

基于多领域分布式 SDN 控制器的故障移除研究

骆冬冬, 束永安

(安徽大学 计算机科学与技术学院, 安徽 合肥 230601)

摘 要: 针对多领域分布式 SDN 控制器的故障移除问题, 提出了 FMMD(多领域分布式 SDN 故障转移机制)故障移除方法, 主要解决了由于控制器故障而产生的独立交换机重新加入网络而存在的两个问题: 第一、独立交换机如何快速地重新加入到网络中(如何快速地恢复整个网络的通信); 第二、如何防止由于新的交换机的加入而使控制器产生过载. 通过实验表明, FMMD 故障移除方法能够有效降低故障转移时间和端到端平均延迟时间, 还有效提高了 SDN 网络的可扩展性和弹性性能.

关键词: 软件定义网络; 多领域分布式 SDN 控制器; 故障移除

中图分类号: TP393

文献标识码: A

文章编号: 1000—7180(2017)05—0079—04

Failover Based on Multi-Domain Distributed SDN Controller Research

LUO Dong-dong, SHU Yong-an

(College of Computer Science and Technology, Anhui University, Hefei 230601, China)

Abstract: The method of FMMD(Failover Mechanisms for Multi-domain Distributed SDN Controllers) Failover is proposed in this paper to solve the problem of fault removal for multi-domain distributed SDN controller. The two problems that independent switch rejoin in the network resulting from a failure of the controller are mainly solved. First, how to quickly rejoin the independent switch to the network (how to quickly restore the communication of whole network); Second, how to prevent the overload of controller due to the addition of new switches. Experiment shows that the method of FMMD Failover can effectively reduce the time of fault transfer and the average delay time of the end to end, and the scalability and flexibility of the SDN network are effectively improved.

Key words: Software-Defined Network (SDN); Multi-domain distributed controllers; Failover

1 引言

随着互联网的高速发展, 互联网已经成为世界上基础设施之一. 伴随着云计算的出现, 物联网的发展和无线网络的普及, 基于 TCP/IP 协议的传统网络架构已经很难满足这些技术的要求了. 由此 SDN 应运而生.

最初对 SDN 的研究中, 主要思想是单一的控制 器集中管理网络, 而单一集中控制器架构突出的缺 点是可扩展性差和单一节点问题. 而 SDN 提出的主 要设想是将其应用于 Wide Area Network(WAN)

和覆盖网络中, 这样的情况下就要求网络的本身具 有很强的可扩展性和弹性. 在此情况多领域分布式 SDN 控制器随即被提出, 而在多领域分布式 SDN 控制器上处理控制器故障问题, 也就成为研究者的 一个热门的研究方向.

多领域分布式 SDN 控制器就是指是一个在逻辑 上集中, 物理上分布的一种新型的 SDN 控制器架 构^[1]. 它将整个网络分成多个域, 而每个领域内可以 有一个或者多个控制器控制该域内交换机的资源转 发和接收的编排, 维护和拓展网络拓扑等.

如图 1 所示, K. Phemius, M. Bouet 和 J. Le-

guay^[2]提出了多领域分布式的 SDN 网络架构,它的每个域一个控制器,每一个控制器可以控制多个交换机通信.控制器与控制器通过 AMQP 协议进行连接,而交换机与交换机是通过 LLDP 协议进行连接.

