一种 SDN 中基于 SR 的多故障恢复与规避机制

黄建洋,兰巨龙,胡宇翔,马 腾
(国家数字交换系统工程技术研究中心,河南郑州 450002)

摘 要: 针对 SDN 中的多故障恢复问题,提出了一种基于分段路由的多故障恢复与规避机制. 为实现故障的快速恢复,在数据平面通过 SR 技术预部署一个可对多故障提供快速恢复能力的链路环备份系统;为防止故障恢复时造成部分链路负载过重引发再故障,提出最优化流量平滑算法对负载超过设定阈值的链路实施流量均衡处理. 实验结果表明,与已有的 SDN 多故障恢复机制 OAM-Based 和 Fast Failover 相比,所提机制的平均故障恢复成功率分别提高了 6.7% 和 8.5%,且控制器通信开销仅为传统 OpenFlow 方案的 9%,同时,所提机制对故障恢复时导致的链路负载过重问题进行了处理,从故障规避的角度为网络提供保护.

关键词: 软件定义网络; 分段路由; 多故障恢复; 故障规避; 流量均衡
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2017)11-2761-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2017.11.025

A Multi-fault Recovery and Avoidance Mechanism of Software-Defined Network Based on Segment Routing

HUANG Jian-yang, LAN Ju-long, HU Yu-xiang, MA Teng
(*National Digital Switching System Engineering & Technology Research Center, Zhengzhou, Henan* 450002, *China*)

Abstract: Considering the multi-fault problem in Software-Defined Network (SDN), a segment routing based multi-fault recovery and re-breakdown avoidance mechanism is proposed. In order to achieve fast recovery of failures, a linkage-loop backup system aiming at fast recovery for multi-fault problem is proposed with SR technology in dataplane. To avoid second failure caused by breakdown in some links due to the overload problem in failure recovery, an optimal traffic smoothing algorithm is proposed to balance the traffic among links where the traffic goes beyond the threshold. Experiments result show that compared with the schemes OAM-Based and Fast Failover, the proposed scheme improves the success rate of failure recovery by 6.7% and 8.5%, and the cost in controller is only 9% of the traditional OpenFlow scheme. At the same time, the proposed scheme also deals with the overload problem in failure recovery, which protects the network in failure avoidance scope.

Key words: software-defined network; segment routing; multi-fault recovery; fault avoidance; traffic balance

1 引言

随着实时多媒体应用和关键性业务规模的爆炸式增长,服务可靠性已成为当前网络部署和管理研究领域的重要议题. 软件定义网络(Software-Defined Network, SDN)是一种新兴的基于软件的网络结构及技术,由于其具有松耦合的控制平面和数据平面、支持集中化的网络状态控制和底层网络设施对上层应用透明等特征,目前已开始被应用到多种现实网络中,例如:光网络[1-3]、运营商网络和工业网[4]等. 这些网络中的节点或链路故障,都会造成大量用户数据的丢失,所有与其相关的传输服务都将失效,影响十分严重. 在 SDN 网络架构下,如何利用控制平面强大的集中管控能力,以及数据平面快速的数据转发能力实现

收稿日期: 2016-04-21;修回日期:2017-03-06;责任编辑:马兰英
基金项目:国家 973 计划资助项目(No. 2012CB315901, No. 2013CB329104);国家自然科学基金(No. 61572519, No. 61502530);国家 863 计划资助项目(No. 2013AA013505, No. 2015AA016102)

网络故障快速恢复的同时兼顾资源的合理调配,已成为业界研究的重点.

现有研究中,文献[5]设计了一种基于 OpenFlow 的运营商网络故障恢复机制,但每次故障发生时都需要交换机向控制器通知拓扑变化情况,带来较大的故障恢复时延和控制器开销. 与此同时,文献[6]将该模式应用到了 IP 网络,但并没有解决故障恢复时延和控制器开销较大的问题. 文献[7]设计了一种由数据平面完成预计算备份路径切换的恢复机制,其在各交换节点处安装 OAM 工具,故障发生时由这些 OAM 工具完成备份路径的切换,但由于预计算的备份路径数量较少,该方案只能恢复数量较少的故障. 文献[8]对 OpenFlow1. 1 协议进行了扩展,采取数据平面各交换机间发送周期性探针消息的方式对每条流进行监控,实现了故障的快速发现和通告. 文献[9]基于 OpenState 也提出了一种数据平面的快速故障恢复机制,该机制利用回溯包进行故障信息通告,重路由节点在接收到回溯包后对流进行传输路径切换,解决了路由环路问题且在一定程度上减小了故障恢复时延. 文献[10]为了避免故障发生时的带宽浪费,由数据平面停止所有与故障链路相关的传输流量,之后由控制器完成路由恢复. 为实现快速故障恢复同时减少控制器开销,新版本的 OpenFlow 协议[11]提出了一种数据平面快速失效备援机制,该机制在故障发生时由交换机执行第一个有效活动端口动作桶中的指令,可实现快速的局部路由恢复.

