

# Raft 算法在 SPTN 分布式控制器集群中的应用

徐姝瑶, 陈 辉\*

(桂林电子科技大学 信息与通信学院, 广西 桂林 541004)

**摘要:**为提高软件定义分组传送网(SPTN)信息传输的灵活性和可用性,在软件定义网络(SDN)构架下引入了集中管控的 SDN 控制器作为 SPTN 的核心,实现控制转发分离。通过使用 Raft 算法在 SPTN 中集群部署,使多个 SDN 控制器协同工作,解决了单一的控制节点在可靠性、扩展性方面的问题。SDN 控制器集群利用 ETCD 数据库实现了 Raft 算法的选举、控制器主备切换、数据同步和持久化等功能。实验结果表明:Raft 算法在 SDN 控制器集群中可以正常应用,并且在网络业务便捷开通和网络信息传输的灵活性和可靠性等方面性能有所提升。

**关键词:**Raft 算法;软件定义网络控制器;软件定义分组传送网

中图分类号:TN914 文献标志码:A 文章编号:1002-5561(2021)11-0031-05

DOI:10.13921/j.cnki.issn1002-5561.2021.11.006

开放科学(资源服务)标识码(OSID):



## Application of Raft algorithm in SPTN distributed controller cluster

XU Shuyao, CHEN Hui\*

(School of Information and Communications, Guilin University of Electronic Technology, Guilin Guangxi 541004, China)

**Abstract:** In order to improve the flexibility and availability of information transmission in the software defined packet transport network (SPTN), a centralized control software defined network (SDN) controller is introduced as the core of the SPTN under the framework of the SDN to realize the separation of control and forwarding. This paper studies the cluster deployment in SPTN by using the Raft algorithm to make multiple controllers work together to solve the reliability and scalability problems of a single controller node. The SDN controller cluster uses the ETCD database to realize the functions of Raft algorithm election, controller master/backup switching, data synchronization and persistence. The experimental results show that the Raft algorithm can be used normally in the SDN controller cluster, and its performance has been improved in terms of convenient network service opening, flexibility and reliability of network information transmission.

**Key words:** Raft algorithm; software defined network controller; software defined packet transport network

### 0 引言

随着 5G 时代的到来,传送网管控系统和上层业务协同编排的交互联动需求更加明确,协同管控、网络切片管控和智能运维成为新的研究热点,软件定义网络(SDN)控制器架构<sup>[1-2]</sup>的出现对未来网络的发展起

着至关重要的作用。为了保证 SDN 工作的逻辑策略及其它重要元素都集中在控制层<sup>[3]</sup>,控制器成为提高 SDN 可扩展性的关键。使多控制器之间保持一致性的方法是在多控制器集群节点上运行一致性算法。Paxos 算法是目前应用最广泛、解决分布式一致性问题最有效的算法之一,但是该算法难于理解,在实际中难以实现。因此,本文设计在软件定义分组传送网(SPTN)中集群部署 SDN 控制器,并使用分布式一致性算法——Raft 算法<sup>[4-5]</sup>,以确保分布式控制器节点之间关键数据复制的强一致性,提高网络的可用性和稳定性<sup>[6]</sup>。

### 1 SDN 控制器设计与实现

本文设计的 SDN 控制器提供资源全局分配、路径动态规划、流量集中管控和业务自动编排等功能。逻辑集中的控制平面收集全网资源状态信息,维护全局

收稿日期:2021-02-03。

基金项目:广西无线宽带通信与信号处理重点实验室基金项目(GXKL0614104)资助。

作者简介:徐姝瑶(1996—),女,河北石家庄人,硕士研究生,现就读于桂林电子科技大学信息与通信学院电子与通信工程专业,主要研究方向为光通信技术应用、SDN 智能分组传输等,参加过国际光学会议和广西省科研项目。

\*通信作者:陈辉(1976—),男,博士,副教授。主要研究方向为光通信、激光成像和复杂环境下电波传播特性。



徐姝瑶,陈辉:Raft 算法在 SPTN 分布式控制器集群中的应用

网络视图,极大地简化了上层应用的开发。SDN 控制器组成结构如图 1 所示。控制器是部署在整个网络设备之外的软件框架和部件,对数据平面的硬件转发进行控制,同时向更上层的应用提供服务。SDN 控制器主要由管理控制核心、集群同步、北向接口和南向接口组成。其中,北向接口采用 REST API 接口,方便对接网络管理软件;通过高效的 gRPC 南向编程接口控制 SPTN 转发设备,实现网络管理和业务管理等功能;东西向同步的集群模块使用 Raft 算法保证各个控制器上的数据的一致性,通过选举主控制器和备控制器可以解决因网络状态改变导致的网络故障,比如节点故障、链路故障和网络拥塞等。

