theta_2 \\ 1 - \frac{1}{\theta_2} ({}^*D_{P2} + t), & {}^*D_{P2} - \theta_2 \leq -t < {}^*D_{P2} \\ 1, & -t \geq {}^*D_{P2} \end{cases} \quad (13)$$

为实现网络快速响应，本节同时考虑平均时延最小化模型(P1)和最大时延最小化模型(P2)，将二者的目标函数进行整合，形成多目标规划问题，得出时延优化模型(P3)。该模型的两个目标函数在优化过程中可能发生冲突，因此，需引入折中机制，这里采用模糊集理论整合可能发生冲突的两个目标函数<sup>[15,16]</sup>。

首先，引入 ${}^*D_{P1}$ 和 ${}^*D_{P2}$ 作为(P1)和(P2)的两个单目标最优值；然后，依据各目标重要性程度不同，给出两个单目标函数的伸缩性指标 $\theta_1$ 和 $\theta_2$  ( $\theta_1, \theta_2 > 0$ )， $\theta_i$ 越小，目标函数越重要，反之，当 $\theta_i$ 趋近于无穷大时，目标函数 $P(i)$ 趋近于无效，多目标问题转变为单目标问题；最后，得到两个模糊目标集 $F_1$ 和 $F_2$ ，其隶属函数具体表示如式(12)，式(13)。

令 $F = F_1 \cap F_2, \lambda = F_1(y_c) \wedge F_2(y_c)$ ，设 $y_c^*$ 满足 $F(y_c^*) = \sup_{y_c \in Y} F(y_c) = \sup_{y_c \in Y} (F_1(y_c) \wedge F_2(y_c))$ ，至此，该线性规划问题转化为普通线性规划问题(P4)，使用单纯形法可求出该线性规划问题的最优解 $\lambda^*$ ，同时得出最优部署方案 $y_c^*$ ：受控制器 $c_m$ 控制的交换机集合 $\{c_m^v\}$ 和对应集合中交换机数量 $|c_m|$ 。

$$(P4) \quad \left. \begin{aligned} \max(\lambda) \\ \text{s.t.} \quad & (D_P^{v,c_m}(y_c) + D_T^{v,c_m}(y_c)) \leq t, \\ & 1 - \frac{1}{\theta_1} \left( {}^*D_{P1} + \left( \overline{D_P^C}(y_c) + \overline{D_T^C}(y_c) \right) \right) \geq \lambda, \\ & 1 - \frac{1}{\theta_2} ({}^*D_{P2} + t) \geq \lambda, \{c_m^v\}, |c_m|, C \end{aligned} \right\} \quad (14)$$

### 3 部署算法

为验证发送时延在网络响应速度以及控制器部署等方面的重要程度，本文首先参照文献[9]，设计了PA，然后依据上文提出的传输时延和发送时延，设计了TPA。

两种算法定义网络节点到控制器的单向传输方向为数据传输方向，认为控制器的部署位置位于网络拓扑中的网络节点处，当某网络设备的位置与控制器的位置重叠时，二者的传输时延为0；当部署多个控制器，需计算其中一个控制器到网络节点的时延时，两种算法设定其余控制器和与之连接的网络节点处于不能通信状态，即传输时延设为无穷大。PA和TPA都以迪杰斯特拉(Dijkstra)算法和贪心(Greedy)算法为基础，以实际拓扑作为算法的输入，得出不同条件下的最优部署方案。二者的不同之处在于：

(1)PA 将拓扑中的网络节点视为普通点，只计算传输时延，通过寻找最短路径，得到网络设备到控制器的最短距离，进而获得基于传输时延的最佳部署方案；为统一网络参数，在PA的最后，增加了发送时延，表示在网络运转情况下，即有数据在网络中传输时，网络响应时延的情况；

(2)TPA 将传输时延和发送时延作为部署方案的参考因素，当计算拓扑中某个网络节点到控制器的响应时延时，TPA提出将网络节点以外的其他节点设置为与该网络节点相同的发送时延；

