

SDRS: 集中与分布控制相结合的 弹性多路径路由机制

高先明 王宝生 邓文平

(国防科技大学计算机学院 长沙 410073)

摘 要 传统的单播路由协议根据单一路由策略(如最小跳步数、最低代价等)计算出到达目的网络的最短路径。受限于局部视图的不完整性、局部策略的自私性,路由计算结果往往为非最优;同时,数据流传输完全依赖于单条路径,容易产生网络拥塞和由单点故障引起的通信中断。以软件定义网络(SDN)为代表的集中控制架构,通过感知网络状态,从全局视图进行路由计算并下发策略,促使网络路由趋近最优目标。文中提出了软件定义路由系统概念(Software Defined Routing System, SDRS),在分布式路由协议计算的分布路径基础上,将集中控制机理引入到路由控制系统,集中管理器通过最大多路径覆盖树算法计算出无环的集中路径集,并根据编排的策略选择性地吧集中路径部署在网络中,从而实现分布与集中路径同时并行传输数据流,达到二者的有机耦合,既保留了原来分布式路由协议的柔性抗毁、快速自愈的能力,又吸收了集中控制的全局优化能力。最后,实现了 SDRS 原型系统。实验结果表明,在网络拥塞或单点故障时 SDRS 性能明显优于单播路由协议的性能,分布式路由协议的丢包率为 20% 时,SDRS 系统的丢包率仅为 2.94%。

关键词 路由系统;控制机理;分布/集中;柔性抗毁;快速自愈

中图法分类号 TP393

DOI 号 10.11897/SP.J.1016.2018.01976

SDRS: Elastic Multi-Path Routing Mechanism with Integration of Centralization Control and Distribution Control

GAO Xian-Ming WANG Bao-Sheng DENG Wen-Ping

(School of Computer, National University of Defense Technology, Changsha 410073)

Abstract Traditional unicast routing protocols deployed in the Internet just calculate shortest forwarding path for the same destination network by using single routing metric, such as least hops, lowest cost. Even unicast routing protocols can use equal cost multiple protocol (ECMP) to calculate several forwarding paths for one pair of end-to-end communication, they just work well under specific situations with proper network topology and proper metric. Due to incomplete information caused by partial-view of network status and selfishness of local strategy calculated by unicast routing protocols, the routing strategies usually are not optimal. Meanwhile, data-flow transmission completely relies on single forwarding path, leading to network congest and communication interrupt by a single failure (e. g. failure of link, failure of node, etc.). Software defined network(SDN), as the representative of centralized control mechanism, can obtain the global network status, perform routing calculation from global-view, and install routing strategies, hence to promote network toward optimal target. Unfortunately, centralized control architecture generally has bad system robustness and long delay of network status synchronization, thus it

收稿日期:2017-01-29;在线出版日期:2017-07-28. 本课题得到“抵御违反路由策略攻击的可验证安全路由协议的研究”(61472438)资助。

高先明,男,1988年生,博士研究生,主要研究方向为路由协议、可重构网络设备平台。E-mail: nudt_gxm@163.com. 王宝生,男,1971年生,研究员,中国计算机学会(CCF)会员,主要研究领域为计算机网络与通信、网络安全。邓文平,男,1981年生,博士,讲师,主要研究方向为下一代网络体系、路由协议。

hardly has ability of flexible survivability and fast self-healing. Drawing further on this, this paper puts forwards Software Defined Routing System (SDRS), by introducing centralized control mechanism into existing routing system on basis of distributed paths generated by distributed routing protocol. SDRS establishes a closed-loop control system by integrating distributed routing protocol with centralized controller. On one hand, real-time link-status topology from distributed routing system is as the input for centralized controller. On the other hand, routing strategies of centralized paths calculated by centralized controller in return make influence on distributed routing protocol. Besides, centralized controller can calculate centralized path set without routing loops based on Maximum Multi-path Overlay Tree Algorithm, and selectively deploy centralized paths into network according to orchestration strategy. Thus, distributed path and centralized path can simultaneously transmit data-flow in parallel by coupling these two control mechanisms. These two aren't in conflict with each other, because centralized paths generated by centralized controller include distributed paths. SDRS not only preserves the properties of flexible survivability and fast self-healing in distributed routing protocol, but also keeps global optimal ability of centralized controller. Finally, we implement SDRS prototype system, and our experimental results reveal that the SDRS can significantly promote the network performance especially in case of network congest or failure, e. g., when distributed routing protocol has 20% packet-loss-ratio, SDRS just has 2.94% packet-loss-ratio.

Keywords routing system; control mechanism; distribution/centralization; flexible survivability; fast self-healing

1 引言

单播路由协议位于 TCP/IP 模型中网络层,为端到端通信提供可达的传输通道.通过分布式的网络节点同步网络状态,按照预先约定的规则(如最小跳步树、最低代价等)计算出到达目的网络的最短路径,并更新路由策略^[1-2].同时,单播路由协议具有良好的柔性抗毁和快速自愈的能力.尤其在网络出现故障时,分布式路由协议感知到网络状态的改变并迅速将消息洪泛到每个网络节点;由后者进行路由重新收敛过程,并将新的路由策略更新到网络节点中.尽管单播路由协议在现存的 Internet 体系中取得巨大成功,但面临着诸多挑战,主要体现在以下几个方面:(1)局部视图的不完整性,仅能获取到网络链路拓扑信息,而无法感知到网络状态,导致所获取的网络信息是不完整的;(2)局部策略的自私性,仅能按照预先约定的规则计算策略,致使部分节点或链路资源竞争激烈而局部资源处于空闲状态;(3)单条路径传输数据流,不仅局部热点资源竞争激烈导致网络拥塞、报文丢包率增加等问题,而且网络单点故障时路由重收敛会导致端到端通信的中断.

为了解决单播路由协议中单条路径传输数据

流问题,研究学者提出了多路径路由机制,在单播路由协议基础上为端到端通信构建若干的等价或非等价的转发路径,并行或选择性地传输数据流.一些代表性的多路径路由机制被提出,如 ECMP^[3]、OSPF-OMP^[4]、MPA^[5]、DSPA^[6]、CRA^[7]、MPD^[7]、MDVA^[8]、MPATH^[9-10]、MNHR^[10]等.尽管这些方案能很好地解决单条路径传输数据流的问题,但依旧无法解决局部视图的不完整性和局部策略的自私性导致路由计算结果非最佳的问题.与此同时,多路径路由机制中路径计算规则都是预先约定的,难以根据网络应用的场景灵活地调整路由计算的策略.比如,限制路径最大跳步数小于 16,避免报文永远无法到达目的网络;限制到达同一目的网络的多路径数量,避免网络节点维护多路径开销的过大等.

以 SDN(Software Defined Network,软件定义网络)为代表的集中控制架构能在感知网络状态的前提下计算出全局最优策略^[11-14],从而促使网络路由趋于最优目标.同时,集中控制架构要求路由协议开放接口,便于上层应用程序通过调用路由协议的接口实现网络自动化管理^[15-18].这为解决多路径路由机制面临的挑战提供了一种解决思路.但集中控制架构自身存在健壮性较差、网络状态同步延迟大等问题,并不具备分布式路由协议的柔性抗毁、快速

自愈的能力。

本文提出了软件定义路由系统概念(Software Defined Routing System, SDRS),它将集中控制理念引入到分布式路由系统中,并把分布式路由和集中控制管理有机地耦合在一起,构建了一种集中与分布相结合的弹性多路径路由机制。SDRS 由分布式路由协议和集中管理器两类路由计算系统组成,协同计算路由策略。分布式路由协议负责计算最低代价的分布路径,确保网络通信的可达性;集中管理器在分布式路由协议通告的链路拓扑基础上利用最大多路径覆盖树算法计算出集中路径集,并根据编排的策略选择性地把集中路径部署在网络中。最后,搭建了 SDRS 原型系统,在最大多路径覆盖树算法性能测试、路由系统性能、转发策略配置时间以及端到端的传输性能测试等方面验证了 SDRS 可行性和有效性。SDRS 的创新体现在以下几个方面:(1)实现集中与分布控制的耦合,既具备集中控制的全视图、全局策略、开放控制的属性,又保留分布的柔性抗毁的能力。同时,集中与分布控制相互弥补各自不足,集中控制加强了分布路由协议的快速重路由的能力,分布路由协议则弥补集中控制在维护路径方面存在“低效率”的问题;(2)多路径并行传输的能力,数据流由多路径同时并行传输,能有效地避免局部节点或链路资源热点导致通信服务质量降低的问题,降低网络拥塞的程度以及减少网络通信中断的次数;(3)弹性的路由抗毁机制,在保留分布式路由协议的柔性抗毁、快速自愈的能力基础上,利用分布式路由协议负责集中路径的撤销任务。相比较 SDN 网络中多路径方案^[15-16],能避免集中管理器撤销、集中路径效率较低对网络造成严重影响。