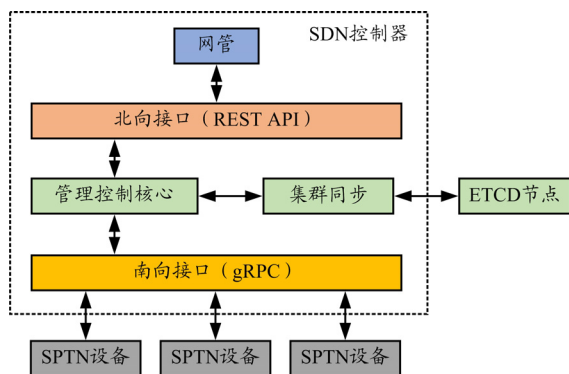


图 1 SDN 控制器组成图

SDN 控制器集群管理的主要思想是将网络分为多个不同的区域,根据每个区域的大小和流量的多少部署一个或多个 SDN 控制器管理区域内的交换机,网络内的所有控制器通过保持整个网络的网络状态信息的一致性来实现对整个网络的协调管理。

网络管理软件通过控制器向 SPTN 转发设备下发业务意图,SDN 控制器进行相应的业务计算并实时写入 ETCD 数据库。SDN 控制器管理控制核心包括设备管理、拓扑管理、标签管理、带宽预留、隧道(LSP)管理、重路由、伪线(PW)管理、业务管理、配置项管理、告警管理和性能统计等功能。

本文旨在分布式控制器环境下,采用 Raft 算法和 ETCD 数据库来实现控制器节点数据一致性,利用 SDN 控制器提高整个网络拓扑的业务开通性能和抗毁性能。

## 2 Raft 算法原理

Raft 算法是一种基于日志复制管理的一致性算法,能解决分布式环境下多个副本间数据一致性问题,提高分布式系统的可靠性和稳定性。Raft 算法的主

要组成包括领导者选举、日志复制、日志压缩、快照复制和数据读写等内容,本文只介绍重点内容——领导者选举和日志复制。

### 2.1 领导者选举

ETCD 数据库是一个分布式 key/value 存储系统,能完整地实现 Raft 算法,并做出了一些改进,确保分布式环境下多个副本数据的强一致性,已被业界广泛应用。

ETCD 集群 Raft 算法设计了 3 种集群成员的角色:领导者、跟随者和候选者,这 3 种角色转换情况如图 2 所示。

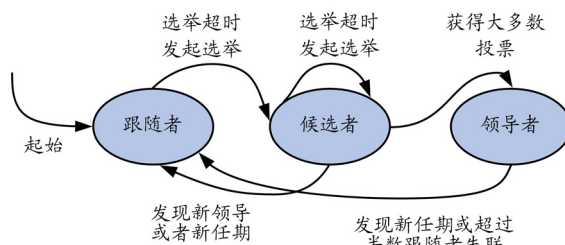


图 2 Raft 算法中选举角色的转换图

Raft 算法实现一致性是通过在集群中选举得到领导者,然后让领导者全权负责管理分布式日志。领导者从客户端接收日志条目,然后将日志条目复制到其它服务器,并告诉服务器何时将日志条目应用到状态机是安全的。

定时器在 Raft 选举过程中起重要的作用,与选举过程相关的 2 个关键定时器如表 1 所示。

表 1 选举定时器

| 定时器类型 | 取值                                                                 | 超时处理                           |
|-------|--------------------------------------------------------------------|--------------------------------|
| 选举超时  | 缺省设置 1000 ms, 为避免同时发起选举, 实际取值在 $[1000, 2 \times 1000 - 1]$ 范围内随机变化 | 选举超时<br>定时器计数<br>清零, 候选者发送投票消息 |
| 心跳超时  | 缺省设置 100 ms                                                        | 领导者发送心跳消息                      |

ETCD 集群 Raft 算法的选举流程如下:

①每个集群节点启动后,直接进入跟随者角色,启动选举超时定时器,每个节点的选举超时定时器值在一个范围内随机选取,避免同时发起选举。

②选举超时定时器最先到期的节点,转换为候选者角色,发起投票选举。

③如果其它节点收到投票消息时,其选举超时定时器还未到期。首先,将选举超时定时器计数清零;其

次,如果投票消息的节点任期大于自己的任期,赞成投票,继续保持跟随者角色,不再给其它相同任期的候选者投票;如果投票消息的节点任期小于自己的任期,拒绝投票;如果没有节点选举成功,则当前任期加一,重新发起选举。

