计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210100051

基于混合软件定义网络的单节点故障保护方法

耿海军[1,2]　王　威[1]　尹　霞[3]
1 山西大学计算机与信息技术学院　太原 030006
2 山西大学自动化与软件学院　太原 030006
3 清华大学计算机科学与技术系　北京 100084

摘　要　软件定义网络(Software Defined Network,SDN)是由美国斯坦福大学 Clean Slate 课题组提出的一种新型网络体系架构,该架构通过解耦控制平面和转发平面的功能来实现网络流量的灵活转发。但是,由于经济开销和技术条件的限制,互联网服务提供商的骨干网必定长期处于传统设备和 SDN 设备共存的混合 SDN 状态。因此,在混合 SDN 网络中研究应对单节点故障情形的路由保护方法是一个关键的科学问题。文中首先描述了混合 SDN 网络中应对单节点故障情形时需要解决的问题,然后通过两种启发式方法来解决该问题,最后在真实拓扑结构和模拟拓扑结构中对提出的启发式算法进行测试。实验结果表明,在传统骨干网中,仅需要将一小部分传统设备升级为 SDN 设备,所提算法就可以应对网络中所有可能的单节点故障情形。

关键词:混合软件定义网络;实时应用;单节点故障;路由保护算法;启发式算法
中图法分类号　TP311

Single Node Failure Routing Protection Algorithm Based on Hybrid Software Defined Networks

GENG Hai-jun[1,2],WANG Wei[1] and YIN Xia[3]
1 School of Computer and Information Technology,Shanxi University,Taiyuan 030006,China
2 School of Automation and Software Engineering,Shanxi University,Taiyuan 030006,China
3 Department of Computer Science & Technology,Tsinghua University,Beijing 100084,China

Abstract　Software defined network (SDN) is a new network architecture proposed by the clean slate research group of Stanford University. The significant feature of this architecture is to decouple the functions of control plane and forwarding plane,and to flexibly forward the network traffic. Based on this,Internet service providers have deployed SDN technology in their backbone network to maximize the utilization of network resources. However,due to the limitation of economic cost and technical conditions,the backbone network of internet service providers must be in the hybrid SDN network for a long time. The studies have shown that single network node failure is inevitable and occurs frequently. Therefore,it is a key scientific problem to study the routing protection me-thod for single network component failure in hybrid SDN networks. In this paper,the route protection method for single network component failure in hybrid SDN network is described,and then two heuristic methods are used to solve the problem. Finally,the proposed heuristic algorithms are tested in real and simulated topologies. The experimental results show that in the traditional backbone network,only a part of the traditional devices need to be upgraded to SDN devices,and the algorithms proposed in this paper can deal with all possible single network node failure cases in the network.

Keywords　Hybrid software defined network,Real-time application,Single node failure,Routing protection algorithm,Heuristic algorithm

1　引言

随着互联网的快速发展,互联网中自治系统的规模和数量急剧增长,给域内路由带来了许多迫切需要解决的问题,其中路由可用性[1-2]显得尤为突出。针对网络故障的研究表明,网络中的故障频繁出现,并且不可避免。然而,发生网络故障时,传统路由协议并不能在 50 ms 内完成收敛,无法实现实时应用,如 IP 语音、股票在线交易、远程手术、视频流媒体和

到稿日期:2021-01-06　返修日期:2021-05-07　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:山西省应用基础研究计划(20210302123444);国家自然科学基金(61702315);山西省重点研发计划(201903D421003);国家高技术研究发展计划(863)(2018YFB1800401)
This work was supported by the Applied Basic Research Plan of Shanxi Province(20210302123444),National Natural Science Foundation of China(61702315),Key R&D Program of Shanxi Province China(201903D421003) and National High Technology Research and Development Program of China(863)(2018YFB1800401).
通信作者:耿海军(ghj123025449@163.com)

即时通信等对网络收敛时间的需求[3]。因此，提高域内路由可用性成为互联网亟需解决的关键科学问题。

软件定义网络[4]是最近兴起的一种新型的网络体系架构，该架构最大的特点是解耦了控制平面和转发平面的功能。SDN 网络相比传统网络具有许多优点，如可灵活控制网络流量、易于网络管理和安全策略的实施等。因此，许多互联网服务提供商(Internet Service Provider，ISP)将 SDN 技术部署在他们的骨干网中，以为其客户提供更加优质的服务。但是，在短期内将骨干网中所有的传统网络设备升级为 SDN 设备将是一件不可能完成的工程。这是因为一方面部署 SDN 技术面临巨大的人力和物力开销，另一方面部署 SDN 技术可能会导致网络中断，影响用户体验。基于此，互联网必将长期处于传统设备和 SDN 设备共存的混合 SDN 网络[5]。

作为下一代互联网体系结构，混合 SDN 网络吸引了学术界和工业界的密切关注，并且成为了网络中的一个热点研究领域，为研究路由可用性问题提供了新的解决思路。学术界普遍采用路由保护方法来提高路由可用性，因此本文研究混合 SDN 网络中的路由保护方法，以保证该方法能够解决网络中所有可能出现的单节点故障情形。