本文第 1 节首先指出单播路由协议面临的挑战;针对上述不足,第 2 节提出一种 SDRS 架构,并从网元层面描述集中管理器和网络节点;在此基础上,第 3 节针对集中管理器采取多路径计算的算法——最大多路径覆盖树算法进行了描述;第 4 节描述 SDRS 中路径维护机制以及维护集中与分布路径的主要机理;第 5 节搭建 SDRS 原型系统,验证 SDRS 的可行性与有效性;第 6 节总结本文研究内容以及下一步工作。

2 SDRS 架构

针对单播路由协议有局部视图的不完整性、局部策略的自私性以及单条路径传输数据流的局限性,结合以 SDN 为代表的集中控制理念,提出了软

件定义路由系统 SDRS,将集中控制架构引入到分布式路由系统中,构建了一种集中与分布控制相结合的弹性多路径路由机制,如图 1 所示。

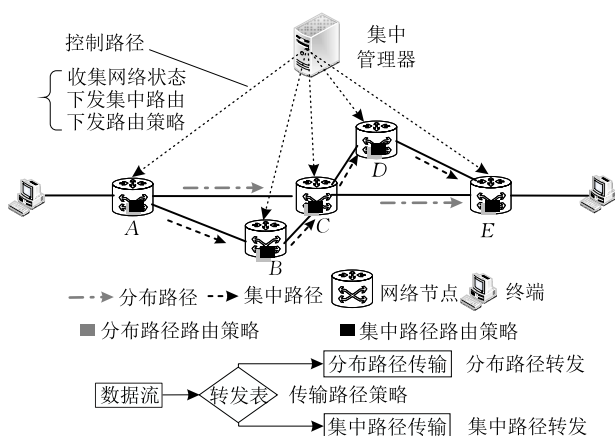


图 1 SDRS 框架

SDRS 由集中管理器和分布式的网络节点组成,集中管理器计算并生成集中路径的路由策略;网络节点通过分布式路由协议计算并生成分布路径的路由策略。两种路径的路由策略共同作用于数据流的传输,使得到达目的网络的转发路径通常由单条分布路径和若干集中路径组成,实现数据流同时并行地沿着多条转发路径进行传输。分布路径和集中路径的计算都是采取相同的 metric 值计算,因而它们具有相同的优化目标和相同角色。集中管理器是多路径路由策略的决策者,基于分布式路由协议感知的链路拓扑信息计算集中路径,将集中路径的路由策略部署到网络中,并由分布式路由协议负责集中路径的撤销任务。同时,集中管理器通过开放控制接口允许对多路径业务功能进行编排(如路径的最大跳步数、集中路径的最大数量等),增加了 SDRS 系统的灵活性和可编程性。分布式路由协议除了计算分布路径外,还承担着将链路拓扑信息通告到集中管理器以及维护集中路径的任务。

SDRS 将分布式路由协议和集中控制架构耦合为闭环控制系统:分布式路由协议集中控制架构的输入,将链路拓扑信息通告到集中管理器,而集中控制架构则施加路由策略作用于分布式路由协议,优化路由系统的性能。从路由机制协作方式来讲,分为自主路由和协同两种方式。自主路由体现在分布式路由协议不需要依赖于集中控制即可以单独地计算路由策略。自主路由为网络传输提供“基础版”的服务模式。协同则体现在集中管理器通过分布式路由协议感知到链路拓扑,并计算出集中路径用于辅助分布式路由协议来优化网络传输性能;同时,分布式

路由协议负责集中路径的撤销任务,并将撤销结果通告到集中管理器.协同路由为网络传输提供“增强版”的服务模式.

集中管理器的引入并不会丧失分布式路由协议的优势,反而增强了路由系统的柔性抗毁、快速自愈的能力,体现在以下两个方面:

(1)集中路径由分布式路由协议负责撤销.由于分布式路由协议能迅速感知到网络拓扑的改变,在集中路径的状态改变或运行的集中路径与新建的分布路径构成路由环路时,必须快速撤销集中路径,避免集中路径对网络造成严重危害.与SDN网络的集中控制理念不同,SDRS充分利用分布式路由协议的优势来弥补集中控制架构在健壮性、可靠性等方面存在的不足.

(2)强化路由系统的柔性抗毁、快速自愈的能力.由于集中管理器计算出集中路径并行传输数据流,在分布路径的状态改变时,存活的集中路径依旧执行数据流传输任务,有效地避免分布式路由协议路由重收敛导致端到端通信中断的问题,从而达到与快速重路由机制相同的目的.

2.1 集中管理器

集中管理器感知链路拓扑信息,从全局视图角度计算出无环的集中路径集,并根据编排路由策略选择性地集中路径部署到网络中,从而促使网络性能趋于最优目标.集中管理器的基本框架包括链路状态数据库、集中路径库、策略编排界面、最大多路径覆盖树算法与外部控制接口等,如图2所示.

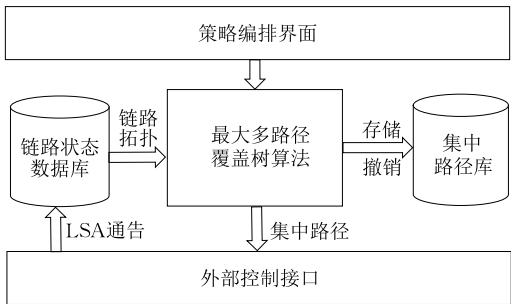


图2 集中管理器

集中管理器通过外部控制接口与分布式路由协议交互信息:一方面,通过外部控制接口从分布式路由协议获取到链路拓扑信息,拥有网络全局信息;另一方面,集中路径的路由策略通过外部控制接口下发至网络节点,从而实现集中路径的部署.集中管理器在通过外部接口接收到链路状态消息时(如LSA^[19]、PDU^[20]等),存储至链路状态数据库中,并触发最大多路径覆盖树算法来计算集中路径路由策略.在集中路径计算完成后,将集中路径的基

本信息(如路径编号、路径经过节点等)存储到集中路径库中.

为了打破现有多路径路由机制的封闭性,通过开放策略编排界面允许配置路由策略,如集中路径的数量、集中路径的跳步数等.根据SDRS系统部署的网络场景灵活地配置多路径路由的计算策略,克服了现有分布式多路径机制的配置复杂、计算规则僵化等不足.例如,SDRS可以配置局部热点节点部署集中路径最大数量,有效地避免局部热点节点成为网络性能的瓶颈;同时,SDRS通过编排多路径业务能力,支持集中管理器不在故障频繁节点部署集中路径,避免路由震荡对网络造成严重危害.

2.2 网络节点

网络节点是路由策略执行体,直接决定报文转发行为.网络节点包括控制平面和数据平面两类功能面.控制平面运行分布式路由协议,主要用于计算分布路径、接收集中路径的路由策略以及维护集中与分布路径;数据平面则根据控制平面下发的转发策略来确定报文转发的下一跳.

(1)控制平面

控制平面主要是分布式路由协议的承载者,负责分布路径的计算、链路拓扑感知与通告以及集中路径撤销等任务.控制平面基本架构包括外部控制模块、服务代理模块、单播路由协议以及内部控制接口等功能模块,如图3所示.

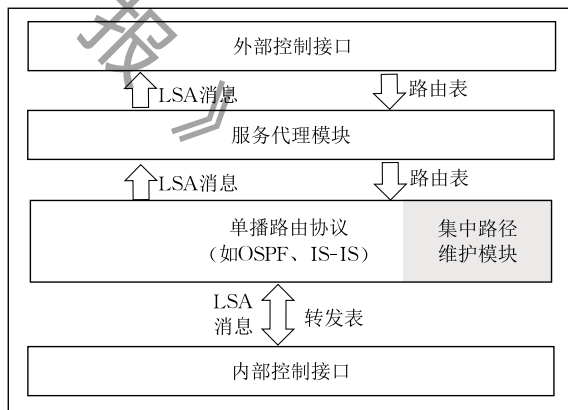


图3 控制平面

控制平面通过外部控制接口与集中管理器交互信息,如链路状态消息、集中路径的路由策略等.同时,控制平面还通过内部控制接口与数据平面通信,包括控制平面将转发表通过内部控制接口转至数据平面以及将路由协议生成控制消息(如链路状态消息LSA、PDU等)通过内部控制接口传递给数据平面转发.控制平面通过内部控制接口接收数据平面接收到邻居节点的控制报文,用于计算分布路径.控