(3)TPA可直接使用第2节中的3个数学模型，PA在使用上述模型时，需去掉模型中的发送时延一项。

具体算法伪代码如表1和表2所示。

### 4 性能评估

本节以欧洲实际拓扑为例进行了仿真，数据源自文献[17,18]。该参考实例提供了37个城市节点的经纬度坐标、节点间的连接关系和发送信息量(Information, I)。仿真按照地名的首字母由1至37重新排列了网络节点序号，参考实例链路，建立了节点关系，得到如图1所示的网络拓扑。在仿真前，分别设置链路的传输速率 $S$ 为光速的2/3，网络节点的网口收发速率 $B$ 为1.5 Gbps；仿真过程中，控制器的个数从1到8分别设置，按照PA和TPA，得到两组最优部署方案。图1(a)，图1(b)分别表示时延优化模型下，控制器数量为5时，PA和TPA得出的控制器部署方案，实心标记处为控制器位置。从图1能看出，不同算法下得出的部署位置明显不同，下面从最大时延和平均时延对比分析不同模型不同算法的最优方案。

图2为利用PA和TPA求出的3种优化模型P1, P2, P3的时延对比图, mean表示平均时延, max表示最大时延, mm表示平均时延与最大时延的和。图2(a)表示不考虑信息量I, PA得出的3种优化模型分别对应的平均时延(mean), 最大时延(max)和总时延(mm)随控制器数量增加的变化情况; 图2(b)表示在图2(a)得出的3种最优部署方案下, 增加I后, 时延随控制器数量增加的变化情况; 图2(c)表示使用TPA得出的3种优化模型分别对应

表1 PA算法

## 算法1 PA

输入: 网络拓扑  $G(V, E)$ , 拓扑节点数目  $q$ , 控制器数量  $k$ , 发送时延  $I_{c_m}^v(t)(c_m \in C, v_s \in V)$ , 伸缩性指标  $\theta_1$  和  $\theta_2$ ;  
输出: 最大时延最小化部署方案; 平均时延最小化部署方案; 时延优化部署方案;

- (1) 初始化: S(拓扑节点状态); dist(拓扑节点到控制器的时延);  
 $C = \{c_1, c_2, \dots, c_k\}$   
 enddist(拓扑节点到控制器的最短时延);  
 totaltime(总时延);
- (2) for  $m=1$  to  $q$  do  
 $S[m]$  and  $dist[m] \leftarrow \emptyset$   
 enddist  $[m] \leftarrow \infty$  {初始化拓扑节点状态、拓扑节点到控制器的时延和最短时延}
- (3) for  $(c_x, \dots, c_y) \subseteq C$  do  
 for  $m=1$  to  $q$  do {得出  $(c_x, \dots, c_y)$  下, 节点所属的控制器}  
 for  $c = c_x$  to  $c_y$  do {得出每个控制器到连接节点  $v_m$  的最短时延}  
 $L_{\{C: c_m \notin C\}}^m \leftarrow \infty$  {将  $c_m$  以外的控制器与其它节点连接的传输时延设为无穷大}  
 $S[m], dist[m]$  and  $\{c_m^v\}$  {得出每一种部署方案下, 拓扑节点到控制器的最短时延, 并依据最短时延, 确定受控制器  $c_m$  控制的交换机集合}  
 end for  
 end for
- (4) If (enddist[wi]>dist[wi]) then enddist[wi]=dist[wi]{依据伸缩性指标得出相应的最佳部署方案}  
 end if  
 end for  
 totaltime = enddist[wi] +  $I_{c_m}^v(t)$  {在该部署方案下, 将发送时延的时延添加至拓扑节点到控制器的最短时延, 得出基于传输时延最优部署方案的总时延}

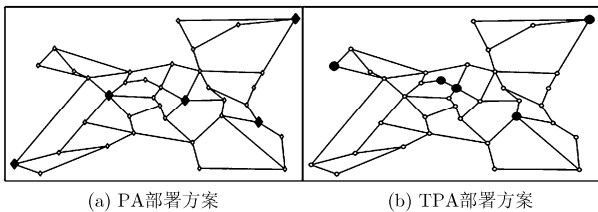


图1 网络拓扑图

表2 TPA算法

## 算法2 TPA

输入: 网络拓扑  $G(V, E)$ , 拓扑节点数目  $q$ , 控制器数量  $k$ , 发送时延  $I_{c_m}^v(t)(c_m \in C, v_s \in V)$ , 传输时延  $L_{\{C: c_m \notin C\}}^v(c_m \in C, v_s \in V)$ , 伸缩性指标  $\theta_1$  和  $\theta_2$ ;