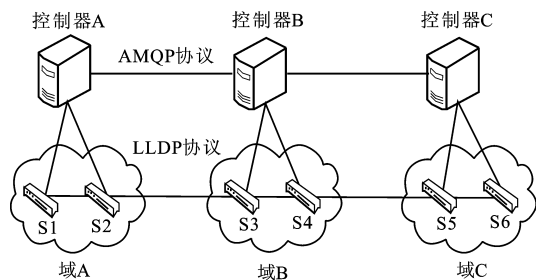


图1 多领域的 SDN 结构图

2 多领域分布式 SDN 控制器故障相关工作

针对单一集中式 SDN 控制结构的可扩展性和弹性不足,以及单一节点故障问题,文献[3]提出在网络中存放了一些备用的控制器.如果主控制器发生故障,则备用控制器将接管主控制器.文献[4]等人提出 DC 方法,提前对控制器的优先级随机定义,如果一个控制器崩溃,它将选择优先级最高的接管其工作.在 HyperFlow^[5]中任意选择一个控制器更换故障控制器.文献[6]提出分布式 sdn 控制器的故障转移,在每一个交换机中放一个该交换机相邻领域的控制器列表,当控制交换机的控制器发生故障时,通过 PPF 算法选择相应的控制器,再加入其中. Yi-Chen Chan 等人^[7]通过 EMA(及指数平均指标)公式来估计超时延时从而对故障检测时间进行改善,在对故障转移的方法主要的思想是在每一个领域内除了拥有一个 Master 控制器外,还要增加一个 slave 控制器.它对故障的处理有三种方法:(1)当 Master 控制器发生故障时由 Slave 控制器接管;(2)当 Master 和 Slave 控制器都发生故障时,则在网络中选择 Master 和 Slave 控制器都健康的领域来接管;(3)当整个网络中没有 Master 和 Slave 控制器是健康的,则选择最近的 Slave 控制器接管.

在以上文献中,在控制器发生故障时,故障控制器下的交换机重新分配到适合的控制器中,都存在一些问题.

3 多领域分布式 SDN 故障快速转移机制 (FMMD)

3.1 FMMD 方法的提出

为了使网络中控制器的资源可以得到充分的利

用,根据 OpenFlow1.3.0 协议^[8]提出交换机的主从控制器的概念,即一个交换机只能有一个 Master 控制器,但是可以有多个 Slave 控制器,Master 控制器可以访问交换机中的所有信息,并且可以接收异步信息,而 Slave 控制器只能读取交换机的信息,而不能接收异步信息.

在故障转移的研究中,最主要的研究方向是基于多领域分布式 SDN 控制器,以及如何将因故障而产生的独立的交换机在最短的时间内重新加入到网络中.所以我们可以以本领域内的控制器为 Master 控制器,而相邻域的控制器都作为该控制器下交换机的 Slave 控制器.如图 2 所示,相对于 A 领域而言,控制器 A 为 A 领域下交换机的 Master 控制器,而 B 和 C 控制器都可以成为 A 领域下交换机的 Slave 控制器.当领域中的 Master 控制器发生故障时,则要选择合适的 Slave 控制器来代替 Master 控制器.例如:交换机 S_1 的 Master 控制器是控制器 A,而控制器 B 和控制器 C 则是交换机 S_1 的 Slave 控制器.当控制器 A 发生故障时,控制器 A 下的交换机 S_1 通过下文的算法在控制器 B 和控制器 C 中选择承受代价最小的,再将交换机 S_1 加入到相应的控制器中.

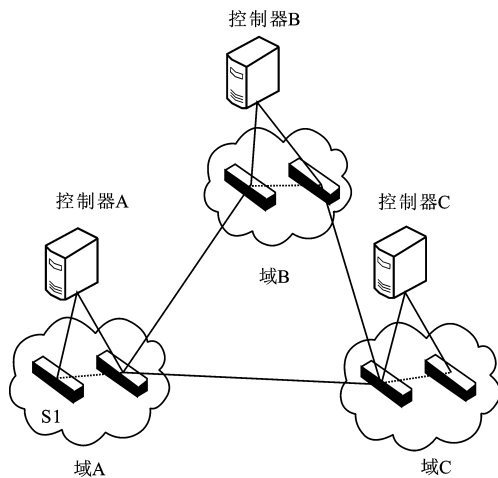


图2 分布式 SDN 框架图

3.2 选择 Master 控制器

在一个 SDN 网络中,Slave 控制器代替 Master 控制器是本文论述的重点.

首先,要从相邻域中确定各个控制器的优先级,可以通过式(1)来确定,这里的 $d_{i,j}$ 是交换机 S_i 到控制器 C_j 的距离; τ_i 是交换机 S_i 发送个故障交换机的 packet-in 信息数量; $Cost_{i,j}$ 是交换机 S_i 到控制器 C_j 所承受的代价.在这里可以认为 $Cost_{i,j}$ 越小则该控

制器的优先级就越高^[7].

Cost_{*i,j*} = *d*_{*i,j*} * τ_i ;

(1)

其次,还要考虑的是出现故障控制器下的交换机在加入相邻域控制器后,相邻域控制器会不会出现过载状况. 在这里我们假定控制器 *j* 的最大控制能力用 *C_{mj}* 来表示;控制器 *j* 当前的负载是 *P_j*;Δ*C_j* 则是控制器 *j* 剩余控制力;*K_i* 表示的是交换机 *S_i* 所需要控制器的控制能力. 通过式(2),(3) 可知当新的交换机加入控制器时,它对控制器的控制要求不能超过 Δ*C*. *C_j* 是交换机 *S_i* 所需要加入的控制器 *j*.