制平面通过内部接口接收到控制消息时,执行操作包括:①单播路由协议处理链路状态消息,并生成分布路径的路由策略,通过内部控制接口将其下发到数据平面,并将链路状态消息传递到服务代理模块;②服务代理模块封装控制消息,并通过外部控制接口传递到集中管理器,并监听到集中管理器下发的集中路径的路由策略,解封装后安装至单播路由协议中,由后者将路由策略通过内部控制接口安装到数据平面;③单播路由协议接收到数据平面上传的接口状态消息,如果接口状态从 UP 转变 DOWN 时,将下一跳与接口号相同的集中路径的路由策略删除,通过内部控制接口更新数据平面中转发表。

控制平面利用分布式路由协议同步网络状态并计算出分布路径,通过迅速感知网络状态来确保网络具备柔性抗毁、快速自愈的能力。同时,集中路径的撤销由分布式路由协议负责,由于后者能迅速感知到网络链路拓扑变化的信息并及时地撤销集中路径,可以防止集中路径对网络造成影响。这确保路由系统不会产生因为集中路径的部署导致网络无法正常运转问题。

(2) 数据平面

控制平面将转发策略(包括分布路径和集中路径)安装到数据平面,数据平面通过查找转发表来确定报文转发的下一跳。到达目的网络的转发策略可能由若干条构成,它们具备相同的前缀值和不同的下一跳。因此,提出一种两级转发表结构,由前缀表和策略表组成,如图 4 所示。

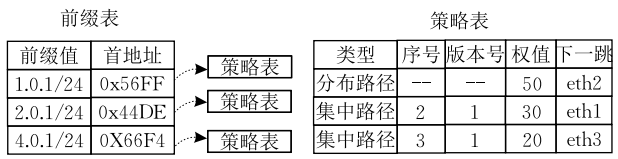


图 4 转发表结构

报文查找转发时,首先查找前缀表,以报文的地址为索引通过最长前缀匹配方式查找前缀表项,通过前缀表项中首地址索引到策略表。策略表包括路径基本信息(如类型、序号、版本号、权值等)以及下一跳。

策略表中路径基本信息主要目的在于维护集中路径:①类型用于区分路径类型,如集中路径或分布路径,分布路由协议在更新分布路径的路由策略时,只能更新类型为分布路径的策略表;②序号表示到达某个目的网络的路径编号是唯一的,且下一跳相同而序号不相同的表项并不聚合;中路径的状

态改变时,只撤销路径编号相同的集中路径,最大程度地保留集中路径;③版本号用于表明集中路径创建时间,说明集中路径的新旧程度,当撤销集中路径版本不低于表项版本号时则删除对应的表项,否则,不执行表项撤销工作;④权值用于表示达到目的网络的每条路径承载的数据流量,使得数据流按比例在路径上传输。

SDRS 支持数据流同时并行沿着若干路径传输,要求确保同一条数据流必须沿着相同路径传输,避免相同数据流沿着不同路径传输时报文到达次序与发送次序不一致,出现报文乱序问题。结合转发表的二级结构,提出一种基于 Hash 桶的转发路径选择策略,如图 5 所示。利用 Hash 桶思想计算出数据流归属的路径,相同数据流计算得出的路径相同,确保相同数据流沿着同一路径传输。

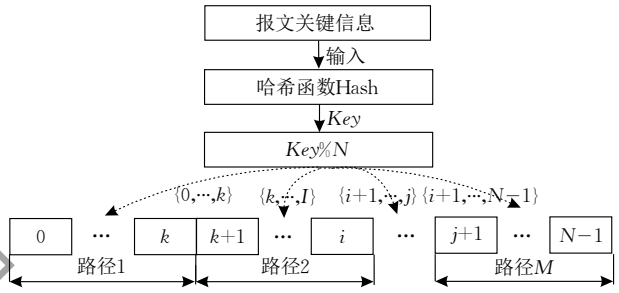


图 5 转发路径选择过程示意图

基于 Hash 桶的转发路径选择过程描述如下:①令 Hash 桶的规模为 N ,那么到达同一目的网络的所有路径的权值等于 N ,即单个策略表中路径的权值之和为 N ,并且每条路径具备权值区间,如路径 1 的权值区间为 $[0, k]$,路径 2 的权值区间为 $[k+1, i]$;②数据平面接收到报文时提取报文的关键信息,采用五元组{源 IP 地址,目的 IP 地址,源端口号,目的端口号,协议类型}的方式提取报文的关键信息;③以报文关键信息为输入调用 Hash 函数计算出 Key 值;同一条数据流计算的 Key 必定相等;④将 Key 值对 N 求余数 R , R 的取值区间为 $[0, N-1]$;⑤检索余数 R 值所属于的权值区间,则选择对应的传输路径。

Hash 桶的规模 N 和报文关键信息是直接影响到数据流能否按比例分配的两个重要因素。Hash 桶的大小关系到数据流按比例分配的精度: N 值越大,则数据流按比例分配的精度越高;相反,数据流按比例分配的精度越低。同时,报文关键信息直接关系到数据流按比例分配的关键:报文关键信息越多,数据流按比例分配精度越高,Hash 函数计算开销越大;报文关键信息越少,数据流比例分配精度越

低.同时,为了确保集中管理器对网络控制的健壮性,控制报文需通过分布路径进行传输,借助访问控制列表配合实施.

3 最大多路径覆盖树算法

集中管理器采用最大多路径覆盖树算法计算集中路径集.最大多路径覆盖树算法通过广度优先遍历网络拓扑图建立覆盖树,利用无环替代路由的变种对覆盖树进行剪枝,确保路径不构成路由环路,并保证计算出集中路径集包含所有转发路径.最大多路径覆盖树算法由多路径集计算和路径权值计算两部分组成:前者计算出多路径集,后者计算出每条路径承载的数据流的比重.

3.1 多路径集计算

多路径集计算是无环替代路由(Free Alternative Routing)的变种^[21],利用广度优先遍历算法构建覆盖树,确保覆盖树包含所有转发路径,优化剪枝规则对覆盖树进行剪枝,避免出现路由环路.多路径集计算包括以下两条剪枝规则:

规则 1. 端到端的路径上相同节点不可出现若干次.在构建覆盖树时,解决节点多次存在同一条路径的问题,避免产生路由环路.

规则 2. 无环替代路由变种, $Distance_opt(N, D) - Distance_opt(N, E) < Path_opt(N, D) + Distance_opt(N, E)$ (条件 1) 和 $Distance_opt(N, D) < Distance_opt(N, E) + Distance_opt(E, D)$ (条件 2). 由于路径 $Path_opt(N, D)$ 属于路径集 $Distance_opt(N, E)$, 所以 $Path_opt(N, D)$ 值小于等于 $Distance_opt(N, E)$ 值. 因此, 条件 1 是条件 2 的充分条件, 即路径满足条件 1 则必满足条件 2. 无环替代路由变种在无环替代路由条件 2 的基础上增加了条件 1, 提高了剪枝的效率.

3.2 路径权值计算

多路径集算法计算到达同一目的网络的路径集,通常包含若干条路径.多条路径可以同时并行传输数据流,但每条路径的性能指标不同,其承载的数据流的比重自然也不同.因此,每条路径分配权值,用于表示路径承载数据流的比重.

本文主要考虑路径代价和路径最高代价两个因素.路径代价指路径的总开销,用于表示网络传输的平均性能;路径代价越低,表明数据流的平均传输时延越低;相反,数据流的平均传输时延越大.路径最高代价是指路径中代价最大的链路,用于表示网络传输的性能瓶颈;路径最高代价越低,表明路径成为

网络瓶颈的概率越低;相反,路径成为网络瓶颈的概率越高.利用“指数惩罚”方式使得数据流尽可能在路径代价低、路径最高代价低的路径传输.

(1) 路径代价

为了确保端到端通信具有较低的传输时延,路径代价低的路径承载数据流比重大.因此,采用“指数惩罚”的方式来确保数据流在低代价路径分配的权值较大,高代价的路径分配的权值较小,如式(1)所示.

$$\frac{f(P_{s,t}(i))}{f(P_{s,t}(j))} \approx \frac{e^{-C(P_{s,t}(i))}}{e^{-C(P_{s,t}(j))}}$$

$$\text{iff } (P_{s,t}(i), P_{s,t}(j) \in P_{s,t}, P_{s,t}(i) \neq P_{s,t}(j)) \quad (1)$$

其中, $P_{s,t}(i)$ 表示源节点 S 到目的节点 t 的路径集中第 i 条路径; $P_{s,t}$ 表示源节点 S 到目的节点 t 的路径集; $f(P_{s,t}(i))$ 表示路径 $P_{s,t}(i)$ 承载的数据流比重; $C(P_{s,t}(i))$ 表示路径 $P_{s,t}(i)$ 的路径代价.