针对网络中的故障恢复问题,文献[12]提出要做好两方面的工作:(1)通过流量工程对故障恢复后的通信负载进行均衡;(2)对快速故障恢复方案进行互操作协议测试,重新设计路由器的体系结构. 为了将 SDN 网络的故障恢复时延控制在 50ms 以内,几乎当前所有的方案都采用数据平面链路备份的方式,但都易造成部分核心链路负载过重而引发再故障. 现有的研究主要解决了故障发生时的重路由问题,却没有考虑到路由恢复后的再故障规避问题.

基于以上分析,本文提出一种 SDN 中基于分段路由(Segment Routing, SR)[13,14]的多故障恢复与规避(Multiple Fault Recovery and Avoidance, MFRA)机制及其相关算法. 本文主要的创新点有:(1)提出网络拓扑嵌入算法和备份环选取算法,为数据平面中的每条链路计算一个最小环进行链路保护;(2)利用 SR 技术实现数据平面链路最小环的部署,可在故障发生时不需要路由聚合完成快速路由恢复;(3)提出基于 SR 的最优化流量平滑算法,对故障恢复时出现的高负载链路实施流量均衡处理.

2 概念描述

2.1 SR 介绍

为了使当前的 IP/MPLS 网络变得更加面向服务和高效,IETF 于 2013 年提出了 SR 的概念. SR 的核心是源路由,即由源节点决定数据转发路径并将其保存在数据包头部,数据转发时,中间节点只需根据包头保存的路径信息对数据包进行转发. SR 分别用节点段标识(Node-Segment Identifier, Node-SID)和邻接段标识(Adjacency-Segment Identifier, Adj-SID)表示网络中的节点和链路,数据转发路径则由一条段标识(Segment Identifiers, SIDs)序列表示. 数据转发时,中间节点只需依据包头最外部的 SID 对数据包实施操作.

为详细阐述 SR 的运行机制,以图 1 为例进行说明. 图中 Node-SID 被设定为节点环回地址前缀,是一种全局适用并通过扩展 IGP 协议分发的段标识;Adj-SID 则由链路两端节点产生,是一种局部适用的段标识. 当分发完 SID 并完成最短路径计算后,每个节点处会存储一个 SR 转发规则表,如图 1 所示. 转发规则表第一列表示节点/邻接段的 SID 号,第二列表示相应的处理动作,节点会根据包头 MPLS 标签栈顶的 SID 号对数据包作出处理. 数据传输时,源节点将计算的转发路径以 SIDs 表示并入栈,中间节点仅根据栈顶 SID 号对数据包实施操作. 假设图 1 中链路{*B-D*}和{*C-E*}都发生拥塞,节点 *A* 需向 *G* 通信,此时数据流只能沿路径{*A-C-D-F-G*}进行转发,则 *A* 节点将以表 1 的方式将数据包发送到 *G*.

表 1 SR 控制数据包转发过程

节点	栈顶 SID	操作
A	∅	{1001,3001,107}入栈
A	1001	出栈,端口 1
C	3001	出栈,端口 2
D	107	端口 3
F	107	端口 1
G	107	出栈
G	∅	接收数据包

2.2 图论相关概念

首先,描述本文所使用的图论相关基本概念,作为后续多故障恢复方案及相关算法的基础.

定义 1 一个网络图 *G* 若能在平面 *S* 上画出,使任意两条链路在非节点处不相交,则称 *G* 可以在 *S* 上嵌入. 可嵌入的网络图称为可嵌网络图,否则称为非可嵌网络图. 如图 2(*b*)为对图(*a*)进行网络图嵌入后的结果.

定义 2 设 *H* 是 *G* 的子网络图,满足下列条件的 *G-E*(*H*)的连通子网图 *B* 为 *G* 加在 *H* 上的桥,其中 *E*(*H*)为 *H* 中的所有链路:

(1)*B* 中任意两条链路之间都存在路径,其中间节

点均不属于 H.