④如果其它节点收到某节点的投票消息时,其选举超时定时器已经到期,也转换成了候选者角色,并且自己以相同的任期发起投票选举,那么拒绝投票给该节点,因为已经投票给自己。

⑤发起投票的候选者收到其它节点的投票响应最先超过半数赞成票(包括自己)的节点胜出,转换为领导者,开始发送日志/快照复制和心跳。

⑥未胜出的候选者收到领导者发送的任意消息后,将选举超时定时器计数清零,跟随新的领导者。

⑦如果集群参与节点数量是偶数,并且刚好出现平票,那么本轮选举失败,等到下一个选举超时定时器最先到期的节点增加任期,重新发起投票选举。

## 2.2 日志复制

日志是 Raft 算法的核心。Raft 算法通过日志同步的方式实现数据的一致性;同时,日志也是影响 Raft 算法选举结果的重要因素。ETCD 日志复制是指通过 Raft 算法实现日志的集群同步,该过程是一个连续的过程,领导者需要时刻跟踪所有跟随者的日志同步状态和跟随者通过多次的消息交互,才能实现日志和数据的最终同步。

日志复制从一个任期的领导者选举完成后开始,在任期内领导者接收到用户的请求命令也伴随着日志复制的过程,任期内有新的成员加入集群也会触发日志复制过程。当领导者失去连接,所有跟随者日志复制过程的进度直接影响下一任期内新领导者的选举结果。

## 3 SDN 控制器集群主备模式的实现方法

### 3.1 控制器选举

控制器集群采用一主多备模式,只有一台主控制器工作,其它控制器作为候选者备份。在一个网络分区内,所有参与的控制器通过 Raft 算法选举过程选出一个主控制器。主控制器将用户下发的静态业务配置和动态计算出的隧道配置通过 ETCD 数据库实时同步到所有备控制器,备控制器保持和主控制器完全相同的数据状态运行。

当主控制器掉线后,选举出新的主控制器,完成主备同步流程;恢复配置数据,开启配置下发功能和

重路由功能,并继续通过北向接口提供数据;同时,新的主控制器需要检测转发路径的有效性,如果有转发路径失效,立即触发一次重路由,优化切换前拓扑变化造成的转发路径切换和失效,避免了由于主节点失效造成的网络瘫痪等问题。

主控制器的选举采用 ETCD 集群 Raft 算法的领导者选举结果,控制器节点和 ETCD 节点按照 1:1 数量部署,如图 3 所示。控制器需要周期性地监听 ETCD 集群的领导者选举结果状态,一旦本地 ETCD 节点的领导权发生了变化,控制器的主备切换立刻执行。设备和链路等拓扑信息不通过 ETCD 数据库同步,所有的备控制器直接通过链路状态路由协议(LSRP)获取网络拓扑信息,并维护设备在线状态和链路在线状态。

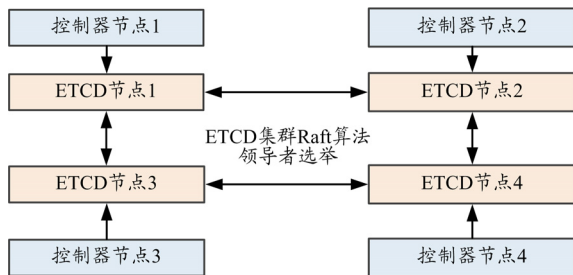


图3 主控制器选举示意图

### 3.2 控制器数据同步

控制器同步的数据是指控制器部分核心运行状态和下发到 SPTN 设备的配置项,其中业务相关的数据包括 vpls、pw、pw 保护组、service policy 和 filterclass;隧道相关的数据包括 lsp、lsp 保护组和 oam。

控制器主备选举完成后,主控制器对接网络管理系统,负责网络业务维护和网络状态呈现。当主控制器失效后,新选举出的主控制器接替网络管理职责,因此,新的主控制器必须同步旧的主控制器的完整数据才能继续正确运行。