路径代价越高,路径权值越低,表明路径分配的数据流量比重越低;路径代价越低,路径权值越高,表明路径分配的数据流量比重越高.“指数惩罚”能有效地保证端到端通信尽可能利用低代价的路径进行数据流的传输.

(2) 路径最高代价

数据流量分配与路径最高代价密切相关,路径最高代价由路径上代价最高的链路决定,如式(3)所示.为了避免路径成为网络传输性能的瓶颈,路径最高代价承载数据流比重要低.因此,采用“指数惩罚”的方式来确保数据流在最高代价低的路径分配的路径权值较大,最高代价高的路径分配的路径权值较小,如式(2)所示.

$$\frac{g(P_{s,t}(i))}{g(P_{s,t}(j))} \approx \frac{e^{-MAXC(P_{s,t}(i))}}{e^{-MAXC(P_{s,t}(j))}}$$

$$\text{iff } (P_{s,t}(i), P_{s,t}(j) \in P_{s,t}, P_{s,t}(i) \neq P_{s,t}(j)) \quad (2)$$

其中, $MAXC(P_{s,t}(i))$ 表示路径 $P_{s,t}(i)$ 的最高代价.路径最高代价越高,表明路径成为网络性能概率越高,则路径分配的数据流量比重越低;路径最高代价越低,表明路径成为网络性能瓶颈概率越低,路径分配的数据流量比重越高.因此,“指数惩罚”有效地保证数据流尽可能在路径最高权值低的路径上传输.

$$MAXC(P_{s,t}(i)) = MAX\{\sum C(l_{m,n})\}$$

$$\text{iff } (l_{m,n} \in P_{s,t}(i)) \quad (3)$$

路径权值与路径代价和路径最高代价两个因素相关.结合式(1)和式(2),每条路径承载数据流量比重计算如式(4)所示.

$$\phi(P_{s,t}(i)) \approx \frac{e^{-\text{MINC}(P_{s,t}(i)) - \text{Cost}(P_{s,t}(i))}}{\sum_{j=0}^N e^{-\text{MINC}(P_{s,t}(j)) - C(P_{s,t}(j))}}$$

$$\text{iff } (i, j \in [0, \text{NUM}(P_{s,t})]) \quad (4)$$

其中, $\phi(P_{s,t}(i))$ 表示路径 $P_{s,t}(i)$ 承载的数据流量比重. 当路径代价越高和路径最高代价越高, 路径分配数据流量比重越低; 相反, 当路径代价越低和路径最高代价越低, 路径分配数据流量比重越高, 从而确保低代价、最高代价低的路径权值大.

3.3 算法时间复杂度

最大覆盖树算法以每个节点为根节点, 利用广度优先算法遍历拓扑图. 假设网络拓扑图为 $G(V, E)$, 则 $|V|$ 表示拓扑图 G 的节点数量, $|E|$ 表示拓扑图 G 的边数量. 由于广度优先算法的时间复杂度为 $O(|V| + |E|)$, 则最大覆盖树算法的时间复杂度为 $O(|V| * (|V| + |E|))$.

SDRS 系统是域内路由机制, 域内的节点数和链路数相对是有限的, 如域内的节点数通常在几十台或几百台内. 最大覆盖树算法可满足域内的多路径的计算.

4 路径维护机制

链路拓扑变化时, 分布式路由协议自主完成分布路径的撤销与重计算的任务. 为了避免集中管理器撤销集中路径的低效性, SDRS 利用分布式路由协议实现集中路径的撤销. 一方面, 分布路径的状态改变时, 不受影响的集中路径不被撤销, 确保数据流通过存活的集中路径继续传输, 有效地避免路由重收敛导致端到端的通信中断; 另一方面, 集中路径的状态改变时, 分布式路由协议只撤销受影响的路径, 并不撤销到达同一目的网络的其他集中路径. 从分布路径和集中路径两个方面展开描述 SDRS 的集中路径维护机制.

4.1 分布路径

分布路径的状态改变(如节点故障、链路故障等)和新邻居关系建立(如新的节点加入、故障节点恢复等)都会引起分布路径的维护. 针对两种情况分别描述集中路径维护机制.

(1) 分布路径的状态改变

由于集中路径是环路拓扑中非最低代价路径, 集中路径与分布路径并不相交, 所以集中路径并不需要撤销, 继续执行数据流的传输, 从而避免端到端的通信中断. 在新的分布路径重新收敛完毕时, 首先证明新收敛的分布路径与已存的集中路径不构成路

由环路.

假设在网络拓扑图 $G(V, E)$ 从源节点 S 到目的节点 T 间构建路径集, 且满足规则 2 无环替代路由变种.

证明: 网络拓扑图从 $G(V, E)$ 变为 $G'(V', E')$, 后者是前者的子图. 在子图 $G'(V', E')$ 执行最大路径覆盖树算法时, 计算的路径集同样满足满足规则 2. 因此, 一条路径在图 $G(V, E)$ 满足规则 2, 则在图 $G'(V', E')$ 也必满足规则 2. 因此, 新收敛的分布路径与原有的集中路径不构成路由环路.

(2) 新邻居关系建立

新邻居关系建立时, 分布式路由协议重新收敛路径, 并计算出新的分布路径. 但新的分布路径与存活的集中路径可能存在路由环路问题. 以图 6 为例, 节点 C 到节点 D 路径包括分布路径($C-M-D$)和集中路径($C-E-F-D$). 在某时刻, 链路 $C-F$ 建立连接, 节点 C 和 F 通告到全网. 此时, 节点 E 到 D 的分布路径($E-C-F-D$), 与集中路径($C-E-F-D$)形成路由环路. 因此, 在新邻居关系建立时, 应充分考虑新分布路径与已建立集中路径的冲突问题.

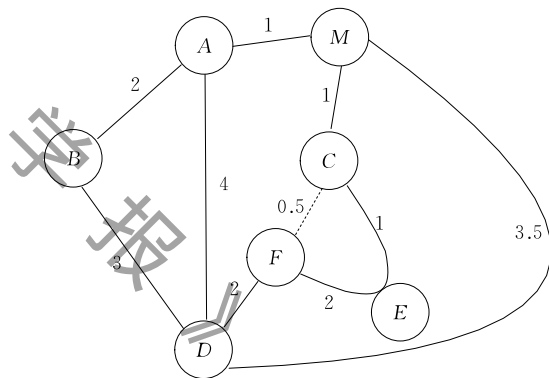


图 6 链路 $C-F$ 建立连接

因此, 新的邻居关系建立时, 节点通过计算出分布路径与到达同一目的网络的集中路径为输入判断是否满足规则 2; 如果满足规则 2, 表明已建立集中路径与分布路径不冲突, 无须撤销集中路径; 如果不满足规则 2, 表明存活的集中路径与分布路径冲突, 必须撤销集中路径, 否则形成路由环路, 集中路径撤销算法的描述如算法 1 所示. 首先收到新的链路状态报文 LSA, 重新计算出新的路由表项 R (前缀值为 $prefix$, 下一跳为 x) (步骤 1). 以 {前缀值 = $prefix$; 类型 = 分布路径} 查找路由表 (步骤 2). 如果返回结果 S 不为空, 表明该节点存在到达目的节点 T 的分布路径, 并检查路由表项 S 的下一跳是否等于 x ; 如果相等, 表明分布路径并没有改变, 则直

接返回(步骤 4~5);否则,删除旧的分布路径的路由表项 S ,并添加新的路由表项 R (步骤 6~8).同时,以{前缀值= $prefix$;类型=集中路径}再次查找路由表,得到路由表项集合 $S1$,遍历集合 $S1$ 中每条路由表项 $S1[j]$ (步骤 9~10).如果路由表项 $S1[j]$ 的下一跳等于 x ,表明分布路径与集中路径重合,则删除集中路径的路由表项,并添加分布路径表项(步骤 11~14).如果路由表项 $S1[j]$ 的下一跳不等于 x ,表明存在集中路径与分布路径到达同一目的端,则判断两条路径是否满足规则 2:如果满足规则 2,表明集中路径与分布路径不冲突,则直接添加分布路径的表项(步骤 17~19);否则,集中路径与分布路径构成环路,则删除集中路径表项,并添加分布路径的表项(步骤 20~23).

算法 1. 集中路径撤销算法.