本文的主要研究工作如下：

(1)现有的路由保护方案多数是在单一网络体系结构中研究的，但是随着 SDN 体系结构逐步在骨干网中部署，研究混合体系结构中的路由保护方案就成为了一项很重要的研究课题。

(2)针对上述科学问题，本文提出了两种启发式的方法用于快速解决上述问题，并对算法的复杂度进行了理论分析。

(3)理论和实验结果均表明，本文算法不仅具有较小的计算开销，而且可以应对网络中所有可能的单节点故障情形，更加符合实际应用的需求。

2 国内外研究现状

大量针对网络故障的研究表明，网络中的故障频繁出现，并且不可避免[6]，工业界和学术界通常都会采用路由保护方案[7]来应对网络中频繁发生的故障。等价多路径路由(Equal-Cost Multipath Routing，ECMP)是学术界最早采用的一种最简单的路由保护方案，但是研究证实，该方案无法提供较高的路由可用性[8]。针对 ECMP 存在的问题，国际互联网工程任务组(Internet Engineering Task Force，IETF)发布了快速重路由的框架，在该框架的基础上提出了 LFA[9]、基于 Not-Via 的路由保护方案[10]和基于隧道的路由保护方案[11]等。在所有的路由保护方案中，LFA 因其简单高效的优点而受到学术界的密切关注，并且得到了华为和华三等路由器厂商的部署和支持。虽然 LFA 简单且容易部署，但是 LFA 有一个致命的缺点，即 LFA 无法保护网络中所有可能出现的单节点故障情形。为了解决 LFA 存在的问题，文献[12]利用图论的理论知识分析了 LFA 故障覆盖率的问题，通过调整网络中链路的代价来增加 LFA 的路由可用性，但是该方法并不一定保证能应对所有的单节点故障情形。为此，文献[13]从理论上详细分析了 LFA 的路由可用性和网络拓扑之间的关系，并通过增加链路来提高 LFA 的单节点故障保护情形。文献[14]研究了如何将 LFA 部署至 SDN 网络中，以解决网络中所有可能出现的单节点故障问题。文献[2]研究了如何在大规模SDN网络中快速地为所有节点对计算备份路径，从而应对网络中所有可能出现的单链路故障情形，该方法将计算备份路径的过程分为两个阶段，分别是索引阶段和查询阶段。索引阶段仅计算一些中间结果，而不需要计算出整条备份路径。查询阶段将利用索引阶段获得的中间结果计算出最终的备份路径。文献[15]首先将在 SDN 网络中实现故障恢复的方法归结为一个整数线性规划模型，然后设计了一个有效的启发式算法(CALFR)用于解决该问题。大量仿真结果表明，CALFR 不仅可以实现快速恢复，而且还可以避免故障恢复期间的网络拥塞。文献[16]提出了一种在软件定义网络(SDN)中具有流聚集的局部快速重路由(LFR)算法。在 LFR 中，如果检测到链路故障，则受该故障影响的所有业务流将被聚合为一个新的大流。然后，SDN 控制器为聚合流动态地计算新的路径。LFR 减少了 SDN 控制器和交换机之间的流操作数。实验结果表明，该算法在保证 SDN 入口流量最小的前提下，实现了网络故障的快速恢复。文献[17]提出了一种基于拥塞感知的快速故障恢复方案(Congestion-aware Fast Failure Recovery Schem，CAFFE)，该方案不仅能快速地从各种故障场景中恢复受影响的流，而且还能避免恢复后网络中潜在的拥塞。实验结果表明，在交换机和控制器中，CAFFE 都能以较低的开销实现网络故障的快速恢复。文献[18]提出了一种新的用于软件定义广域网流量工程的故障恢复系统 Sentinel。Sentinel 采用预先计算出备份路径的方法来加快故障恢复。当链路发生故障时，交换机会将受故障影响的流重定向到备份隧道，并立即在数据平面中恢复，这样可以大大降低网络拥塞现象。Sentinel 仅需要引入少量的附加转发规则就可以轻松实现网络故障恢复，并且可以很容易地在 Openflow 交换机上实现。针对混合 SDN 网络中单链路故障的恢复，文献[19]将其归结为一个 0-1 整数规划问题，并利用启发式算法来计算该问题对应的近似最优解，实验结果表明，该算法仅需将传统网络中的少部分结点升级为 SDN 结点，即可应对网络中可能出现的单链路故障情形，但文献[19]并没有考虑到混合 SDN 网络中可能出现的单节点故障问题。针对路由保护方案中默认路径和备份路径包含的公共边数量较高以及为计算公共边数量较少的路径而对默认路径进行限制的问题，文献[20]提出了一种转发算法，经实验测试，该算法不仅具有较低的计算复杂度，而且还可以降低默认路径和最短路径包含的公共边数量，从而提升网络可用性。文献[21]提出了一种多隧道路径修复机制，经实验验证，它能够在发生故障时增加可使用的潜在修复路径数量，并且还提出了一种 SDN 候选选择算法，有效缩短了修复路径的长度。然而，已有的路由保护方案都是在传统网络体系结构或者 SDN 网络体系结构的基础上单独展开研究，不能直接应用到混合 SDN 网络结构中。因此，本文主要研究如何将路由保护方案应用至混合 SDN 网络中，以保证该方法能够解决网络中所有可能出现的单节点的故障问题。本文的研究更符合互联网的演化过程，更易于在实际网络中部署。