主备控制器的数据同步采用基于 Raft 算法的 ETCD。Raft 算法使用大多数原则,只有超过半数的节点认可数据提交后,才认为数据提交成功,因此充分保证了分布式系统各个节点数据的一致性。多个控制器和多个 ETCD 均采用分布式部署,每个节点包含一个控制器和一个 ETCD。ETCD 作为控制器数据同步持久化平台,以 key/value 的数据表达方式增加、删除数据,同时采用 Raft 算法把数据同步到对端控制器。

Raft 算法写数据流程如图 4 所示,具体过程如下:节点 1 主控制器向本节点的 ETCD 发起写入请求,由于本节点 ETCD 不是领导者而是跟随者,所以将请求

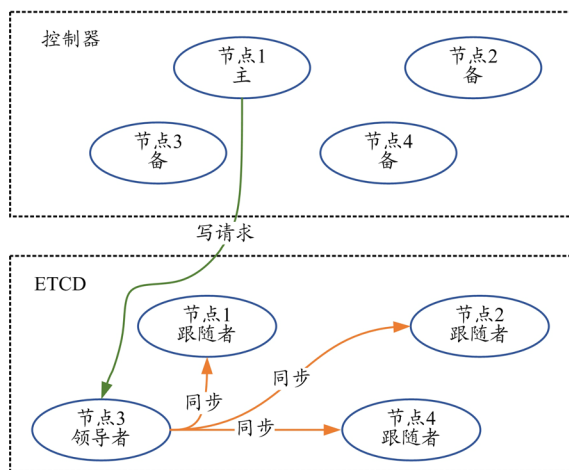


图4 Raft 算法写数据流程

转发给节点3的ETCD;节点3的ETCD收到请求,开始Raft算法同步过程,超过半数节点确认后,数据提交成功;节点1的ETCD数据提交成功后,返回结果给节点1主控制器。

Raft算法读数据流程如图5所示,具体过程如下:节点1作为新主控制器发起读请求,节点1的ETCD收到读请求,如果是本地读,则直接进入本地数据库读,返回结果给节点1主控制器;如果是一致性读,则进入Raft模块,向节点3ETCD发出读索引请求,获取当前ETCD集群已提交的最新数据;节点3ETCD收到读索引请求后,向节点1ETCD同步该索引的最新数据;节点1ETCD收到数据更新后,提交到本地数据库,并返回最新的数据给节点1主控制器。

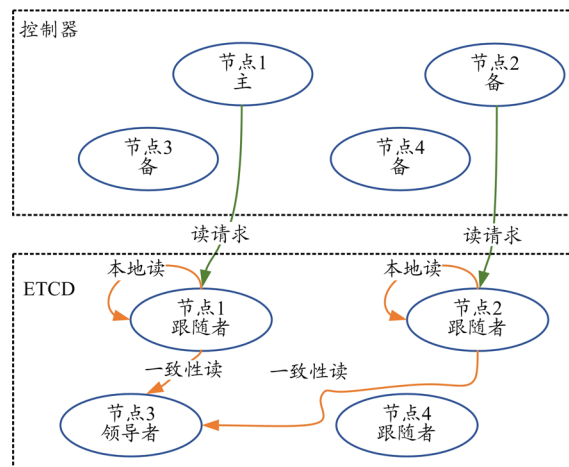


图5 Raft 算法读数据流程

ETCD的数据同步是实时热备份的,即一旦有新的数据产生,主控制器会立即向备控制器同步。控制器的数据同步通过ETCD实现,可以实时热备份,即控制器实时监听ETCD的数据同步事件并立即完成控制

器自身的数据恢复。正常情况下,主控制器只需要从ETCD数据库读数据,因为Raft算法是强一致性的和最终一致性的,可以保证各个节点的数据是相同的。

### 3.3 控制器主备切换

若主控制器发生网络故障掉电,即与主控制器1:1部署的ETCD节点也相应掉电,此时发生主备切换。剩余工作的ETCD节点重新进行Raft算法选举,选出新的ETCD主节点,对应部署的控制器也相应切换成主控制器。由于ETCD数据库实时热备份同步数据,剩余工作的ETCD数据库里都有完整记录之前掉电的主控制器计算的业务数据,此时只需要新的主控制器从ETCD数据库里读取业务数据,再与SPTN设备南北对账,按需补充虚拟专用局域网业务(VPLS)业务数据的配置,并同步给其它备控制器。主备切换涉及到控制器利用Raft算法从ETCD读数据和向ETCD写入新数据的过程。