输入:新 LSA

输出:路由表更新操作

```

1. 分布式路由协议以最低代价为  $metric$  计算分布路径,路由表项  $R$  前缀值为  $prefix$ ,下一跳为  $x$ ;
2.  $S = Search(\{前缀值 = prefix; 类型 = 分布路径\}, 路由表); // 查找路由表$ 
3. IF( $S \neq NULL$ )
4.   IF( $S.nextHop = x$ )
5.     return SUCCESS; // 路由表项已存在无须更新
6.   ELSE
7.     Delete( $S$ , 路由表); // 删除分布路径的路由表项
8.     Add( $R$ , 路由表); // 添加路由表项
9.    $S1[] = Search(\{前缀值 = prefix, 类型 = 集中路径\}, 路由表); // 查找路由表$ 
10.  WHILE( $j$  in  $S1$ ) //
11.    IF( $S1[j].nextHop = x$ )
12.      Delete( $S1[j]$ , 路由表); // 删除集中路径的路由表项
13.      Add( $R$ , 路由表); // 添加路由表项
14.      return SUCCESS; // 路由表项更新成功
15.    ELSE
16.       $S2 = 路径(S1, T)$  和  $路径(S1, E, D)$  调用规则 2;
17.      IF( $S2 == 满足$ )
18.        Add( $R$ , 路由表); // 添加路由表项
19.        return SUCCESS; // 路由表项更新成功
20.      ELSE
21.        Delete( $S1[j]$ , 路由表); // 删除集中路径
22.        Add( $R$ , 路由表); // 添加路由表项
23.        return SUCCESS; // 路由表项更新成功
24.  END

```

4.2 集中路径

新的邻居关系建立只会导致集中管理器执行最大多路径覆盖树算法重新计算出集中路径集,但并不

不会引起路由环路的问题.集中路径的状态改变时,如果分布式路由协议不将该集中路径撤销,报文继续沿着集中路径进行传输,导致报文丢包.分布式路由协议负责集中路径的撤销,避免集中管理器感知网络链路拓扑状态变化迟钝、集中路径的路由策略更新缓慢等导致路由收敛速度慢的问题.但传统分布式路由协议并不具备维护集中路径的功能,在分布式路由协议基础上扩展一类新的链路状态消息,即增强链路状态消息(Enhanced Link Status Advertisement, ELSA).当节点检测到集中路径的状态改变,将集中路径信息封装在 ELSA 消息中,并洪泛至网络中每个节点;从而确保每个网络节点感知到集中路径的状态改变信息,并撤销相应的集中路径.

集中路径维护机制包括两部分:(1) ELSA 消息生成与洪泛,节点检测到集中路径的状态时,生成包含集中路径信息的 ELSA 消息,并洪泛至网络;(2) ELSA 消息处理与路由更新,节点接收到 ELSA 消息,通过分析 ELSA 消息来执行路由更新,从而达到集中路径维护的目的.

(1) ELSA 消息生成与洪泛:节点中分布式路由协议检测自身某个网络接口的状态从 UP 状态变为其他状态(如 Down 状态等)时,表明网络节点与邻居节点的邻居关系失效.分布式路由协议查找路由表,找到路由类型为集中路径且下一跳等于该网络接口号.分布式路由协议生成 ELSA 消息,包括前缀值、路径编号、版本号等,并将 ELSA 消息洪泛至整个网络中.

(2) ELSA 消息处理与路由表更新:分布式路由协议接收到 ELSA 消息时,首先判断 ELSA 消息是否先前已接收;如果分布式路由协议已接收到相同 ELSA 消息,则将 ELSA 消息丢弃;否则,分布式路由协议执行下一步解析 ELSA 消息工作.然后以 ELSA 消息中前缀值为关键字查找路由表,如果前缀值完全匹配且查找到表项类型是集中路径、编号相同、版本号不高于 ELSA 消息中路径版本号,则删除该路由表项,并将 ELSA 消息洪泛至相邻节点.

以图 7 为例描述 ELSA 消息实现集中路径撤销的过程.假设集中管理器在节点 A 到节点 F 间建立一条集中路径(A-C-D-F),路由表项的信息如图 7(a)所示.在某一时刻,链路 D-F 出现单点故障.节点 D 检测到与节点 F 间的邻居关系失效,以{下一跳= $eth2$;类型=集中}为关键字查找路由表,并生成 ELSA 消息包含一条路由表项内容{前缀值=202.195/16;序号=1;版本号=44},将 ELSA 消息

洪泛至与其相连的邻居节点. 其他节点接收到 ELSA 消息时, 以 {前缀值=202.195/16; 序号=1; 版本号=44} 为关键字查找路由表: 一旦存在路由表项满足 {前缀值完全匹配; 类型为集中路径; 序列号等于 1; 版本号不高于 44} 时, 删除该路由表项, 如图 7(b) 所示.

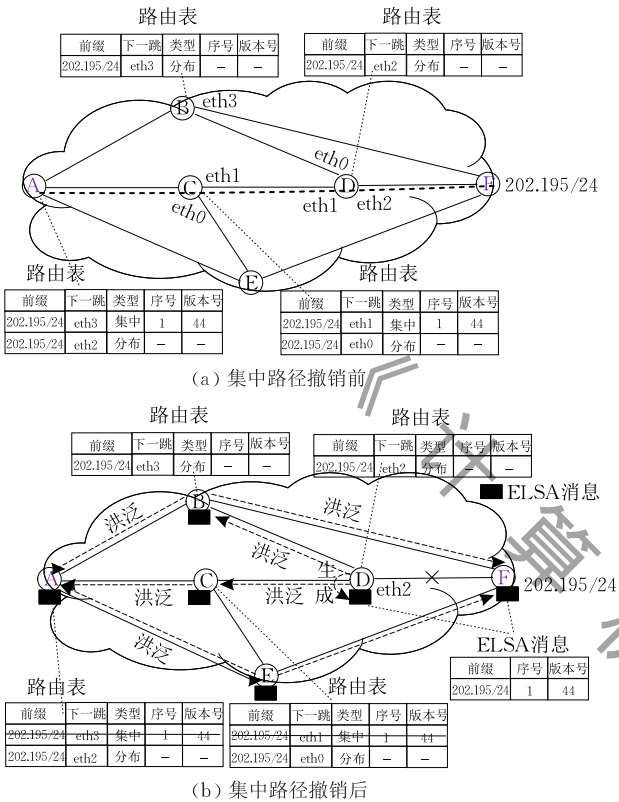


图 7 集中路径维护过程示意图

5 实验与验证

从最大多路径覆盖树算法性能测试、路由系统性能以及端到端的传输性能测试等三个方面展开实验, 通过实验数据来验证 SDRS 机制的可行性和有效性.

首先, 为了验证基于最大多路径覆盖树算法的性能, 选取了 10 个典型的网络拓扑图来测试算法的有效性, 如表 1 所示. 10 个典型网络拓扑图的信息包括拓扑图名称、节点数量、链路数量以及所属国家与地区, 是从 Internet Topology Zoo 筛选的代表性拓扑图^[22].

其次, 利用开源软件搭建了 SDRS 原型系统: 数据平面是基于 Click 模块化路由器^[23], 控制平面是基于 Quagga 路由器并选择 OSPF 路由协议^[24], 集中管理器则利用 Floodlight 控制器^[25]. SDRS 原型系统由 10 个网络节点、1 台管理器和 2 台终端组成. 本实验利用 CORE 模拟器^[26] 通过容器技术构建若

表 1 典型网络拓扑图

名称	序号	时间	节点	链路	国家/地区
CERNET	1	2010	35	49	中国
Abilene	2	2005	11	14	美国
Renater	3	2005	35	46	法国
Asia pacific	4	2010	15	21	亚太
Canerie	5	2010	24	31	加拿大
Ernet	6	2010	16	18	印度
Janet	7	2011	22	33	英国
Garr	8	2011	36	45	意大利
Gnet	9	2010	33	37	希腊
Pionier	10	2011	27	32	波兰

干虚拟机, 将网络节点和管理器的软件运行在不同的虚拟机中, 从而验证 SDRS 原型系统的传输性能.

5.1 最大多路径覆盖树算法性能测试

本实验对链路权值和子网信息作如下限定:

(1) 最大多路径覆盖树算法选取 Metric 值为路径跳数.

(2) 每个网络节点仅连接子网的数量为 1.

最大多路径覆盖树算法是基于无环替代路由的变种, 用于优化覆盖树剪枝效率. 算法执行时间用于表示最大覆盖树算法与无环替代路由算法的比值, 测量了在 10 种网络拓扑图下最大覆盖树算法与无环替代路由算法的比值, 如图 8 所示.