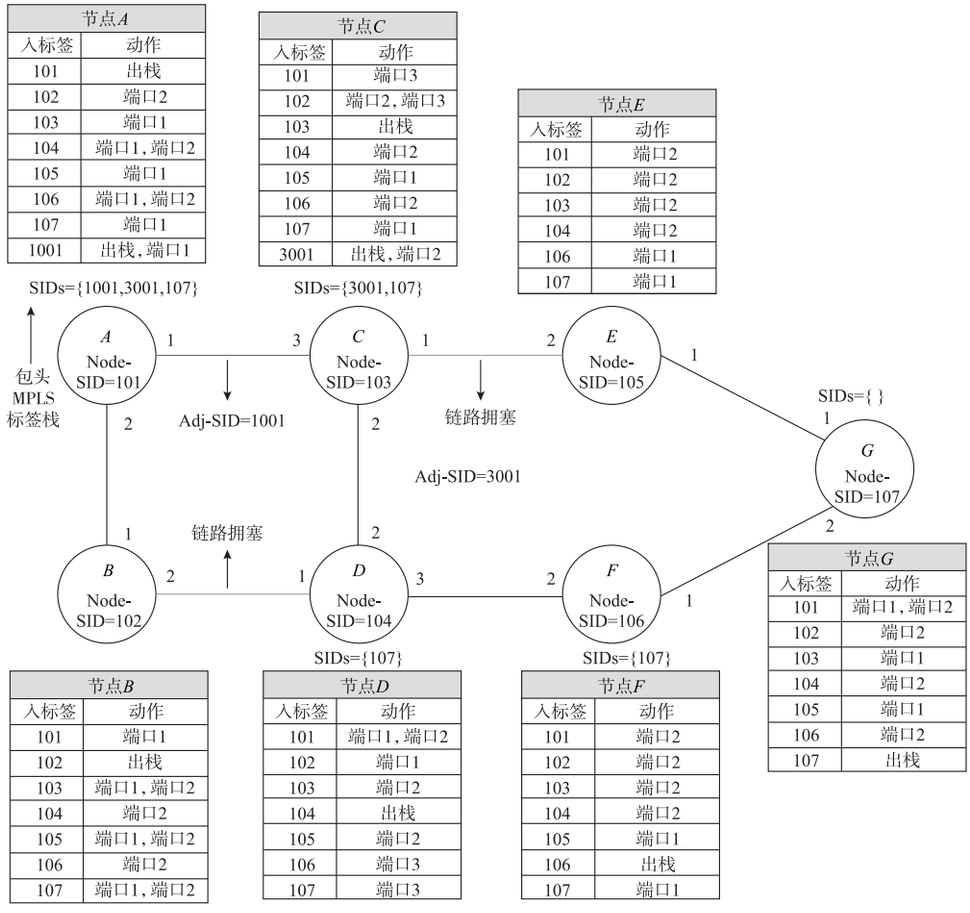
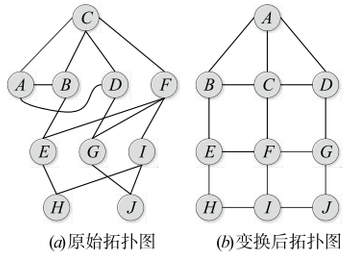


图1　SR运行机制



(a)原始拓扑图　(b)变换后拓扑图

图2　网络图的嵌入

(2)B 是满足条件(1)的 $G-E(H)$ 的极大连通子网图,即任何以 B 为真子网图的 $G-E(H)$ 的连通子网图都不满足(1).

定义 3　设 H 是 G 的一个可嵌子图,$\widetilde{H}$ 是 H 的嵌入图,B 是 G 中 H 上的一个桥,f 是 $\widetilde{H}$ 的一个面,如果 B 的附着点全部在 f 的边界上,则称 B 在 $\widetilde{H}$ 的面 f 上是可展的,并把 $\widetilde{H}$ 中 B 是可展的那些面的集合记为 $F(B,\widetilde{H})$.

根据图可嵌入性的必要条件,可得如下定理,其正确性已在文献[15]处证明.

定理 1　若 G 是一个可嵌网络图,H 是 G 的一个可嵌网络子图,$\widetilde{H}$ 是 H 的网络图嵌入,且能在 $\widetilde{H}$ 的基础上把 G 全部嵌入在一个曲面上,则对 $\widetilde{H}$ 上的每一个桥 B,恒有

$$F(B,\widetilde{H})\neq\varnothing \tag{1}$$

定义 4　(链路最小环)从链路 l_{ij} 的端点 i 出发,经过一条逆时针最短路径途经 j 回到 i 形成的环称为该链路的链路最小环,记为 $C_{l_{ij}}$.

链路顺时针最小环称为链路最小逆环,记为 $\overline{C}_{l_{ij}}$. 图2(b)中链路 l_{BC} 的链路最小环为 C_{BEFCB},链路最小逆环为 C_{BACB}.

定义 5　(拓扑最大环)从网络拓扑 $G(V,E)$ 中任意一条边缘链路 l_{ij} 的端点 i 出发,通过逆时针途经 G 的所有边缘链路回到 i 而形成的环称为拓扑最大环,记为 C_G.

顺时针方向的最大环称该网络的拓扑最大逆环,记为 $\overline{C}_G$. 图2(b)中网络的拓扑最大环为 $C_{ADGJIHEBA}=C_{DGJIHEBAD}=\cdots$,拓扑最大逆环为 $C_{ABEHIJGDA}=C_{BEHIJGDAB}=\cdots$.