本文第 3 节对网络模型进行了介绍，同时对本文需要解决的问题进行了描述；第 4 节提出了两种算法来解决如何在混合 SDN 网络中部署针对单节点故障的路由保护方案；第 5 节在不同网络拓扑结构中对算法进行了实验模拟，并且总结

了实验结果；最后总结全文。

3 网络模型和问题描述

3.1 网络模型

图 $G=(V,E)$用于表示一个网络拓扑结构，其中 V 为该拓扑中节点的集合，E 为该拓扑中边的集合。对于 $\forall v\in V$，$N(v)$表示该节点的所有邻居节点，$spt(v)$为以节点 v 为根的最短路径树，对于 $\forall x,y\in V(x\neq y)$，$sp(x,y)$为节点 x 到节点 y 的最短路径上的节点集合，$cost(x,y)$表示在网络 G 中节点 x 到节点 y 的最小代价，$dn(x,y)$为节点 x 到节点 y 的最优(默认)下一跳，$sn(x,y)$为节点 x 到节点 y 的最短路径的第二跳，$bn(x,y)$为节点 x 到节点 y 的备份下一跳。

本文通过一个例子来解释上面的网络模型。图 1 给出了以节点 c 为根的最短路径树 $spt(c)$。节点 c 的邻居节点可以表示为 $N(c)=\{a,b\}$，节点 c 到节点 g 的最短路径 $sp(c,g)=(c,a,h,g)$，节点 c 到节点 g 的最短路径的代价为 $cost(c,g)=3+2+1=6$，节点 c 到节点 g 的默认下一跳为 $dn(c,g)=a$，节点 c 到节点 g 的第二跳为 $sn(c,g)=h$。同理，节点 c 到节点 f 的最短路径 $sp(c,f)=(c,b,f)$，节点 c 到节点 f 的最短路径的代价为 $cost(c,f)=5+1=6$，节点 c 到节点 f 的默认下一跳为 $dn(c,f)=b$。

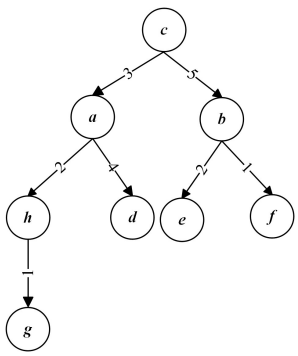

图 1 以节点 c 为根的最短路径树 $spt(c)$

Fig. 1 Shortest path tree rooted at node c

定义 1 对于任意的源 s-目的 d 节点对，假设它们之间的最短路径是$(s,bn(s,d),sn(s,d),\cdots,d)$，若网络中存在一个节点 t 满足下面两个条件，则称该源-目的节点对被保护。

(1)$bn(s,d)\notin sp(s,t)$；

(2)$bn(s,d)\notin sp(t,sn(s,d))$或者存在一个节点 $x\in N(t)$，$bn(s,d)\notin sp(x,sn(s,d))$。

本文用图 2 来解释定义 1，在该图中，源 s-目的 d 的最短路径是$(s,a,b,c,\cdots,d)$，如果存在一个节点 t 满足 $a\notin sp(s,t)$和 $a\notin sp(t,b)$同时成立或者节点 t 有一个邻居节点到节点 b 的最短路径不包含节点 a，则称该源-目的节点对被保护。

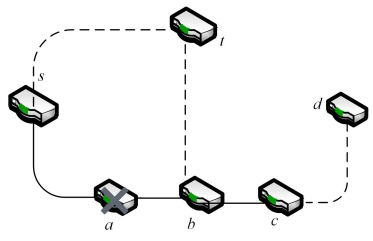

图 2 解释定义 1 的例子

Fig. 2 Example for explaining definition 1

定义 2 故障保护率 $R(G,M)$可以定义为被保护的源目的节点对的数量除以网络中所有源目的节点对的数量，其中 M 为 SDN 节点的集合。

3.2 问题描述

现在互联网部署的域内路由协议主要是链路状态路由协议，如 OSPF 等。在这种路由协议中，网络中的全部路由器都存有本自治域内的完整拓扑结构。当网络处于稳定状态时，这些路由器中存储的拓扑结构是一致的。网络中的每个路由器根据该网络拓扑结构，利用最短路径优先算法(Shortest Path First，SPF)来计算一棵以自己为根的最短路径树(Shortest Path Tree，SPT)，之后根据该树构造出路由表。由上述描述可知，若源节点到目的节点的默认下一跳出现故障，则会使传输到该点的报文丢失，从而造成网络中断，无法满足用户体验。根据上述描述可知，目前互联网采用最短路径转发报文，当网络出现故障时会导致网络中断，严重影响网络性能。因此，本文研究如何在混合 SDN 网络中实现，可以保护所有可能出现的单节点故障情形的路由保护算法。本文需要解决的问题可以描述为：给定一个网络拓扑 $G=(V,E)$，如何设计一种高效的路由保护算法，使该算法从网络中选择一组数量最小的节点部署 SDN 技术，并且能够应对网络中所有可能出现的单节点故障情形。