Δ*C_j* = *C_{mj}* − *P_j*;

(2)

K_i ≤ Δ*C_j*;

(3)

在这里我们假定控制器有 *n* 个相邻域,*D* 集合表示的是出现故障控制器的相邻域的集合,*D* = {*d*₁,*d*₂,*d*₃,*d*₄⋯,*d_n*},这里的域就是一个控制器所控制的范围;*C* 表示相邻域的控制器集合,*C* = {*c*₁,*c*₂,*c*₃,⋯,*c_n*};*S* 表示出现故障控制器的交换机的集合,*S* = {*s*₁,*s*₂,*s*₃⋯,*s_m*}. 符号描述如表 1 所示.

表 1 符号描述

符号	符号描述
<i>s_i</i>	交换机 <i>i</i>
<i>c_j</i>	交换机 <i>S_i</i> 所需要加入的控制器 <i>j</i>
τ_i	交换机 <i>S_i</i> 发送个故障交换机的 packet-in 信息数量
Cost _{<i>i,j</i>}	交换机 <i>S_i</i> 到控制器 <i>C_j</i> 所承受的代价
<i>C_{mj}</i>	控制器 <i>j</i> 的最大控制能力
<i>P_j</i>	控制器 <i>j</i> 当前的负载
Δ <i>C_j</i>	控制器 <i>j</i> 剩余控制力
<i>K_i</i>	交换机 <i>S_i</i> 所需要控制器的控制能力
<i>n</i>	控制器相邻域的个数
<i>D</i>	出现故障控制器的相邻域的集合
<i>C</i>	相邻域的控制器集合
<i>S</i>	出现故障控制器的交换机的集合

算法步骤如下所示.

步骤 1:对 *S* 的集合进行循环;

步骤 2:计算出 Cost_{*i,1*} 的值,并将该值赋给 min;

步骤 3:循环相邻域的控制器集合;

步骤 4: 如果 min > Cost_{*i,j*}∧*K_i* ≤ Δ*C_j*, 则将 Cost_{*i,j*} 的值赋给 min,并把控制器标号记录下来;否则,计算下一个控制器的 Cost 的值;

步骤 5:计算完所有的控制器后,则将交换机 *S_i* 加入到控制器 *C_a* 中,同时将交换机 *S_i* 从交换机集合 *S* 中移除;

Input:*K_i* 交换机 *i* 所需的控制能力;

P_j 控制器 *j* 当前的负载;

C_{mj} 控制器的最大控制能力;

Initialization:min = Cost_{*i,1*} 表示距离代价最短值;

While(*S*)

{

Cost_{*i,1*} = *d*_{*i,1*} * τ_i ;

Min = Cost_{*i,1*};

for(*j* = 0;*j* < *C*.max;*j*++)

{

Cost_{*i,j*} = *d*_{*i,j*} * τ_i ;

If(min > Cost_{*i,j*}∧*K_i* ≤ Δ*C_j*)

{

min = Cost_{*i,j*};

a = *j*;// 记录最小代价的控制的标号

}

}

则将交换机 *S_i* 加入到控制器 *C_a* 中,同时将交换机 *S_i* 从交换机集合 *S* 中移除;

}

4 实验及评估

4.1 仿真场景描述

在这里我们使用 Floodlight OpenFlow controller^[9]为控制平面和使用 EstiNet OpenFlow network simulator^[10]来实现数据平面,链路宽带在这里我们设计为 10 Mb/s,利用 mininet^[11]进行仿真,实验环境为 64 位 12. 04 版本 Ubuntu 虚拟机,OpenFlow 交换机. 模拟实验工具如表 2 所示.