3　多故障恢复与再故障规避相关算法

3.1　多故障恢复相关算法

网络故障一般可分为节点故障和链路故障,由于

节点故障会导致与其直连的所有链路失效,因此本文将节点故障看作多链路故障. 为了降低故障恢复时延,与文献[7,9,11]相同,采用数据平面链路备份的方式完成快速路由恢复. 同时,为处理多链路故障且防止环路产生,提出在数据平面为每条链路寻找一个最小环进行备份,同时对所有最小环进行合理组织,产生一个可对多故障提供有效恢复能力的环备份系统.

由于一般网络拓扑的呈现方式可能在非节点处有链路交叉,很难参照拓扑为链路寻找最小环. 因此,本文在选取链路备份环路之前,对网络拓扑实施网络图嵌入处理. 由定理 1 作为判断拓扑是否可嵌入的终止条件,提出如下网络拓扑嵌入算法.

算法 1 网络拓扑嵌入算法

输入:网络拓扑 G;
输出:G 的嵌入图 $\bar{G}$;
1. $G_1 = RandSelectCircle(G)$;
2. $i = 1$;
3. $\bar{G}_1 = Embed(G_1)$; //对 G_1 进行平面嵌入得到 $\bar{G}_1$
4. while$(E(G)/E(G_i)! = \varnothing)$ do
//将 G 中链路加入 G_i,直至无链路剩余
5. $S_{AllB} = Allbridge(G, \bar{G}_i)$;
6. for $(j = 1; j <= GetNumber(S_{AllB}); j++)$ do
7. $B_j = GetBridge(S_{AllB})$;
8. $F(B_j, \bar{G}_i) = GetExtendface(B_j, \bar{G}_i)$;
//获取 B_j 在 $\bar{G}_i$ 上可展的面的集合
9. if$(F(B_j, \bar{G}_i) == \phi)$
10. $EndAlgorithm()$;
//不满足定理 1 则终止算法
11. endif;
12. endfor;
13. $B = RandSelectItem(S_{AllB})$;
//随机选取桥,$|F(B, \bar{G}_i)| = 1$ 的优先选取
14. $f = RandSelectItem(F(B, \bar{G}_i))$;
15. $P = SelectPath(B, \bar{G}_i)$;
//选取路径 $P \subset B$,且 P 的起点和终点是 B 在 $\bar{G}_i$ 上的附着点
16. $G_{i+1} = Combine(G_i, P)$; //$G_{i+1}$ 等于 G_i 中加入路径 P
17. $\bar{G}_{i+1} = Embed(\bar{G}_{i+1})$;
18. $i = i + 1$;
19. endwhile;

算法 1 通过逐链路的方式完成网络图的嵌入,每次 while 循环中,设 G 中 $\bar{G}_i$ 上的平均桥个数为常数 M,函数 $Allbridge(G, \bar{G}_i)$ 的时间复杂度为 $O(M)$,则算法 1 的时间复杂度为 $O(ME)$,又因为 M 是远小于 E 的常数,因此算法 1 的时间复杂度为 $O(E)$.

为应对多链路故障,在经算法 1 得到的拓扑嵌入图的基础上,提出一种备份环选取算法. 为避免链路之间出现相互备份,对拓扑中的链路通过其链路最小环进行备份,对不存在链路最小环的链路通过拓扑最大环进行备份.

算法 2 备份环选取算法

输入:G 的嵌入图 $\bar{G}$;
输出:所有链路的链路备份环;
1. $\bar{C}_G = GetTopoMaxAntiCircle(\bar{G})$
//获取嵌入图 $\bar{G}$ 的拓扑最大逆环
2. For$(i = 1; i <= GetNumber(V(\bar{G})); i++)$ do
3. $n = GetNode(\bar{G}, i)$;
//获取嵌入图 $\bar{G}$ 的第 i 个结点
4. $Polar[n] = SortPolar(\bar{G}, n)$;
//获取节点 n 的出链路极角序
5. endfor;
6. for$(i = 1; i <= GetNumber(E(\bar{G})); i++)$ do
7. $l = GetLink(\bar{G}, i)$;
8. if $(IsExistCircle(l, Polar) == 1)$
9. $C_l = GetMinCircle(\bar{G}, l)$;
//如果链路 l 的链路最小环存在,获取它
10. $SetBackupCircle(l, C_l)$;
11. else
12. $C_l = \bar{C}_G$;
//不存在链路最小环,用拓扑最大逆环
13. $SetBackupCircle(l, C_l)$;
14. endfor;

上述算法中,第一个 for 循环的时间复杂度为 $O(V)$,第二个 for 循环的时间复杂度为 $O(E)$. 综合起来可知算法 2 的时间复杂度为 $O(E+V)$,又因为 E 远大于 V,故算法 2 的时间复杂度为 $O(E)$.