4 GARPHSDN 和 IPGARPHSDN 算法

本文的解决思路如下：首先计算出所有源-目的节点对之间的最短路径，根据该最短路径记录所有的源-第二跳节点对的集合；然后通过在网络中部署 SDN 节点，计算所有源-第二跳节点对的不包含最优下一跳的路径，以使该方法能够解决网络中可能出现的单节点故障问题。为了便于读者理解，我们用图 3 来解释本文的思路。图 3 中，源 s 到目的 d 的最短路径为$(s,a,\cdots,d)$，假设 SDN 节点为 c，则该节点到节点 f 的最短路径不经过节点 a，或者节点 c 的邻居 e 到 f 的最短路径不经过节点 a。网络中没有故障时，当有报文从源 s 被转发到目的 d 时，该报文的转发路径为$(s,a,\cdots,d)$。当节点 a 出现故障时，源 s 将把报文首先转发给 SDN 节点 c，然后 SDN 节点 c 或者节点 c 的邻居节点 e 将该报文转发到节点 f，节点 f 就会把报文转发给目的节点 d。

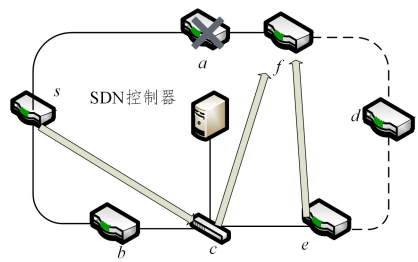

图 3 GARPHSDN 和 IPGARPHSDN 的核心思想

Fig. 3 Key idea of GARPHSDN and IPGARPHSDN

4.1 贪心算法

本节将介绍如何采用贪心算法来解决上述问题。算法 1 介绍了 GARPHSDN(Greedy Algorithm for Routing Protection Based on Hybrid Software Defined Networks)是如何

运行的。首先设置部署 SDN 节点集合的初始值 $M=\emptyset$，设置故障保护率的初值 $R(G,M)=0$，计算所有源-目的节点对之间的最短路径，然后将所有源-第二跳节点对存储在变量 $L=\{(s,d),s,d\in V\}$ 中(见算法 1 中的第 1－3 行)。为了获得部署 SDN 节点的集合 M，算法 1 需要执行一系列的迭代过程，直到节点故障保护率 $R(G,M)=1$ 或者部署 SDN 节点的集合 $M=V$ 成立。函数 $nei(V)$ 的作用是在集合 V/M 中随机选择一个节点 v 来部署 SDN 技术。函数 $\arg\max_{v\in nei(V)} R(G,M\cup v)$ 的功能是，计算出保护的集合 L 中节点对数量最多时对应的节点 k。节点 k 是源-目的节点对 s 和 d 的 SDN 节点必须满足的条件，即节点 $dn(s,k)$ 不在节点 s 到节点 k 的最短路径上，并且节点 k 至少有一个邻居节点到节点 d 的最短路径不包括链路节点 $dn(s,k)$。然后将节点 k 加入到集合 M 中，更新故障保护率(见算法 1 中的第 4－8 行)。最后返回部署 SDN 节点的集合 M(见算法 1 中的第 9 行)。

算法 1 GARPHSDN

Input: G=(V,E)

Output: M

1. M=∅
2. R(G,M)←0
3. L={(s,d),s,d∈V}
4. While R(G,M)<1 and M≠V do
5. k←arg max R(G,M∪v)
6. M←M∪k
7. 计算故障保护率 R(G,M)
8. End While
9. Return M

4.2 IPGARPHSDN 算法

GARPHSDN 算法是一种典型的贪心算法，为了从网络中选择一个节点来部署 SDN 技术，该算法需要经过数次的迭代过程，因此该算法的时间复杂度较高。为了降低 GARPHSDN 的时间复杂度，使算法更容易在实际网络中部署，本文提出了一种改进的贪心算法 IPGARPHSDN(Improved Greedy Algorithm for Routing Protection Based on Hybrid Software Defined Networks)，用于降低算法的复杂度。算法 2 详细描述了 IPGARPHSDN 算法的具体执行过程。首先计算出网络中所有源-第二跳节点对集合 $L=\{(s,d),s,d\in V\}$(见算法 2 中的第 1 行)。然后针对集合 $(s,d)\in L$ 中的任意源-目的节点对，根据定义 1 的规则计算每个节点对之间的所有 SDN 节点 $D(s,d)$，统计网络中所有节点的 $\sum_{(j,k)\in V} y(i,j,k)$，$i\in V$，其中 $y(i,j,k)$ 表示节点 i 是否是源-目的节点对 j 和 k 的 SDN 节点，如果节点 i 是源-目的节点对 j 和 k 的 SDN 节点，则该值为 1，否则为 0。设置部署 SDN 节点集合的初始值 $M=\emptyset$(见算法 2 中的第 2－4 行)。下面给出一个循环过程，直到节点故障保护率 $R(G,M)=1$ 或者部署 SDN 节点的集合 $M=V$ 成立。每次选择一个 $\sum_{(j,k)\in V} y(i,j,k)$ 的值最大的节点 m 部署 SDN 技术，更新集合 M(见算法 2 中的第 6－7 行)，因为我们规定每一对源-目的节点只选择唯一的 SDN 节点，所以对于集合 $(s,d)\in L$ 中的任意源-目的节点对，如果 $m\in D(s,d)$，则该源-目的节点对之间的 SDN 节点就确定了，不必再为其计算 SDN 节点，并且将 $D(s,d)$ 中的内容清空(见算法 2 中的第 8－12 行)；然后更新除去节点 m 的其他所有节点对应的 $\sum_{(j,k)\in V} y(i,j,k)$，并计算故障保护率(见算法 2 中的第 13－14 行)。最终返回部署 SDN 节点的集合 M(见算法 2 中的第 16 行)。