### 3.4 控制器数据持久化

数据持久化是指通过ETCD永久保存控制器的配置和状态数据。当全网设备包括控制器全部重启时,业务数据可以恢复之前的状态,不用每次重启控制器都需要去重新创建业务。在控制器运行过程中,主控制器将数据写入ETCD进行持久化存储。当所有控制器掉电重启后,选举出主控制器,由主控制器从ETCD恢复出重启之前的所有配置数据和状态数据;同时,控制器发生主备切换后,如果新的主控制器没有数据,也需要从ETCD恢复数据。数据恢复后,主控制器经过隧道数据有效性校验,再通过南向对账,确认设备配置是否齐全,并按需下发配置给所有设备。

## 4 系统测试

相较于传统的同步数字系列(SDH)光传输设备,本文提出的SDN控制器应用在SPTN中,在业务开通、业务增删、整网主备切换和整网抗毁性能等方面都有显著的提升。SDN控制器系统开发使用GO语言,具有开发速度快、可维护性强、稳定性和可移植性较高等特点。CPU使用龙芯3A3000,操作系统为中标麒麟,数据存储数据库使用ETCD3.4版本。

为验证Raft算法在SDN控制器中的正常应用、业务开通和网络性能的提升,本文使用5节点控制器组成的集群,控制10台SPTN转发设备组成的转发网络。网络管理软件下发一条VPLS业务到控制器,控制器进行业务计算,再向SPTN设备下发业务配置项,转发面执行配置项,建立转发路径并转发业务信息。此

表 2 性能测试结果

| 网络结构                    | 业务下发<br>整网时间/s | SDN 控制器<br>选举时间/s | SDN 控制器<br>主备切换时间/s | SDN 控制器<br>同步数据时间/s | SDN 控制器<br>恢复数据时间/s |
|-------------------------|----------------|-------------------|---------------------|---------------------|---------------------|
| 单节点控制器<br>(10 台传送设备)    | 60             | —                 | —                   | —                   | 60                  |
| 3 节点控制器集群<br>(10 台传送设备) | 30             | 1                 | 11                  | 1                   | 60                  |
| 5 节点控制器集群<br>(10 台传送设备) | 30             | 2                 | 12                  | 1                   | 60                  |
| 5 节点控制器集群<br>(20 台传送设备) | 60             | 2                 | 12                  | 2                   | 120                 |

过程完成了一项业务的开通和下发,用时 1 min,相较于传统 SDH 光网络设备开通一项业务需要小时级的时间有了明显的性能提升。

本文模拟真实网络故障,即主控制器节点宕机或备控制器节点宕机。若主控制器宕机立即发生主备切换,重新选举出新的主控制器从旧主控制器对应的 ETCD 恢复数据并同步数据至集群中其它控制器。模拟测试结果如表 2 所示,可以看出,单节点控制器无法应对网络故障,多节点构成分布式集群可提高网络的冗余抗灾能力,且主备切换时间迅速。控制器业务下发时间和同步、恢复数据时间与网络规模有关,即网络中传送设备越多则时间越长。

5 结束语

本文针对 SDN 的功能需求和整体模型提出了适用于 SPTN 的 SDN 控制器的总体架构;对 Raft 算法在 SPTN 中的分布式控制器集群节点应用适配进行研究,实现了主备控制器节点数据一致性;利用控制器节点主备同步机制提高控制器集群的可用性、可扩展

性和分布式冗余抗灾能力。实验结果表明:应用了 Raft 算法的控制器集群在冗余抗灾方面的性能有了很大的提升,可自动选举主控制器并向全网下发业务。

参考文献:

[1] 江国龙,付斌章,陈明宇,等. SDN 控制器的调研和量化分析[J]. 计算机科学与探索,2014,8(6):653-664.

[2] 徐俭.基于控制层的 SDN 网络架构安全可靠技术探究[J]. 广播电视信息,2018(4):61-64.

[3] 邱帆,郑乐,罗广军. 有线、无线混合网络环境下的 SDN 控制器设计与实现[J]. 光通信技术,2019,43(9):13-16.

[4] ONGARO D, OUSTERHOUT J. In search of an understandable consensus algorithm [C]//USENIX. Proceedings of Annual Technical Conference. Stanford: USENIX, 2014: 305-319.

[5] SAKIC E, KELLERERE W. Response time and availability study of RAFT consensus in distributed SDN control plane[J]. IEEE Transactions on Network and Service Management, 2017, 15(1): 304-318.

[6] SUH D, JANG S, HAN S, et al. Toward highly available and scalable software defined networks for service providers [J]. IEEE Communications Magazine, 2017, 55(4): 100-107.