3.2 再故障规避算法

为避免故障恢复时造成部分链路负载过重引发再故障,本文对负载过重的链路采取基于 SR 的最优化流量平滑处理. 从流量工程的角度来看,将源、目的节点之间的传输路径分两段完成,可获得较大部分的最优化流量均衡效益[16]. 因此本文在利用 SR 方式进行流量调度时,仅在源、目的节点之间增加一个“中转”节点. 建立基于 SR 的最优化流量平滑模型如式 2 所示,模型中用到的符号及含义如表 2 所示.

$$\max \sum_{r \in R} b(r) \sum_{k} X_r^k,$$
$$\text{s. t.} \begin{cases} \sum_{r \in R} g_r^k(l) X_r^k \leqslant c(l) \\ \sum_{k} X_r^k \leqslant 1, \forall r \in R \\ X_r^k \geqslant 0, \forall r \in R, k \in V(G) \end{cases} \tag{2}$$

如果系统无法对流传输服务 r 中的流量进行平滑处理,可认为系统针对 r 的收益为 0,因此式(2)的目标

是最大化流量传输请求. 限制条件中,第一行为链路负载约束;第二行表示对于传输服务 r,对其进行流量平滑调度时只能有一个中转节点;第三行表示系统为传输服务 r 选择节点 k 作为中转节点的概率不能为负.

表 2　最优化流量平滑模型中的符号及含义

符号	含义
r	需接受流量平滑处理的源、目的节点对之间的流传输服务
R	需要接受流量平滑处理的流传输服务集合,$r \in R$
$b(r)$	流传输服务 r 的请求带宽
X_r^k	系统为传输服务 r 选择节点 k 作为中转节点的概率,令 $0 \leqslant X_r^k \leqslant 1$
$g_r^k(I)$	为传输服务 r 选择中间节点 k 作为中转节点时流经链路 l 的流量比例
$c(I)$	链路 l 的负载容量

为了便于求解上述最优化模型,引入对偶变量 $\theta(l)$ 和 $\omega(r)$,得到其对偶规划:

$$\text{s. t.}\begin{cases}\min \sum_r \omega(r) + \sum_l c(l)\theta(l) \\ \omega(r) \geqslant b(r)\left[1 - \sum_l g_r^k(l)\theta(l)\right] \\ \omega(r), \theta(l) \geqslant 0\end{cases} \tag{3}$$

根据上式可得,要满足变量约束条件,则必须有:

$$\sum_l g_r^k(l)\theta(l) \leqslant 1, \forall k, \forall r \tag{4}$$

对于流传输服务 r,若系统接受其服务后仍可满足式(4),则选择最优的传输路径中转节点 k,以 SR 方式对其进行流量平滑处理;否则,系统拒绝其服务请求. 计算时,设定 $\theta(l)$ 为链路 l 的对偶权重值,初始值为 0,每完成一次流传输服务,必须对所有链路的 $\theta(l)$ 值进行更新,参考文献[16]中的启发式算法,可得出 $\theta(l)$ 的更新方式:

$$\theta(l_i) = \frac{1}{n}\frac{e^{\frac{F(l_i)}{c(l_i)}} - 1}{e - 1}, l_i \in E(G) \tag{5}$$

式(5)中的 $F(l_i)$ 表示当前链路 l_i 中的流量值. 同时,基于以上描述,得出如下流量平滑算法:

算法 3　最优化流量平滑算法

```
1.  for (i = 1;i <= GetNumber(E(G~));i ++) do
2.      θ(l_i) = 0;
3.  endfor;
4.  r = GetOverloadService();
      //获取导致链路负载过重的传输服务 r
5.  V = min_{∀k∈V(G)} Σ_{l∈E(G)} g_r^k(l)θ(l);
      //获取基于 SR 的数据传输路径权值
6.  if (V > 1)
7.      Reject(r);
      //权值超过 1 拒绝服务
8.  else
9.      k* = arg min_{∀k∈V(G)} Σ_{l∈E(G)} g_r^k(l)θ(l);
      //提取出最优路径的中转节点
10. for(i = 1; i <= GetNumber(E(G~)); i++) do
11.     θ(l_i) = (1/n)(e^{F(l_i)/c(l_i)} - 1)/(e - 1);
      //更新所有链路对偶权值
12. endfor;
```

算法 3 的复杂度为 $O(VE)$,其中 V 为网络拓扑中节点个数,E 为链路个数.

4　实验验证与结果分析

为验证本文所提多故障恢复与规避(MFRA)机制的性能情况,将其与相关方案在故障恢复成功率、控制器通信开销和故障规避能力三个性能指标上进行对比实验. 实验中,我们选用了三个网络拓扑(具体参数如表 3 所示).

表 3　测试拓扑

网络拓扑	节点数	链路数	链路数/节点数
Abilene	11	14	1.273
Redlris	18	30	1.667
Cisco	76	160	4.21

实验验证时,控制平面选用实现了多故障恢复与故障规避功能的 ONOS,数据平面则用 Mininet 软件虚拟出所有网络节点. 实验中,通过设置故障发生频率及平均持续时间来模拟多链路故障. 链路故障的产生服从均值为每 2 分钟 3 次的泊松分布,故障持续时间服从均值为 30 秒的指数分布. 链路带宽容量设置为 1G,链路时延为 10ms,路由根据节点之间的跳数计算最优路径.