算法 2 IPGARPHSDN

Input: G=(V,E)

Output: M

1. L={(s,d),s,d∈V}
2. 计算集合 L 中每个节点对之间所有的 SDN 节点 D(s,d)
3. 计算网络中所有节点 $\sum_{(j,k)\in V} y(i,j,k)$
4. M=∅
5. While R(G,M)<1 and M≠V do
6. m=max $\sum_{(j,k)\in V} y(i,j,k)$
7. M←M∪m
8. For (s,d)∈L
9. If m∈D(s,d) then
10. 清空 D(s,d)
11. End If
12. End For
13. 更新网络中除去节点 m 的所有节点 $\sum_{(j,k)\in V} y(i,j,k)$
14. 计算故障保护率 R(G,M)
15. End While
16. Return M

4.3 算法讨论

本节将从理论的角度来对 GARPHSDN 算法和 IPGARPHSDN 算法的时间复杂度进行分析。

定理 1 算法 GARPHSDN 的时间复杂度为 $O(|V|^5)$。

证明：为了计算最终需要部署 SDN 技术的节点，算法最多需要执行 $|V|$ 次函数 $\arg\max_{v\in nei(V)} R(G,M\cup v)$，该函数需要计算网络中所有节点对之间部署 SDN 的情况，算法复杂度为 $O(|V|^5)$。因此 GARPHSDN 的时间复杂度为 $O(|V|^5)$。

定理 2 IPGARPHSDN 算法的时间复杂度为 $O(|V|^2)$。

证明：该算法中第 2 行的时间复杂度为 $O(|V|^2)$，第 5－15 行的时间复杂度为 $O(|V|^2)$，因此该算法的时间复杂度为 $O(|V|^2)$。

从上述分析可以看出，IPGARPHSDN 算法的时间复杂度比 GARPHSDN 算法的复杂度降低了 3 个数量级，大大降低了算法的计算开销，因此该算法更容易在实际中进行部署。

5 实验及结果分析

本节将通过实验来模拟 GARPHSDN 算法和 IPGARPHSDN 算法的性能，通过 SDN 节点数量、故障保护率、计算开销以及路径拉伸度等指标对其性能进行评估。显然，网络中部署的 SDN 节点的数量越少，其部署开销就越小。如果某种算法对应的故障保护率为 1，则说明该算法能够对网络中所有可能的单节点故障情况加以应对，否则该算法无法保护网络中的某些故障情形。第 4 节从理论上分析了 GARPHSDN 算法和 IPGARPHSDN 算法的时间复杂度，本节将利用算法的实际计算时间来比较 GARPHSDN 算法和 IPGARPHSDN 算法的计算开销。本文在实验中比较了 GARPHSDN 和 IPGARPHSDN 的路径拉伸度。路径拉伸度

直接影响着网络的时延和路径开销，因此我们希望算法的路径拉伸度尽可能地小。下文将对算法运行的网络拓扑结构进行阐述，然后描述实验结果，同时对实验结果进行详尽的分析。

5.1 网络拓扑

为了评价 GARPHSDN 算法、IPGARPHSDN 算法的性能，我们在大量的拓扑上运行了上述两种算法。本文的实验拓扑包括以下 3 种类型：

(1)真实网络拓扑结构[22-23]。本文从该类拓扑中选择了 4 个真实网络拓扑，参数如表 1 所列。

表 1 真实网络拓扑

Table 1 Real network topology

网络拓扑	结点数量	链路数量
Abilene	11	14
Cernet	14	16
TORONTO	25	55
USLD	28	45

(2)Rocketfuel[24]测量的拓扑结构。我们从该类拓扑中选择了 5 个测量拓扑结构，具体参数如表 2 所列。

表 2 Rocketfuel 拓扑

Table 2 Rocketfuel topology

AS 号码	结点数量	链路数量
1 221	108	153
1 755	87	161
1 239	315	972
3 257	161	328
3 967	79	147