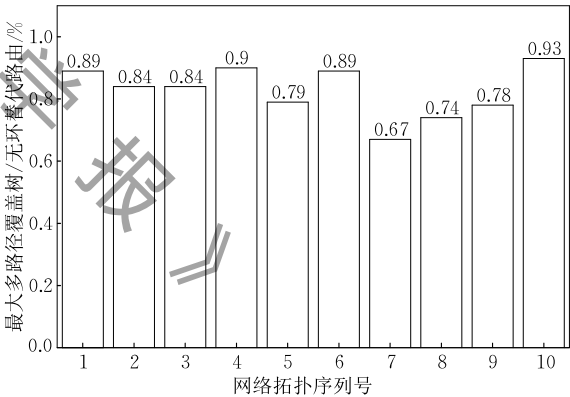


图 8 算法执行时间

最大多路径覆盖树算法的性能优于无环替代路由算法, 这主要因为最大多路径覆盖树算法在借鉴无环替代路由算法基础上进一步提出两种剪枝规则用于提升算法的剪枝效率. 比如, 计算 Janet 拓扑图的路径时, 最大多路径覆盖树算法的时间开销仅为无环替代路由的时间开销的 67%.

路由策略规模

路由策略规模指集中管理器下发的集中路径的路由策略总数. 路由下发策略规模越大, 对集中管理器性能要求也越高. 因此, 测量了 10 种网络拓扑图下需要配置的路由策略规模, 如图 9 所示.

50%网络拓扑结构配置的路由策略规模超过

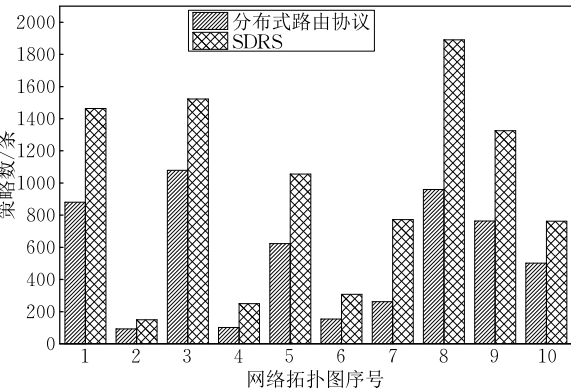


图 9 路由策略规模

1000 条,包括 CERNET、Renater、Canarie、Garr 和 Grnet;只有 30% 网络拓扑结构配置的路由策略规模不超过 310 条,包括 Abilene、Asia pacific 和 Ernet.但它们配置的路由策略规模都小于 2000 条.

SDRS 系统部署的路由策略数要多于分布式路由协议的策略数,通常与平均转发路径数量相关.如 Garr 拓扑图中,SDRS 中路由策略数约为分布式路由协议中路由策略数的 1.97 倍.因而,SDRS 的路由策略的存储代价开销要大于分布式路由协议的路由策略的存储开销.

5.2 路由系统性能

从路由收敛时间、链路拓扑获取时间、路由策略部署负载以及路径撤销时间等 4 个方面评估了 SDRS 路由系统的性能.

(1) 路由收敛时间

为了验证 SDRS 具有分布路由协议的快速收敛性能,测量了不同网络直径下路由收敛时间,如图 10 所示.

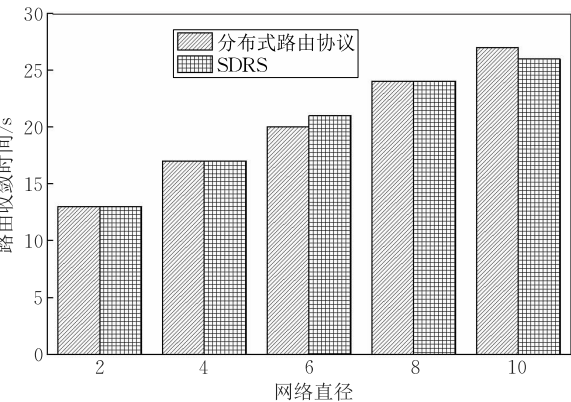


图 10 路由收敛时间

SDRS 系统既提供分布式路由协议的分布路径,又提供集中管理器创建的集中路径.端到端的数据流同时沿着两类转发路径传输.即使 SDRS 未完成集中路径的部署,SDRS 中分布式路由协议已通

过收敛为端到端通信创建分布路径.因而,SDRS 系统的路由收敛时间与传统的分布式路由协议的路由收敛时间是一致的.

(2) 链路拓扑获取时间

链路拓扑获取时间是指集中管理器初次连接时链路状态库同步的时间开销.只有集中管理器完成链路拓扑同步,才基于感知到的链路拓扑基础上计算集中路径.测量了 10 种网络拓扑结构图中链路拓扑获取时间,如图 11 所示.

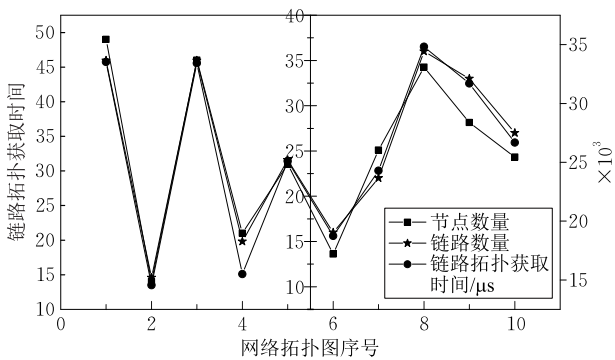


图 11 链路状态拓扑获取时间

网络链路数量与链路拓扑获取时间关联性相对较小.链路拓扑获取时间与网络节点数量呈线性相关:随着网络节点数量增加,链路拓扑获取时间相应增加;随着网络节点数量减少,链路拓扑获取时间相应减少.举个例子,Garr 拓扑图由 36 个网络节点组成,链路拓扑同步时间约为 0.35 s.集中管理器获取链路拓扑时间是几秒内完成,可以迅速感知到链路拓扑的信息.

(3) 路由策略部署负载

集中管理器完成路径计算后将路由策略部署到相应的网络节点.测量了 10 种网络拓扑结构下集中管理器部署路由策略的负载情况,如图 12 所示.

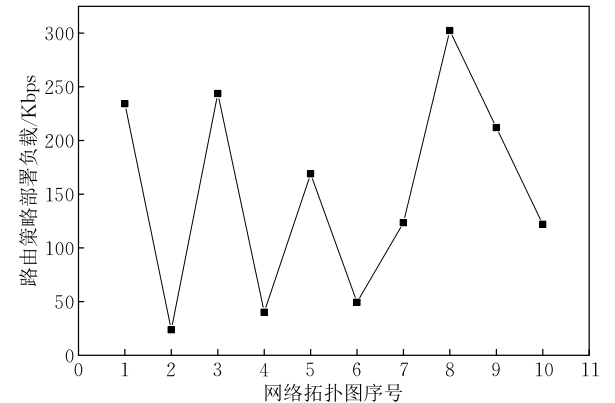


图 12 路由策略部署负载

集中管理器部署路由策略的负载在 0~300 Kbps 范围内,与路由策略规模密切相关.随着子网数量

(即前缀表项)增加,路由策略部署负载相应的增加.尤其,核心路由器中表项的规模达到 50 万条以上^①,集中管理器部署路由策略负载也会很大.例如,在 Canerie 拓扑图中,路由策略部署负载约为 3.67 Gbps.因而,应结合网络应用场景,充分利用 SDRS 的多路径业务编排能力,降低部署路由策略的规模,从而尽可能降低集中管理器的部署路由策略的负载.

(4) 路径撤销时间

SDRS 系统采用分布式路由协议维护集中路径;而现有纯集中架构由集中控制器统一撤销集中路径.因此,在 10 种网络拓扑图下分别测量了 SDRS 系统和纯集中架构下从故障产生到集中路径撤销的间隔,并计算纯集中架构与 SDRS 的路径撤销时间比值,如图 13 所示.

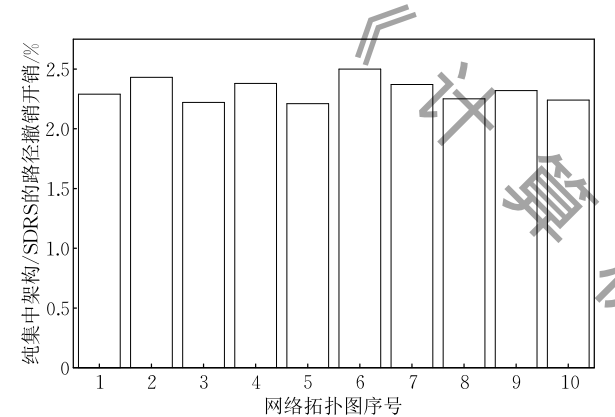


图 13 路径撤销时间开销比值

纯集中架构中路径撤销时间为 SDRS 系统中路径撤销时间的 2 倍以上.这主要因为纯集中架构中路径状态改变消息需要先通告到集中控制器,由后者下发撤销集中路径的策略;而 SDRS 系统中节点检测到集中路径的状态改变,直接生成集中路径撤销消息并洪泛到网络.因而,SDRS 系统采取的分布式路由协议维护集中路径的机制要优于纯集中架构下路径撤销机制.