4.1　数据平面多故障恢复成功率

MFRA 本质上是一种在数据平面采用路径备份解决多故障问题的方法,为验证其算法性能,通过实验与同样采用数据平面路径备份方式的 OAM-Based[7]、OpenState-Based[9] 和 Fast Failover[11] 针对多故障恢复成功率进行对比测试. 实验中,用于通信的源、目的节点以随机数生成的方式产生,针对网络中产生的 $k(k=1,2,\cdots,5)$ 链路故障,图 3 给出了四种方案的平均恢复成功率.

如图 3 所示,相比其它三种方案,MFRA 都具有较高的数据平面多故障恢复成功率. 经计算,三个网络中,MFRA 的平均故障恢复成功率(恢复成功数之比值)分别比 OAM-Based、OpenState-Based 和 Fast Failover 提高了 6.7%、315% 和 8.5%. 其原因在于:OAM-Based

方案仅对部分核心链路进行路径备份,当链路故障数较大时无法对故障进行有效恢复;OpenState-Based 方案仅能处理单链路故障,因此可处理的链路故障数不能大于 1;Fast Failover 方案在故障发生后将数据包从第一个有效活动端口发送出去,是一种无目的性的且容易导致环路的做法,因此故障恢复性能会随着故障数的增加而降低;MFRA 为网络中的每条链路建立了环备份路径,且这些环备份路径被有效组织,能有效地解决网络中的多故障恢复问题.

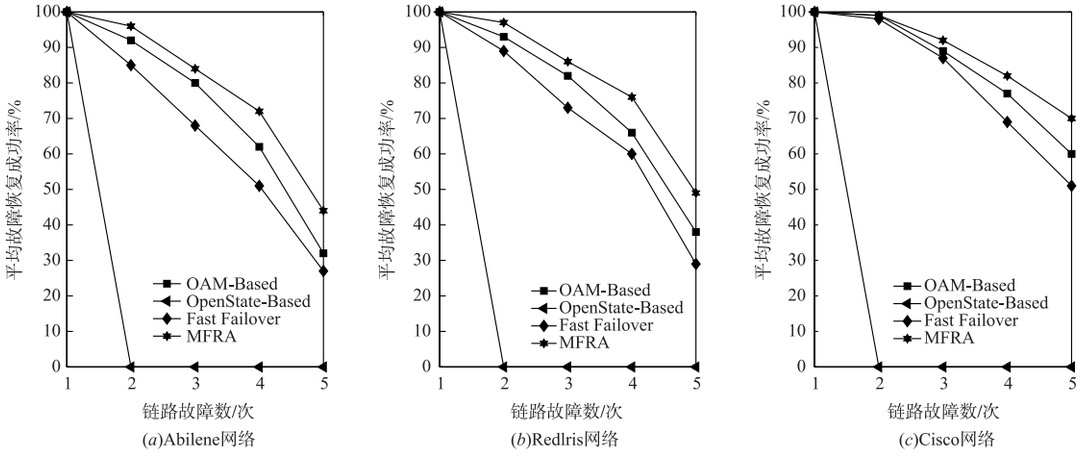


图3 数据平面多故障恢复成功率对比

4.2 控制器通信开销

为验证 MFRA 在进行故障恢复时控制平面与数据平面的通信频繁程度,采用方案 OpenFlow、OAM-Based 和 OpenState-Based 作对比实验,实验拓扑选用 Abilene 网络,其它环境与上一小节相同. 分别通过不同方案对客户端间的 ping 操作实施多故障保护. 每次测量时间为 2min,前 1min 内,源、目的节点间的通信在多故障环境下进行;后 1min 内,源、目的节点间的通信在故障恢复后的环境下进行,总的测试时间为 20min.