(3)利用模拟软件 Brite[1)] 生成的拓扑结构。Brite 使用 Waxman 模型，拓扑结构中节点的数量大于 100 且小于 500，beta 的参数值设为 0.2，alpha 的参数值设为 0.15，设置网络的平均度参数介于 2 至 4 之间，网络中节点的分布服从重尾分布，设置链路的带宽参数介于 10 到 1 024 之间，链路的代价和链路带宽互为倒数。

5.2 SDN 节点的数量

本节通过在传统网路中部署 SDN 节点来应对网络中所有可能出现的单节点故障情形，但是部署 SDN 节点需要额外的开销，部署的 SDN 节点数量越少，网络的额外开销就越小。图 4 和图 5 给出了不同网络拓扑中部署 SDN 节点的数量的结果，在图中 Brite(m,n)表示利用 Brite 软件生成的拓扑结构，节点数量为 m，网络平均度为 n。

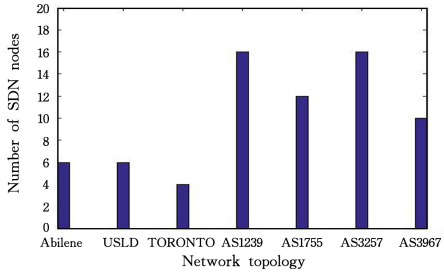

图 4 真实拓扑和 Rocketfuel 中 SDN 节点的数量

Fig. 4 Number of SDN nodes in real and Rocketfuel topologies

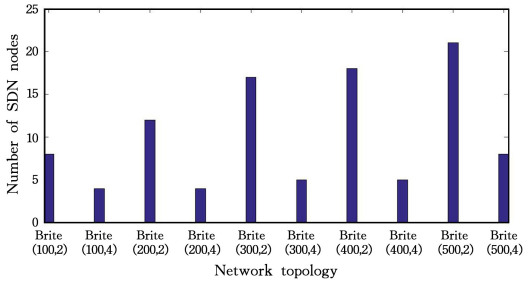

图 5 Brite 拓扑中 SDN 节点的数量

Fig. 5 Number of SDN nodes in Brite topology

从图 4 和图 5 中可以看出，除 Abilene 外，在所有的网络中只需要部署很少的 SDN 节点就可以达到故障全保护的目的。例如，在 USLD 网络中，需要部署的 SDN 节点数量为 6，结点总数为 28，其 SDN 节点数量占总结点数量的 21%。在 Brite(100,2)网络中，部署的 SDN 节点数为 8，占总结点数的 8%。从上述结果可知，在稀疏图中，部署 SDN 节点的数量较多，这是因为 Abilene 拓扑较小，并且节点的度较小；而在稠密图中，部署 SDN 节点的数量相对较少。

5.3 故障保护率

本节将利用故障保护率来衡量 GARPHSDN 算法和 IPGARPHSDN 算法应对故障的能力。图 6 给出了两种算法在不同网络拓扑中对应的故障保护率，从该数据可以看出，GARPHSDN 算法和 IPGARPHSDN 算法对应的故障保护率均为 100%，即他们可以应对网络中所有可能出现的单节点故障情形。这是因为 GARPHSDN 算法和 IPGARPHSDN 算法在传统网络中部署了小部分 SDN 节点，使得报文的转发更具有灵活性。当报文在转发的过程中遇到故障元素时，该节点必定会将报文转发给特定的 SDN 节点，而 SDN 节点最终会将报文顺利转发至目的节点。

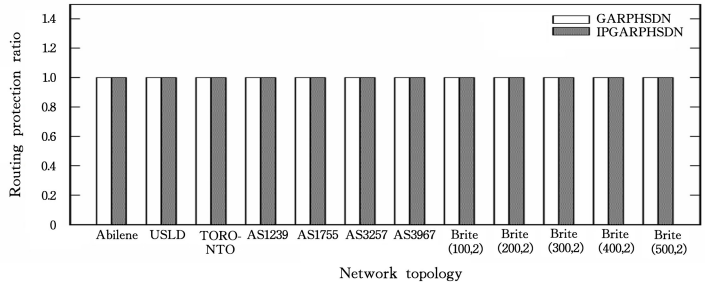

图 6 两种算法在不同网络拓扑中的故障保护率

Fig. 6 Routing protection ratio on different network topologies

1) http://www.cs.bu.edu/brite/

5.4 计算开销

第4节从理论的角度出发对上述两种算法的时间复杂度进行了详细的分析。本节将从不同算法的实际计算时间的角度来衡量他们的计算开销。本实验运行在一台内存为4GB的电脑上，该电脑的处理器为4核、主频为3.3GHz。表3列出了GARPHSDN和IPGARPHSDN在真实网络拓扑和Rocketfuel测量拓扑中的计算开销。从表3可以看出，IPGARPHSDN的计算开销远远低于GARPHSDN的计算开销，例如，在Abilene网络中，IPGARPHSDN的计算开销是GARPHSDN的1/16，在AS1239拓扑中，IPGARPHSDN的计算开销仅仅是GARPHSDN的1/220。因此，IPGARPHSDN大大降低了算法的计算开销，更容易在实际中部署。这是因为GARPHSDN每次都选择性能最优的节点部署SDN技术，这需要大量的计算过程，而IPGARPHSDN利用之前计算出来的结果来决定下一次部署SDN技术的节点，而不是每次从头开始计算。