5.3 端到端的传输性能测试

为了更好地衡量 SDRS 与分布式路由协议的优越性,从端到端的传输性能角度展开测试,包括传输时延、吞吐量以及分布路径故障时吞吐量等.

(1) 传输时延

传输时延指报文从一台终端发送至另一台终端花费的时间. SDRS 利用指数惩罚计算路径权值,并分配数据流在每条路径的分配.假定分布路径中链路延迟分别为 0、2000 μ s、4000 μ s、6000 μ s、8000 μ s 以及多路径数量为 8,测量分布式路由协议、SDRS、

ECMP 下端到端通信的传输时延,如图 14 所示.

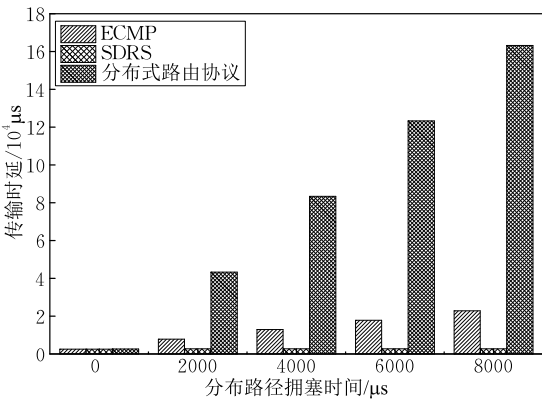


图 14 传输时延

当分布路径非拥塞时,SDRS、分布式路由协议、ECMP 协议的传输时延是一致的.当分布路径拥塞时间增大时,分布式路由协议和 ECMP 的传输时延相对增加;SDRS 的传输时延并没有明显增加.这主要因为 ECMP 为等价多路径协议,每条路径分配的数据流是相同的,分布路径的拥塞时间依旧很大,从而导致端到端传输时延依旧较大;而 SDRS 根据路径代价利用指数惩罚在每条路径分配不同的流量,在分布路径分配的数据流仅约为 0.09%.因此,SDRS 系统在优化网络传输性能方面要优于分布式路由协议和 ECMP.

(2) 吞吐量

吞吐量指终端到另一终端间数据流的流量. SDRS 通过多条路径同时传输数据流,可以降低链路拥塞对用户服务体验质量的影响.假定分布路径丢包率分别为 0、10%、20%、40%以及 80%,并测量不同路径数量下端到端的吞吐量,如图 15 所示.

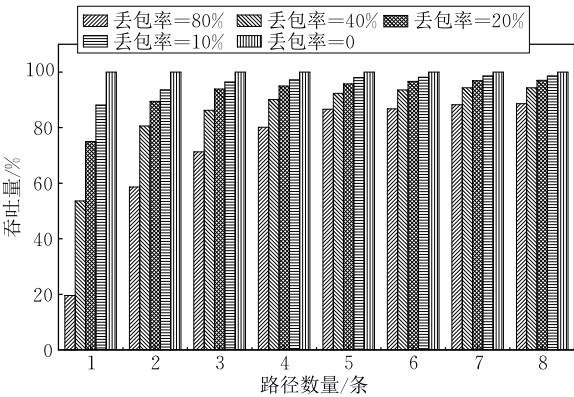


图 15 吞吐量

当分布路径非拥塞时,SDRS 与分布路由协议

① Advanced Network Technology Center in University of Oregon. Route-Views Project, 2017, 1, 31 [Online]. <http://www.routeviews.org>

的吞吐量是基本相同的,表明网络非拥塞时 SDRS 并不能显著提升端到端的吞吐量. 当分布路径拥塞时,SDRS 可以明显地提升端到端吞吐量,性能要优于分布式路由协议的吞吐量. 比如,分布路径的丢包率为 20%时,SDRS 建立路径数量分别为 2、4、8 时,是分布式路由协议的吞吐量 1.19 倍、1.27 倍、1.29 倍. 由于 ECMP 采用等价多路径的方式传输,到达同一目的网络的多路径数量通常要低于 SDRS 中多路径数量. 因此,ECMP 的吞吐量要低于 SDRS 的吞吐量. 实验结果表明,SDRS 有效地提升端到端的吞吐量,尤其在分布路径拥塞程度较强时.

(3) 分布路径故障时吞吐量

分布路径故障时吞吐量指分布路径出现故障时端到端的吞吐量. 分布路径出现故障时,SDRS 中与分布路径不相交的集中路径无须撤销,数据流依旧通过存活的集中路径继续传输,测量在与分布路径不相交的集中路径数量不同情况下端到端的吞吐量,如图 16 所示.

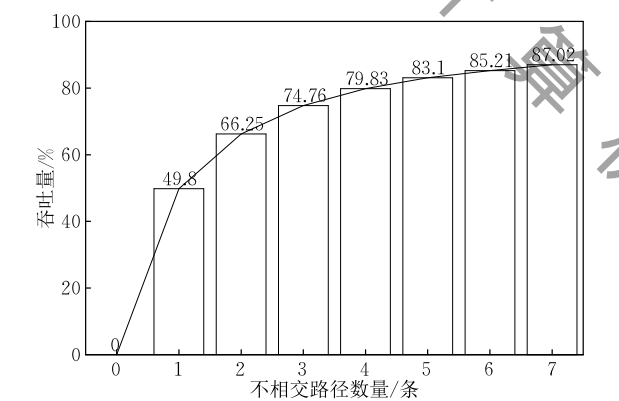


图 16 分布路径故障时吞吐量

分布式路由协议中路径出现故障时端到端通信被彻底中断;只有在分布式路由协议完成路由收敛时,端到端通信才恢复正常. 随着与分布路径不相交的集中路径数量的增加,端到端通信吞吐量增加,表明分布路径故障对端到端通信的影响较低. 比如,在不相交的集中路径数量为 1 时,端到端通信的 49.8%数据流依旧正常传输;当不相交的集中路径数量为 5 时,端到端通信的超过 80%的数据流不会受到分布路径故障影响. 尽管增加不相交的集中路径数量能降低分布路径对端到端通信的影响,但不相交的集中路径数量带来的性能提升程度也在逐步降低. 由于 ECMP 采用等价多路径计算方式,在分布路径故障时 ECMP 撤销多路径,并不具备 SDRS 继续保障数据传输的能力.

5.4 归纳与总结

结合实验测量结果,从路由收敛时间、路径撤销

时间、系统传输性能、路由策略编排以及路由表项规模等五个方面对比 SDRS、ECMP、集中控制等三种机制的优劣,如表 2 所示.

表 2 典型路由控制机制对比

项目	SDRS	ECMP	集中控制
路由收敛时间	分布路径收敛时间等于与 ECMP 相同 集中路径收敛时间与集中控制相同	路由收敛性能最优	路由收敛性能最差
路径撤销时间	集中路径和分布路径撤销时间与 ECMP 相同	路径撤销时间最优	路径撤销时间最慢
系统传输性能	系统传输系统与集中控制相同	系统传输性能最差	系统传输性能最优
路由策略编排	与集中控制同样具备全视图编排能力	不具备	具备全视图编排能力
路由表项规模	路由表项规模与集中控制相当	路由表项规模最小	路由表项规模最大

从上述分析可知,SDRS 机制具备分布式路由协议的快速收敛和柔性抗毁能力;同时,具备了集中控制的全视图、路由策略灵活编排的属性. 尽管 SDRS 导致路由表项规模增加,可通过灵活地设置路由计算策略来达到限制路由表项规模.

6 总结与展望

传统的单播路由协议在局部视图的不完整性、局部策略的自私性以及单条路径传输数据流等方面面临着挑战. 以 SDN 为代表的集中控制架构为解决单播路由协议的问题带来了新的思路. 本文第一次提出了将集中控制机理引入到分布式路由系统的软件定义路由系统概念,把集中与分布控制有机地耦合在一起,充分发挥两类控制架构的优势,相互弥补各自的不足. 最后,搭建了 SDRS 原型系统来验证其可行性和有效性.

下一步工作围绕 SDRS 架构从以下几个方面继续深入探讨:

- (1) 外部接口标准化,Netconf^[27] 和 YANG^[28] 是未来网络控制模型发展趋势. 在 SDRS 系统中,分布式路由协议需将链路状态通告消息 LSA 主动推送给集中管理器,确保集中管理器与分布式路由协议中网络状态的一致性. 目前,Netconf 仅支持主动向分布式路由协议配置路由策略及获取路由信息. 需要扩展 Netconf 协议支持分布式路由协议的事件主动通告到集中管理器的功能.
- (2) 集中路径信息同步,在集中管理器失效重新连接到网络节点时,集中路径继续存在网络中,导

致集中管理器存储的集中路径信息可能与网络中运行的集中路径状态存在不一致. 为了确保两者的一致性, 集中管理器重新接入到网络时首先与网络中集中路径信息进行同步.