表 2 模拟实验工具

参数	实验工具
模拟仿真器	mininet
实验环境	64 位 12. 04 版本 Ubuntu 虚拟机
控制平面	Floodlight OpenFlow controller
数据平面	EstiNet OpenFlow network simulator
链路宽带	10 Mb/s
交换机	OpenFlow 交换机
域的数目	3
交换机数目	31

如图 3 所示,我们所设计的 SDN 网络为 3 个域分别是 A、B、C,每个域内有一个控制器,分别给控制器 A、B、C 配置不同数量的交换机,在这里我们假设给 A 领域配置 9 个交换机,B 领域配置 10 个交换机,C 领域配置 12 个交换机. 对于控制器的控制能力本文的实验通过控制交换机的数目来实现,在此我们假设控制器的最大控制能力是 17 个交换机,同

时每个交换机所消耗控制力为 1. 并且我们假设有 50% 的流是从域 A 发出的, 40% 的流是从域 B 发出的, 还有 10% 的流是从域 C 发出的. 源地址、目的地址、开始时间和持续时间所产生的流都是随机的. 而实验部分和 FCF-M^[7] 主要的的对比数据是故障转移时间和端到端的平均延时时间.

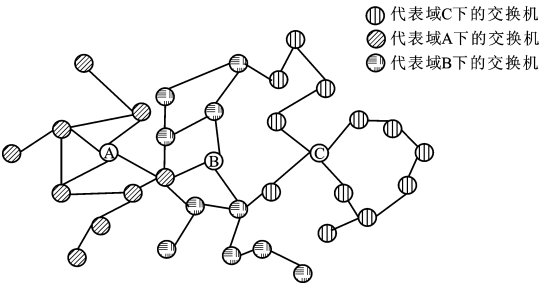


图 3 网络拓扑图

4.2 实验结果

如图 4 所示, 实验主要描述的是故障转移时间, 由图可知 FMMD 方法的故障转移时间与 FCF-M 进行比较, 从图中可以得出 FMMD 的方法相比较于 FCF-M2 和 FCF-M3 在故障转移时间分别提高了约 10 ms 和 40 ms. 图 5 表示的是 FMMD 与 FCF-M 的端到端的平均延迟时间的对比图, 从图 5 中可以看出 FMMD 相对于 FCF-M2 和 FCF-M3 在端到端的平均延迟时间上分别提高了大约 30 ms 和 90 ms.

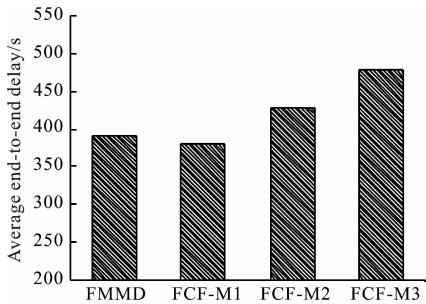


图 4 故障转移时间

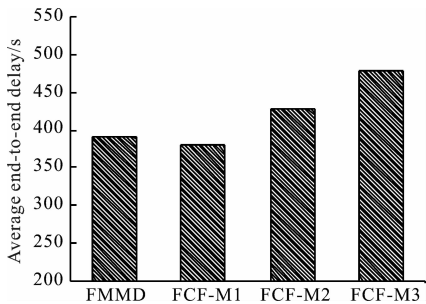


图 5 端到端的平均延迟时间

文献[7]提出的 FCF-M 对故障的处理有三种方法: (1) 当 Master 控制器发生故障时由 Slave 控制器接管; (2) 当 Master 和 Slave 控制器都发生故障时, 则在网络中选择 Master 和 Slave 控制器都健康的领域来接管; (3) 当整个网络中没有 Master 和 Slave 控制器都健康的, 则选择最近的 Slave 控制器接管. 第一种方法(图 4 和图 5 中的 FCF-M1)是交换机通过域内的 Slave 控制器直接接管, 所以它在故障转移时间和端到端的平均延时上要比 FMMD 方法少, 但要跨域接管交换机时, 我们发现, FMMD 相比于 FCF-M2 和 FCF-M3 在故障转移时间和端到端的平均延时上都要少. 同时 FCF-M 方法还在每个域内添加 Slave 控制器, 这样对网络的投入成本相对较高.

5 结束语

通过以上的对比实验, FMMD 方法降低了多领域分布式 SDN 控制器的故障转移时间和端到端的延迟时间, 提高了分布式 SDN 网络的可扩展性和弹性. 同时网络中控制器的利用率也得到了提升, 降低了投入成本.

参考文献:

[1] Tboomas D Nadeau, Ken Gray . 软件定义网络 SDN 与 OpenFlow 解析[M]. 北京: 人民邮电出版社, 2014.

[2] Phemius K, Bouet M, Leguay L. DISCO: Distributed multidomain SDN controllers[C]// Network Operations and Management Symposium (NOMS), 2014 IEEE, IEEE, 2014.