表3 计算开销

Table 3 Computation overhead

网络拓扑	GARPHSDN/ms	IPGARPHSDN/ms
Abilene	0.0089	0.00053
Cernet	0.0091	0.00078
USLD	0.0142	0.0063
TORONTO	0.00124	0.0063
AS1239	325.26	1.475
AS1755	8.974	0.619
AS3257	40.261	3.679
AS3967	2.681	0.0763

图7给出了GARPHSDN和IPGARPHSDN在Brite拓扑结构中的计算开销。

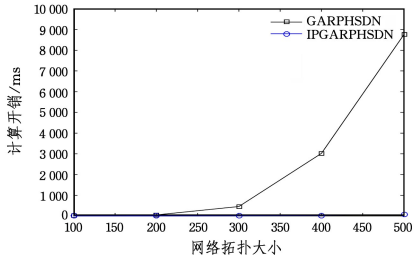

图7 不同算法在公开拓扑中的执行时间

Fig. 7 Computation time on public topologies of different algorithms

从图中可以看出，随着网络拓扑大小的增加，GARPHSDN的计算开销随之增加，当网络拓扑大小为500时，GARPHSDN的计算开销接近9000ms，而IPGARPHSDN的计算开销基本不发生变化，此时GARPHSDN的运行时间大约是IPGARPHSDN的9倍。由此可见，IPGARPHSDN的计算开销远远小于GARPHSDN的计算开销。

上述实验的计算开销是在一台PC机上运行两种算法得到的结果。为了进一步验证两种算法部署在实际网络中的运行情况，在实验中首先在28台电脑上安装路由器软件Quagga和Click来模拟真实的路由器，然后分别按照Abilene，Cernet，USLD和TORONTO拓扑结构相连并运行GARPHSDN算法和IPGARPHSDN算法。图8给出了两种算法的计算开销对应的结果。从图8中可以看出，该实验对应的计算开销和在一台PC机上的计算开销基本是相同的。

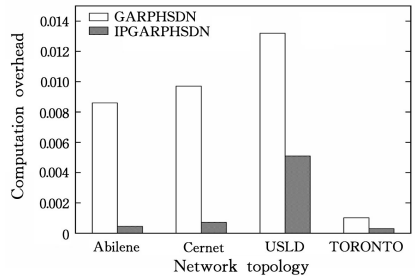

图8 在28台电脑上的计算开销

Fig. 8 Computing overhead on 28 computers

5.5 路径拉伸度

本节将路径拉伸度作为衡量GARPHSDN和IPGARPHSDN的路径开销的指标。本文对路径拉伸度进行定义，即网络出现故障时，利用GARPHSDN和IPGARPHSDN计算出的路径代价和利用OSPF计算的最短路径的代价的比值。图9给出了GARPHSDN和IPGARPHSDN在3种模拟拓扑中的路径拉伸度，从图中可以看出，GARPHSDN和IPGARPHSDN的路径拉伸度是一样的，并且两者的路径拉伸度都小于2，因此不会引入过多的路径开销。例如，在Abilene中，两种算法的路径拉伸度均为1.145，与OSPF收敛完成后计算出的最短路径的代价几乎一致。

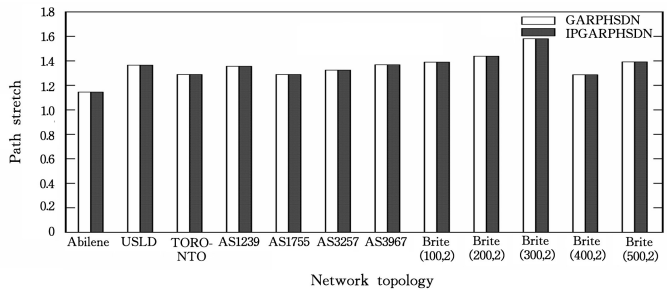

图9 两种算法在模拟拓扑中的路径拉伸度

Fig. 9 Path stretch of two algorithms in simulated topology

结束语 为了给SDN网络中所有可能出现的单个节点故障提供解决方案，我们把在传统网络中部署SDN节点的问题进行了抽象处理，将其抽象为一个整数规划问题，并提出GARPHSDN和IPGARPHSDN这两种算法来解决该问题。然而，GARPHSDN的计算复杂度较高，不适合在实际网络中部署。IPGARPHSDN具有较小的计算开销，因此IP-

GARPHSDN 是一种具有较强竞争力的方案。实验结果表明，仅需要将传统网络中的少数节点升级为 SDN 节点，即可以保证 IPGARPHSDN 算法能够应对所有可能出现的单节点故障情形。然而，本文算法只能保护网络中的单节点故障，无法应对网络中并发节点故障的情形。因此，下一步我们将重点研究如何在混合 SDN 网络中设计一种路由保护方法来应对混合 SDN 网络中并发节点故障的情形。

参 考 文 献

[1] GENG H J，ZHANG H，SHI X G，et al. Efficient computation of loop-free alternates[J]. Journal of Network and Computer Applications，2020，151(5)：1-12.