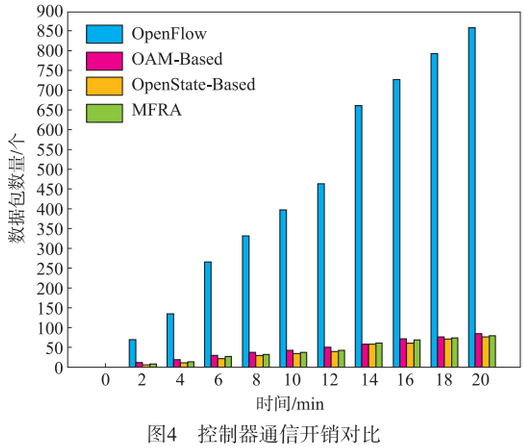


图4 控制器通信开销对比

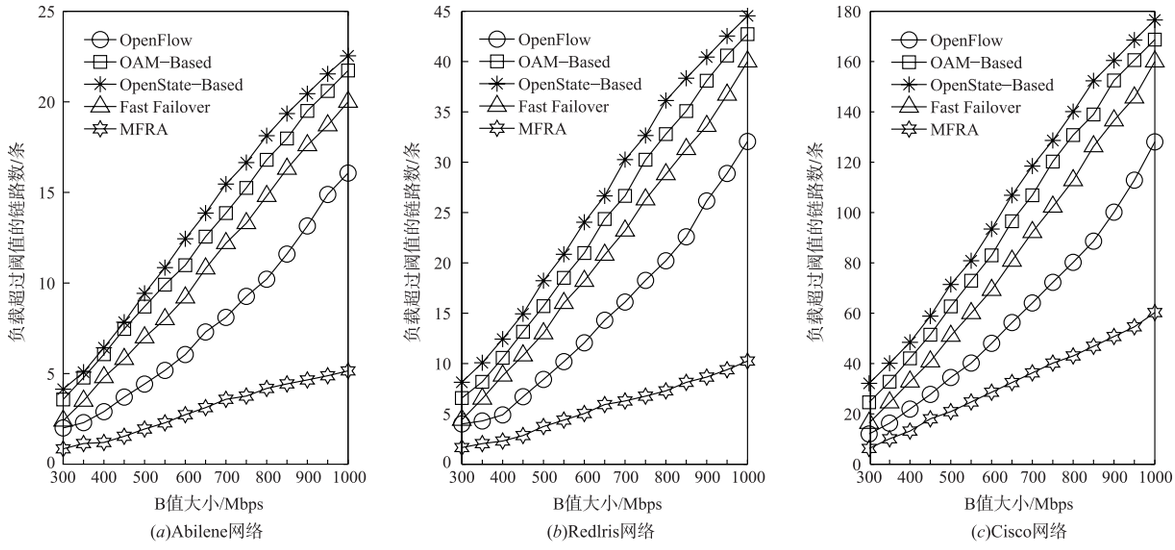
图 4 为统计的控制与数据平面通信总数随时间变化情况,这些数量不包含正常交互包,仅是故障恢复时额外增加的数据包数量. 由图中数据可知,OAM-Based、OpenState-Based 和 MFRA 的通信开销较为接近;OpenFlow 的通信开销最大,20min 后其总通信次数已经达到 860 次,而 MFRA 的只有 78 次,仅是前者的 9%. 其主要原因在于:(1)前 1min 内,每产生一次链路故障,MFRA 的数据平面需向控制平面发送 3 次 of-port-status 消息,而 OpenFlow 方案不仅有 3 次 of-port-status 消息,还有多次用于删除失效流表项的消息;(2)后 1min 内,每恢复一次链路故障,MFRA 的数据平面需向控制平面发送 3 次 of-port-status 消息,而 OpenFlow 方案中不仅有 3 次 of-port-status 消息,还有用于构建流表项的 of-packet-in 和 of-packet-out 消息.

4.3 故障规避能力

为验证 MFRA 的故障规避性能,采用 OpenFlow、OAM-Based、OpenState-Based 和 Fast Failover 作对比实验,统计各方法在故障恢复过程中网络内负载率超过设定阈值的总链路数. 实验中,源、目的节点之间传输的数据带宽为 B(Mbps),持续时间为 1min. 为模拟真实网络场景,令网络中部分节点之间产生 10(Kbps)的背景流量,且产生背景流量的节点数服从均值为每 30 秒 1 次的泊松分布. 每次测试总时长为 2h,得出整个测试过程中新增的负载率超过阈值的链路总数.

图 5 为不同网络拓扑下各方案的故障规避性能测试结果,链路负载阈值设为 80%. 经计算,Abilene 网络中,2h 测试时间内 MFRA 的负载率超过阈值的链路总数分别为前四者的 30%、22.7%、21.7% 和 25%;Redlris 网络中,其数量分别为前四者的 34.4%、25.6%、24.4% 和 27.5%;Cisco 网络中,其数量分别为前四者的 46.9%、35.5%、33.9% 和 37.5%. 由此可见,目前已有的链路故障保护机制在恢复网络故障时并没有考虑

到对故障恢复后的通信负载进行均衡,容易造成网络中部分链路负载过重而引发再故障. MFRA 方案通过在线最优化流量平滑算法对负载率超过阈值的链路实施流量平滑处理,对网络故障具有较好的规避能力,是一种既注重恢复又兼顾规避的网络故障解决机制.