输出: 最大时延最小化部署方案; 平均时延最小化部署方案; 时延优化部署方案;

- (1) 初始化: S(拓扑节点状态); dist(拓扑节点到控制器的时延);  
 $C = \{c_1, c_2, \dots, c_k\}$ ;  
 enddist(拓扑节点到控制器的最短时延);
- (2) for  $m=1$  to  $q$  do  
 $S[m]$  and  $dist[m] \leftarrow \emptyset$   
 enddist  $[m] \leftarrow \infty$  {初始化拓扑节点状态、拓扑节点到控制器的时延和最短时延}
- (3) for  $(c_x, \dots, c_y) \subseteq C$  do {得出最短时延的部署方案}  
 for  $m=1$  to  $q$  do {得出  $(c_x, \dots, c_y)$  下, 节点所属的控制器}  
 for  $c = c_x$  to  $c_y$  do {得出每个控制器到连接节点  $v_m$  的最短时延}  
 $L_{\{C: c_m \notin C\}}^m \leftarrow \infty$  {将  $c_m$  以外的控制器与其它节点连接的传输时延设为无穷大}  
 $I_{c_k}^v \leftarrow I_{c_m}^v$  {将所有连接点的发送时延设为  $I_{c_k}^v$ }  
 $S[m], dist[m]$  and  $\{c_m^v\}$  {得出每一种部署方案下, 拓扑节点到控制器的最短时延, 并依据最短时延, 确定受控制器  $c_m$  控制的交换机集合}  
 end for  
 end for
- (4) If (enddist[wi]>dist[wi]) then enddist[wi]=dist[wi]{依据伸缩性指标得出相应的最佳部署方案}  
 end if  
 end for

的3种时延随控制器数量增加的变化情况。从图2可看出:

(1)图2(a)和图2(c)中, 3种模型得到的部署方案, 在PA信息量为0或TPA情况下, 时延随控制器数量的增多而下降, 控制器数量越多, 网络的响应速率越快;

(2)图2(b)中, 考虑发送时延, 在图2(a)的最优部署条件下, 时延随控制器数量的增多上下波动, 没有固定的变化规律, 不能体现控制器数量增多在部署过程中的优势;

(3)图2(b)和图2(c)中, 信息量相同情况下, TPA算法的3种时延明显低于PA算法的时延。

图3(a), 图3(b), 图3(c)分别代表3种模型最优部署方案的时延情况, 其中横坐标表示控制器数量, 纵坐标表示同一种算法PA或TPA下的最大时延与平均时延的比值。从图3中可以看出:

(1)3种模型下, TPA的比值要明显低于PA的

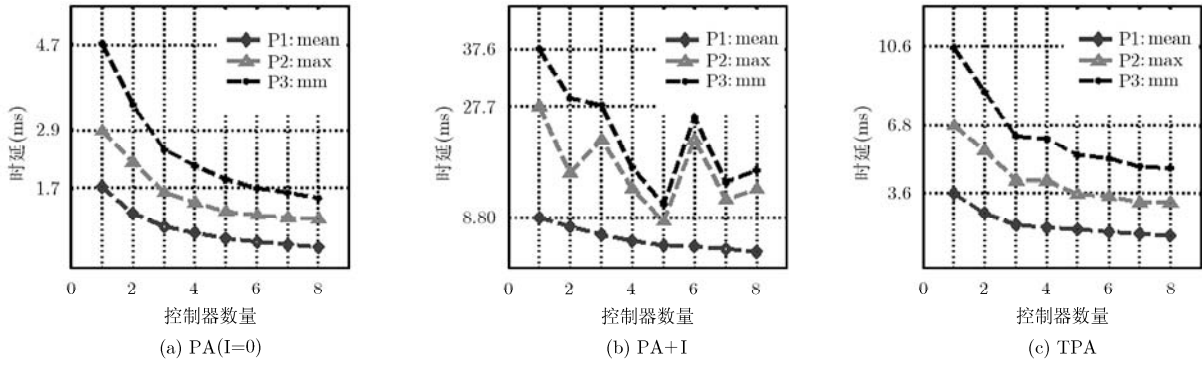


图 2 时延比较图

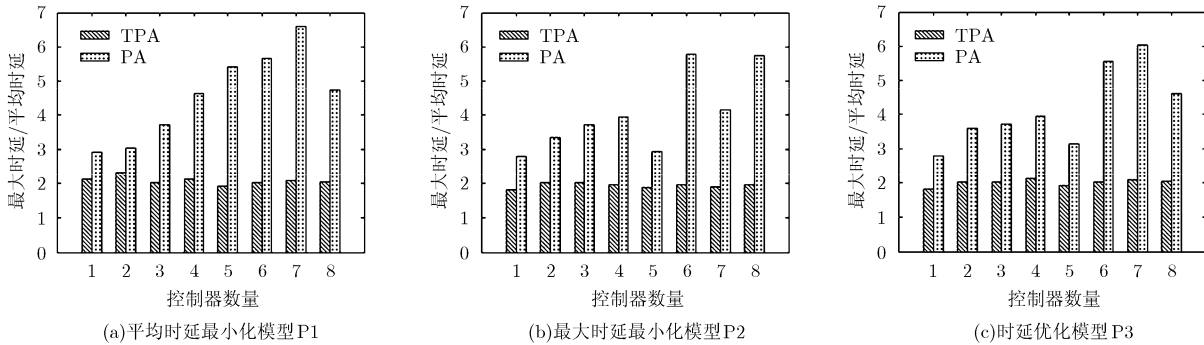


图 3 时延比值图

比值,表明在网络响应平均时延相同的情况下,TPA 较 PA 能够实现更快的网络响应,网络节点到控制器的响应时延波动范围更小。

(2)通过 3 个子图的对比,TPA 的比值比较稳定:控制器数量增多,平均时延下降,最大时延稳步降低;PA 的比值变动较大,无明显规律,当控制器数量为 1 时,比值最小,控制器数量增多,比值大小不一。

图 4(a)和图 4(b)分别表示 PA 算法( $I=0$ )和 TPA 算法下,3 种模型的时延比较图。从图 4 可以看出,模型 P3 的总时延取 P2, P1 二者中较低项,或低于二者。

从上述分析可以看出,较 PA, TPA 在平均时延、最大时延和总时延更低、比值更低,可以更快实现网络响应;时延优化模型较平均时延最小化模型和最大时延最小化模型在总时延更有优势,网络的响应时间更加集中。

### 5 结束语

在充分考虑传输时延和发送时延的前提下,本文提出了基于时延的网络控制器快速响应部署方案。已有时延模型仅考虑传输时延,在此基础上,本文增加了发送时延,得出了平均时延最小化模型和最大时延最小化模型,并结合二者特点,创建了

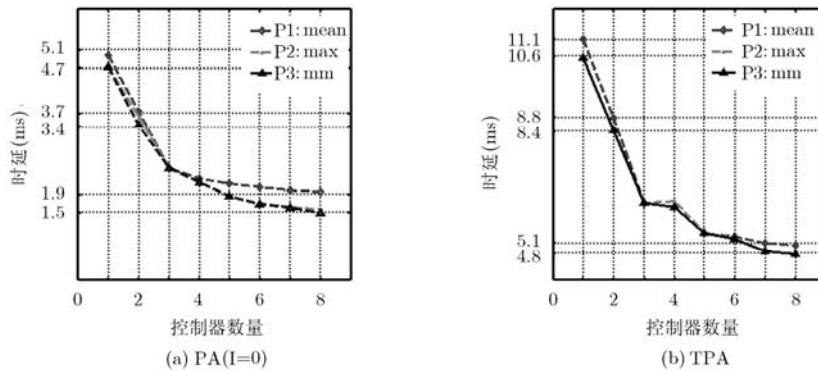


图 4 模型比较图

时延优化模型, 通过优化理论和模糊理论, 对模型的最优解进行了论证。按照是否考虑发送时延, 设计了PA和TPA, 通过仿真结果我们发现, 将发送时延纳入考虑范围可以增加网络的响应速度, 增加网络部署的稳定性; 时延优化模型可以使网络响应时间更加集中。下一步工作将利用网络仿真工具, 引入节点处理时延等变量, 同时从介数角度探讨网络中控制器的部署。