[2] KUN Q，JIN Z，XIN W，et al. Efficient Recovery Path Computation for Fast Reroute in Large-scale Software Defined Networks [J]. IEEE Journal on Selected Areas in Communications，2019，37(8)：1755-1768.

[3] GENG H J，SHI X G，WANG Z L. Intra-domain Routing Protection Scheme Based on Minimum Intersection Paths[J]. Journal of Software，2020，31(5)：1536-1548.

[4] DONG S. Survey on Software Defined Networks Security[J]. Computer Science，2021，48(3)：295-306.

[5] ZHANG J，WANG H，LUO S T，et al. Hybrid Software Defined Network Energy Efficient Routing Algorithm Based on Genetic Algorithm[J]. Computer Science，2020，47(6)：236-241.

[6] YANG Y，XU M，LI Q. Fast Rerouting Against Multi-Link Failures Without Topology Constraint[J]. IEEE/ACM Transactions on Networking，2018，26(1)：384-397.

[7] GENG H，SHI X，WANG Z，et al. A hop-by-hop dynamic distributed multipath routing mechanism for link state network [J]. Computer Communications，2018，116：225-239.

[8] KWONG K W，GAO L，ZHANG Z L. On the feasibility and efficacy of protection routing in IP networks[J]. IEEE/ACM Transactions on Networking，2011，19(5)：1543-1556.

[9] ATLAS A，ZININ A. RFC 5286，Basic specification for ip fast reroute：Loop-free alternates[S]. IETF，2008.

[10] ENYEDI G，RÉTVÁRI G，SZILÁGYI P，et al. IP Fast ReRoute：Lightweight Not-Via without Additional Addresses [C]// International Conference on Computer Communications (INFOCOM). IEEE Computer Society，2009：2771-2775.

[11] ZHENG J，XU H，ZHU X，et al. We've Got You Covered：Failure Recovery with Backup Tunnels in Traffic Engineering [C]// International Conference on Network Protocols (ICNP). IEEE Computer Society，2016：1-10.

[12] RÉTVÁRI G，CSIKOR L，TAPOLCAI J，et al. Optimizing IGP link costs for improving IP-level resilience[C]// International Conference on the Design of Reliable Communication Networks (DRCN). IEEE Computer Society，2011：62-69.

[13] RÉTVÁRI G，TAPOLCAI J，ENYEDI G，et al. Ip fast reroute：Loop free alternates revisited[C]// International Conference on Computer Communications (INFOCOM). IEEE Computer Society，2011：2948-2956.

[14] BRAUN W，MENTH M. Loop-Free Alternates with Loop Detection for Fast Reroute in Software-Defined Carrier and Data Center Networks[J]. Journal of Network and Systems Management，2016，24(3)：470-490.

[15] HE L，LIN R，YU S，et al. Congestion-Aware Local Reroute for Fast Failure Recovery in Software-Defined Networks[J]. IEEE/OSA Journal of Optical Communications & Networking，2017，9(11)：934-944.

[16] ZHANG X，CHENG Z，LIN R P，et al. Local Fast Reroute With Flow Aggregation in Software Defined Networks[J]. IEEE Communications Letters，2017，21(4)：785-788.

[17] ZHU Z，LI Q，XIA S，et al. CAFFE：Congestion-Aware Fast Failure Recovery in Software Defined Networks[C]// International Conference on Computer Communication and Networks (ICCCN). IEEE Computer Society，2018：1-9.

[18] ZHENG J，XU H，ZHU X，et al. Sentinel：Failure Recovery in Centralized Traffic Engineering[J]. IEEE/ACM Transactions on Networking，2019，27(5)：1859-1872.

[19] GENG H J，ZHANG W，YIN X. Routing Protection Algorithm Based on Hybrid Software Defined Network[J]. Computer Engineering，2020，46(6)：209-215.

[20] GENG H J，SHI X G，WANG Z L，et al. Intra-domain Routing Protection Scheme Based on Minimum Intersection Paths[J]. Journal of Software，2020，31(5)：1536-1548.

[21] YANG Z，YEUNG K L. SDN Candidate Selection in Hybrid IP/SDN Networks for Single Link Failure Protection[J]. IEEE/ACM Transactions on Networking，2020，28(1)：312-321.

[22] NETWORK A B. Advanced networking for research and education[OL]. http://abilene.internet2.edu，2003.

[23] ANTONAKOPOULOS S，BEJERANO Y，KOPPOL P. Full Protection Made Easy：The DisPath IP Fast Reroute Scheme [J]. IEEE/ACM Transactions on Networking，2015，23(4)：1229-1242.

[24] SPRING N，MAHAJAN R，WETHERALL D，et al. Measuring isp topologies with rocketfuel[J]. IEEE/ACM Transactions on Networking，2004，12(1)：2-16.

GENG Hai-jun，born in 1983，Ph.D，lecturer，is a member of China Computer Federation. His main research interests include future Internet architecture and routing algorithm.

（责任编辑：柯颖）