(a)Abilene网络　(b)Redlris网络　(c)Cisco网络

图5　故障规避能力对比

5　结论

针对当前 SDN 网络中的多故障恢复问题,本文提出了基于 SR 的多故障恢复与规避机制,并以开放网络操作系统 ONOS 为核心,设计实现了基于 SR 的多故障恢复与规避原型系统. 实验结果表明,与现有的 SDN 网络多故障恢复机制 OAM-Based 和 Fast Failover 相比,本文所提 MFRA 的平均故障恢复成功率分别提高了 6.7% 和 8.5%,且控制器通信开销仅为传统 OpenFlow 方案的 9%,同时,MFRA 对故障恢复时导致的链路负载过重问题进行了处理,从故障规避的角度为网络提供保护.

参考文献

[1] Das S, Parulkar G, McKeown N, et al. Packet and circuit network convergence with OpenFlow[A]. Proceedings of 2010 Conference on Optical Fiber Communication[C]. San Diego, CA, 2010. 1 – 3.

[2] Giorgetti A, Cugini F, Paolucci F, et al. Open flow and PCE architectures in wavelength switched optical networks[A]. Proceedings of Optical Network Design and Modeling[C]. IEEE, 2012. 1 – 6.

[3] M Maier, M Herzog, M Scheutzow, et al. PROTECTORATION: a fast and efficient multiple-failure recovery technique for resilient packet ring using dark fiber[J]. Journal of Lightwave Technology, 2005, 23(10): 2816 – 2838.

[4] J d Decotignie, The many faces of industrial ethernet[J]. IEEE Industrial Electronics Magazine, 2009, 3(1): 8 – 19.

[5] Sharma S, Staessens D, Colle D, et al. Enabling fast failure recovery in OpenFlow networks [A]. 2011 8th International Workshop on the Design of Reliable Communication Networks (DRCN)[C]. IEEE, 2011: 164 – 171.

[6] Yu Y, Shanzhi C, Xin L, et al. A framework of using OpenFlow to handle transient link failu-re[A]. Proceedings of Transportation, Mechanical, and Electrical Engineering (TMEE)[C]. IEEE, 2011. 2050 – 2053.

[7] S Sharma, D Staessens, D Colle, et al. Fast failure recovery for in-band OpenFlow network-s[A]. Proceedings of 2013 9th International Conference on the Design of Reliable Communic-ation Networks(D-RCN)[C]. Budapest, 2013. 52 – 59.

[8] J Kempf, E Bellagamba, A Kern, et al. Scalable fault management for OpenFlow[A]. Proce-edings of IEEE International Conference on Communications (ICC)[C]. Ottawa, 2012. 6606 – 6610.

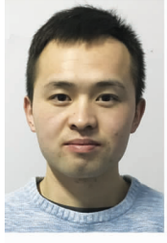
[9] A Capone, C Cascone, A Q T Nguyen, et al. Detour planning for fast and reliable failure recovery in SDN with OpenState[A]. Proceedings of 2015 11th International Conference on the Design of Reliable Communication Networks (DRCN)[C]. Kansas City, MO, 2015. 25 – 32.

[10] Desai M, Nandagopal T. Coping with link failures in centralized control plane architect-ures[A]. Proceedings of Communication Systems and Networks (COMSNETS)[C]. IEEE, 2010. 1 – 10.

[11] B Stephens, A L Cox. Deadlock-free local fast failover for

arbitrary data center netwo-rks[A]. Proceedings of IEEE INFOCOM[C]. San Francisco,CA,2016. 1 -9.
[12] 张民贵,刘斌. IP 网络的快速故障恢复[J]. 电子学报,2008,36(08):1595 -1602.
ZHANG Min-gui, LIU Bin. Fast failure recovery of IP networks[J]. Acta Electronica Sinica,2008,36(08):1595 -1602. (in Chinese)
[13] Filfils C,Nainar N K,Pignataro C,et al. The segment routing architecture[A]. Proceedings of IEEE Global Communications Conference[C]. Austin, Texas, USA: IEEE, 2014. 1 -6.
[14] Filfils C,Nainar N K,Pignataro C,et al. Segment Routing with MPLS data plane[DB/OL]. https://tools. ietf. org/html/draft-ietf-spring-segment-routing-mpls-08,2013.
[15] 邦迪. 图论及其应用[M]. 北京:科学出版社,1984.
[16] Bhatia R,Fang H,Kodialam M,et al. Optimized network traffic engineering using segment routing[A]. Proceedings of IEEE INFOCOM[C]. Hong Kong: IEEE, 2015. 657 -665.

作者简介



黄建洋 男,1991 年出生,陕西合阳人. 2014 年毕业于北京师范大学计算机科学与技术专业,其后进入国家数据交换系统工程技术研究中心攻读硕士学位,主要研究方向为新型网络体系结构、网络空间拟态防御.
E-mail:m15136143225@163. com

兰巨龙 男,1962 年出生,河北张北人. 国家数据交换系统工程技术研究中心总工程师、教授、博士生导师,主要从事新一代信息网络关键理论与技术的研究工作,目前作为首席科学家主持国家 973 项目“可重构信息通信基础网络体系研究”.