[3] Fonseca P, Bennesby R, Mota E, et al. A replication component for resilient OpenFlow-based networking [C]// Proc. IEEE Network Operations and Management Symposium, April 2012. IEEE, 2012.

[4] Tam A W, Xi K, Chao H. Scalability and resilience in data center networks: Dynamic flow reroute as an Example [C] // Proc. IEEE Global Telecommunications Conference, 2011. IEEE, 2011.

[5] Tootoonchian A, Ganjali Y. HyperFlow: a distributed control plane for OpenFlow, [C]// Proc. ACM Internet Network Management Conference on Research on Enterprise Networking, April 2010. IEEE, 2010.

[6] Phemius K, Bouet M, Leguay J. Failover mechanisms for distributed SDN controllers . network of the future (NOF) [C]// 2014 International Conference and Workshop on the Network of the Future (NOF). IEEE, 2014.

- 2215-2220.
- [2] Swaminathan A, Ramapackiyam S S K. Edge detection for illumination varying images using wavelet similarity [J]. *Let Image Processing*, 2014, 8(5):261-268.
- [3] 张志龙, 杨卫平, 李吉成. 一种基于蚁群优化的显著边缘检测算法 [J]. *电子与信息学报*, 2014, 36(9): 2061-2067.
- [4] 李俊山, 马颖, 赵方舟. 改进的 Canny 图像边缘检测算法 [J]. *光子学报*, 2011, 40(1):50-54.
- [5] Marjan Kuchaki Rafsanjani, Zahra Asghari Varzaneh. Edge detection in digital images using Ant Colony Optimization [J]. *Computer Science Journal of Moldova*, 2015, 23(3):343-360.
- [6] 李敏. 高分辨率合成孔径雷达图像高速公路检测法 [J]. *计算机应用*, 2011, 31(7):1825-1828.
- [7] 高翔, 梁志伟, 徐国政. 基于 Hough 空间的移动机器人全局定位算法 [J]. *电子测量与仪器学报*, 2012, 26(6):484-490.
- [8] 周激流, 詹晓倩, 何坤. 基于统计估计的图像边缘检测 [J]. *沈阳工业大学学报*, 2011, 32(6):665-671.
- [9] 张健, 何坤, 郑秀清. 基于蚁群优化的图像边缘检测算法 [J]. *计算机工程*, 2011, 37(17):191-193.
- [10] 李敏花, 柏 猛. 基于蚁群优化算法的复杂背景图像文字检测方法 [J]. *计算机应用*, 2011, 31(7):1844-1846.
- [11] Shih M Y, Tseng D C. Wavelet-based multiresolution edge detection and tracking [J]. *Image and Vision Computing*, 2015, 23(4):441-451.

作者简介:

邢静宇 女, (1981-), 硕士, 讲师. 研究方向为云计算与虚拟化、计算机图像处理. E-mail: nylgxyzjy@sina.com.

高 需 男, (1982-), 博士, 讲师. 研究方向为时空数据查询处理与优化、数据挖掘.

韩 璞 男, (1985-), 硕士, 讲师. 研究方向为数据挖掘、数字图像处理.

(上接第 82 页)

- [7] Chan Yichen, Wang Kuochen, Hsu Yi Huai. Fast controller failover for multi-domain software-defined networks [C]//IEEE Conference Publications, 2015. IEEE, 2015.
- [8] Opennetworking. OpenFlow switch specification v1. 3. 0 [EB/OL]. [2016-01-09]. <https://www.opennetworking.org/images/stories/downloads/specification/openflow-spec-v1.3.0.pdf>.
- [9] Projectfloodlight. Floodlight OpenFlow controller [EB/OL]. [2016-01-09]. <http://www.projectfloodlight.org/floodlight>.
- [10] EstineEstiNet. OpenFlow network simulator [EB/OL]. [2016-01-09]. <http://www.estinet.com>.
- [11] Oliveira R L S, Shinoda A A, Schweitzer C M, et al. Using mininet for emulation and prototyping software-defined networks [C]//Communications and Computing (COLCOM), 2014 IEEE Colombian Conference on. IEEE, 2014: 1-6.

作者简介:

骆冬冬 男, (1990-), 硕士研究生. 研究方向为 SDN(软件定义网络). E-mail: 1262784556@qq.com.

束永安 男, (1964-), 博士研究生, 教授. 研究方向为无线网络、下一代网络体系结构、SDN(软件定义网络).