### 参考文献

- [1] 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.
  - [2] Jain S, Kumar A, Mandal S, *et al.* B4: experience with a globally-deployed software defined WAN[C]. The Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, Hong Kong, 2013: 3-14.
  - [3] ONF. software-defined networking: the new norm for networks, white paper[OL]. <https://www.opennetworking.org>, 2013. 12.
  - [4] Mueller J, Wierz A, Vingarzan D, *et al.* Elastic network design and adaptive flow placement in software defined networks[C]. 2013 22nd International Conference on Computer Communications and Networks (ICCCN), California, 2013: 1-6.
  - [5] 周焯, 杨旭, 李勇, 等. 基于分类的软件定义网络流表更新一致性方案[J]. 电子与信息学报, 2013, 35(7): 1746-1752.  
Zhou Ye, Yang Xu, Li Yong, *et al.* Classification based consistent flow update scheme in software defined network[J]. *Journal of Electronics & Information Technology*, 2013, 35(7): 1746-1752.
  - [6] Dixit A, Hao F, Mukherjee S, *et al.* Towards an elastic distributed sdn controller[C]. The Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, Hong Kong, 2013: 7-12.
  - [7] Sezer S, Scott-Hayward S, Chouhan P K, *et al.* Are we ready for SDN? Implementation challenges for software-defined networks[J]. *IEEE Communications Magazine*, 2013, 51(7): 36-43.
  - [8] Levin D, Wundsam A, Heller B, *et al.* Logically centralized: state distribution trade-offs in software defined networks[C]. Proceedings of the First ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, New York, 2012: 1-6.
  - [9] Heller B, Sherwood R, and McKeown N. The controller placement problem[C]. The First ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, New York, 2012: 7-12.
  - [10] Hu Y, Wendong W, Gong X, *et al.* Reliability-aware controller placement for Software-Defined Networks[C]. 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), Sophia Antipolis, 2013: 672-675.
  - [11] Hock D, Hartmann M, Gebert S, *et al.* Pareto-optimal resilient controller placement in SDN-based core networks[C]. IEEE 2013 25th International Teletraffic Congress (ITC), California, 2013: 1-9.
  - [12] 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 (EWSDN), Berlin, 2013: 68-74.
  - [13] Kurose J F and Ross K W. Computer Networking: a Top-Down Approach Featuring the Internet[M]. India: Pearson Education, 2013: 35-44.
  - [14] Boyd S P and Vandenberghe L. Convex Optimization[M]. Cambridge: Cambridge University Press, 2004: 43-46, 127-152.
  - [15] Zadeh L A, Klir G J, and Yuan B. Fuzzy Sets, Fuzzy Logic, Fuzzy Systems[M]. Singapore, World Scientific Press, 1996: 165-175.
  - [16] Huang D, Yang D, Zhang H, *et al.* Energy-aware virtual machine placement in data centers[C]. 2012 IEEE Global Communications Conference (GLOBECOM), California, 2012: 3243-3249.
  - [17] Christian raack. Sndlib [OL]. <http://sndlib.zib.de/home.action>, 2013.12.
  - [18] Spring N, Mahajan R, and Wetherall D. Measuring ISP topologies with rocketfuel[J]. *ACM SIGCOMM Computer Communication Review*, 2002, 32(4): 133-145.
- 姚琳元: 男, 1988年生, 博士生, 研究方向为下一代互联网网络层关键技术。
- 陈颖: 女, 1987年生, 硕士生, 研究方向为下一代互联网网络层关键技术。
- 宋飞: 男, 1983年生, 副教授, 硕士生导师, 研究方向为下一代互联网网络层、传输层关键技术。
- 张宏科: 男, 1957年生, 教授, 博士生导师, 研究方向为下一代互联网架构、协议理论与技术、移动互联网路由、传感器网络技术等。