(3) 路由策略的原子性, 集中管理器部署单条集中路径通常包含若干路由策略, 并分别部署在不同网络节点中. 要确保部署的一系列的路由策略具有原子性, 具体表项在单个路由策略配置失败时回滚该集中路径的已配置的路由策略; 以及单条路径的若干路由策略部署有严格先后顺序, 避免由于路由策略配置次序问题导致报文丢包.

(4) 应用规模, 本文仅验证了几十个节点的网络规模中 SDRS 可行性和有效性. 在下一步工作着重研究在几百个节点的网络规模下 SDRS 性能分析, 并针对 SDRS 面临的挑战提出相应的解决或优化方案, 从而提供一种通用性的路由机制.

致 谢 在此, 我们向对本文的工作给予支持和建议的同行表示衷心的感谢!

参 考 文 献

- [1] Xu D, Chiang M, Rexford J. Corrections to link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering. *IEEE/ACM Transactions on Networking*, 2015, 23(5): 1702-1703
- [2] Dearlove C, Clausen T. Multi-topology extension for the optimized link state routing protocol version 2 (OLSRv2). USA: Internet Engineering Task Force, RFC: 7722, 2015
- [3] Li L, Hu N, Liu K, et al. AMTCP: An adaptive multi-path transmission control protocol//*Proceedings of the 12th ACM International Conference on Computing Frontiers*. Ischia, Italy, 2015: 1-8
- [4] Guven T, Kommareddy C, La R J, et al. Measurement based optimal multi-path routing//*Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*. Hong Kong, China, 2004: 187-196
- [5] Merindol P, Pansiot J J, Cateloin S. Low complexity link state multipath routing//*Proceedings of the INFOCOM Workshops*. Rio de Janeiro, Brazil, 2009: 1-6
- [6] Ramasubramanian S, Krishnamoorthy H, Krunz M. Disjoint multipath routing using colored trees. *International Journal of Computer & Telecommunications Networking*, 2007, 51(8): 2163-2180
- [7] Zaumen W T, Garcia-Luna-Aceves J J. Loop-free multipath routing using generalized diffusing computations//*Proceedings of the 17th Joint Conference of the IEEE Computer and Communications Societies*. San Francisco, USA, 1998: 1408-1417
- [8] Vutukury S, Garcia-Luna-Aceves J J. MDVA: A distance-vector multipath routing protocol//*Proceedings of the 12th Annual Joint Conference of the IEEE Computer and Communications Societies*. Alaska, USA, 2001: 557-564
- [9] Vutukury S, Garcia-Luna-Aceves J J. An algorithm for multipath computation using distance-vectors with predecessor information//*Proceedings of the 8th International Conference on Computer Communications and Networks*. Boston, USA, 1999: 534-539
- [10] Pham P P, Perreau S. Performance analysis of reactive shortest path and multipath routing mechanism with load balance//*Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications*. San Francisco, USA, 2003: 251-259
- [11] Akyildiz I F, Lee A, Wang P, et al. A roadmap for traffic engineering in SDN-OpenFlow networks. *Computer Networks*, 2014, 71(3): 1-30
- [12] Jammal M, Singh T, Shami A, et al. Software defined networking: State of the art and research challenges. *Computer Networks*, 2014, 72(111): 74-98
- [13] Bennesby R, Mota E, Fonseca P, et al. Innovating on inter-domain routing with an inter-SDN component//*Proceedings of the IEEE 28th International Conference on Advanced Information Networking and Applications*. Victoria, Canada, 2014: 131-138
- [14] Kotronis V, Gämperli A, Dimitropoulos X. Routing centralization across domains via SDN: A model and emulation framework for BGP evolution. *Computer Networks*, 2015, 92(2): 227-239
- [15] Liu W, Zhou W, Wang Y, et al. Flexible multi-path routing for global optimization in software-defined datacenters//*Proceedings of the IEEE Symposium on IEEE Computers and Communication (ISCC)*. Messina, Italy, 2016: 190-195
- [16] Liu Y, Pan Y, Yang M, et al. The multi-path routing problem in the software defined network//*Proceedings of the 11th International Conference on Natural Computation (ICNC)*. California, USA, 2015: 250-254
- [17] Vissicchio S, Tilmans O, Vanbever L, et al. Central control over distributed routing. *ACM SIGCOMM Computer Communication Review*, 2015, 45(4): 43-56
- [18] Alvizu R, Maier G, Tornatore M, et al. Differential delay constrained multipath routing for SDN and optical networks. *Electronic Notes in Discrete Mathematics*, 2016, 52(1): 277-284
- [19] Krishnan Y N, Shobha G. Performance analysis of OSPF and EIGRP routing protocols for greener internetworking//*Proceedings of the IEEE International Conference on Green High Performance Computing (ICGHPC)*. SXCE, India, 2013: 1-4
- [20] Eastlake D, Banerjee A, Ghanwani A, et al. Transparent interconnection of lots of links (TRILL) use of is-is. USA: Internet Engineering Task Force, RFC: 7176, 2014
- [21] Zinin E A. Basic specification for IP fast reroute: Loop-free alternates. USA: Internet Engineering Task Force, RFC:

5286, 2008

- [22] Knight S, Nguyen H X, Falkner N, et al. The Internet Topology Zoo. *IEEE Journal on Selected Areas in Communications*, 2011, 29(9): 1765-1775
- [23] Kohler E, Morris R, Chen B, et al. The click modular router. *ACM Transactions on Computer Systems*, 2001, 18(3): 263-297
- [24] Jakma P, Lamparter D. Introduction to the quagga routing suite. *IEEE Network*, 2014, 28(2): 42-48
- [25] Wallner R, Cannistra R. An SDN approach: Quality of service using big switch's floodlight open-source controller// *Proceedings of the Asia-Pacific Advanced Network*, Daejeon, Korea, 2013, 35: 14
- [26] Ahrenholz J, Danilov C, Henderson T R, et al. CORE: A real-time network emulator// *Proceedings of the IEEE Military Communications Conference*. Washington, USA, 2008; 1-7
- [27] Enns R. NETCONF configuration protocol. *Bootstrap Protocol Stanford & Sun Microsystems*, 2006, 208(2): 367-378
- [28] Xu H, Xiao D. Data modeling for NETCONF-based network management: XML schema or YANG// *Proceedings of the IEEE International Conference on Communication Technology*. Hangzhou, China, 2008; 561-564



GAO Xian-Ming, born in 1988, Ph.D. candidate. His current research interests include programmable routing protocols and network equipment.

WANG Bao-Sheng, born in 1971, professor. His current research interests include next-generation network architecture, and security of network communication.

DENG Wen-Ping, born in 1981, lecturer. His current research interests include next-generation network architecture, and routing protocol.

Background

Unicast routing protocol runs in network layer of TCP/IP model, to provide reachable transmission channels for end-to-end communication. It firstly synchronizes network status among distributed network nodes by flooding way, and then calculates the shortest routing path to destination network according to beforehand rules (e. g. minimum hops, minimum costs, etc.). Therefore, it has ability of flexible survivability and fast self-healing, which is the fundamental reason for its massive success in the Internet. In order to solving data-flow transmission via single path in unicast routing protocol, multi-path routing schemes that establishing several equal or nor-equal routing paths for the same end-to-end communication and transmitting data-flow in parallel, are proposed by research communities. Fortunately, the centralized control architecture (e. g., Software Defined Network) is able to attain real-time network status and calculate the optimal routing strategies, promoting network toward optimal target. Meanwhile, centralized control architecture can support for dynamical configuration and management through the openness of control interface of routing protocol, to obtain orchestration ability of routing system. This way provides a solution to ossification of calculation rules in existing multi-path routing schemes in the foreseeable future.

This paper puts forwards Software Defined Routing System (SDRS), by introducing centralized control mechanism into routing system on basis of distributed paths generated by distributed routing protocol. Centralized controller can calculate centralized path set without routing loops based on Maximum Multi-path Overlay Tree Algorithm, and selectively deploy centralized paths into network according to orchestration strategy. Therefore, distributed path and centralized path can simultaneously transmit data-flow in parallel by coupling these two control mechanisms. It not only preserves the properties of flexible survivability and fast self-healing in distributed routing protocol, but also keeps global optimal ability of centralized controller.

This research is funded by the National Natural Science Foundation of China (NSFC) project Grant No. 61373039 which aim at protection of routing protocol by verifying routing strategies. This paper is also supported by the 973 project "Reconfigurable Network Experimental Platform" Grant No. 2012CB315906 which aim at establishment opening experimental platform via reconfiguring router platform by network functions virtualizations. Our research group mainly researches two key research contents: high elastic routing protocol, and programmable network platform.