基于路由状态因果链的域间路由不稳定溯源检测方法

陈迪[1,2]，邱菡[1]，张万里[1]，朱会虎[1]，朱俊虎[1]，王清贤[1]
（1. 信息工程大学网络空间安全学院，河南 郑州 450002；2. 电子信息系统复杂电磁环境效应国家重点实验室，河南 洛阳 471003）

摘 要：针对现有域间路由不稳定溯源检测方法中检测时间受限于路由更新时延、溯源信息可能被篡改的问题，提出一种基于路由状态因果链的域间路由不稳定溯源检测方法。通过分析路由状态间存在的因果关系，定义能够刻画路由状态及其转移过程的路由状态变更标识，将其随路由更新传播发布并存储于区块链，从而构建去中心化、防篡改的路由状态因果链；通过分析本地路由状态因果链判断路由不稳定类型，追溯失效链路或策略冲突自治域序列，完成路由不稳定的溯源检测。理论证明了所提方法能够追溯导致收敛时延的失效链路和导致路由振荡的策略冲突自治域序列，并基于软件路由器在经典拓扑中进行验证。实验结果表明，所提方法可在不改变 BGP 的前提下及时检测策略与拓扑动态变化导致的路由不稳定现象并确定其源头。

关键词：域间路由安全；路由振荡；收敛时延；区块链
中图分类号：TN393
文献标识码：A
DOI: 10.11959/j.issn.1000−436x.2021221

Interdomain routing instability traceable detection method based on route state causal chain

CHEN Di[1,2], QIU Han[1], ZHANG Wanli[1], ZHU Huihu[1], ZHU Junhu[1], WANG Qingxian[1]
1. Institute of Cyberspace Security, Information Engineering University, Zhengzhou 450002, China
2. State Key Laboratory of Complex Electromagnetic Environment Effect on Electronic and Information System, Luoyang 471003, China

Abstract: To solve the problem of detection time limitation caused by route update delay and the possible tampering of traceability information in existing route instability traceable detection methods, an interdomain routing instability traceable detection method based on route state causal chain was proposed. By analyzing the causal relationship of route states, the route state update token that can describe the route state change and transfer process was defined. Route state update tokens were published and stored in the blockchain during the route update propagation to construct the decentralized and tamper-resistant route state causal chain. By analyzing the route state causal chain, the type of route instability was judged, and the failed links or policy-conflict AS sequences were located to achieve route instability traceable detection. The capability of proposed method to trace the failure link and the policy-conflict AS sequence which could lead to convergence delay and persistent route oscillation respectively was proven theoretically, and validating experiment based on software routers in typical topology was carried out. The experimental results demonstrate that the proposed method can timely detect route instability caused by the dynamic changes of both policy and topology, and determine type and root cause of route instability without modifying BGP.

Keywords: interdomain routing security, route oscillation, convergence delay, blockchain

1 引言

域间路由系统由众多独立运营的自治域（AS, autonomous system）组成。作为现行域间路由系统的事实标准协议，边界网关协议（BGP, border gateway protocol）[1]是策略驱动的路径矢量协议，支

收稿日期：2021−07−23；修回日期：2021−11−17
通信作者：邱菡，qiuhan410@aliyun.com
基金项目：国家自然科学基金资助项目（No.61502528, No.61902447）
Foundation Item: The National Natural Science Foundation of China (No.61502528, No.61902447)

持各自治域自主制定本地选路策略，进而从候选路径中选择到达特定目的网络的路径。然而，自治域配置路由策略的灵活性是以牺牲路由稳定性为代价的[2]。路由不稳定即在路径收敛前自治域不断采用和舍弃路径的状态变更过程，例如由拓扑变化导致的短时路由抖动[3]或由多个自治域策略冲突导致的持续路由振荡[4]。自治域间不必要的路由状态变更会降低网络服务质量，增加网络时延，进而导致服务中断或数据包丢失[5]。

解决域间路由稳定收敛问题的难点在于 BGP 在路由策略配置上分布自治的特性。Griffin 等[6]提出了稳定路径问题（SPP, stable paths problem）作为 BGP 稳定问题的形式化模型，给出了反映自治域间选路策略循环依赖关系的争议轮，证明了域间路由系统收敛的充分条件即不存在争议轮，但同时指出即使在已知所有策略信息的前提下检查系统是否存在稳定路径是 NP 完全问题。Gao 等[7]根据 AS 间的商业层级关系提出了域间路由策略配置的无谷底原则，并表明自治域通过牺牲一定的策略自治性，在均遵守无谷底原则的情况下可保证路由稳定性。然而实际中由于复杂的自治域商业关系，并非所有自治域都会遵守建议的路由策略原则[8]。路由抖动抑制（RFD, route flap damping）机制[9]早期曾作为缓解措施，但研究表明 RFD 对网络可达性存在负面影响[10]。因此，如何提高 BGP 路由的稳定性仍是实际域间路由系统中尚未解决的关键问题。

针对 BGP 路由不稳定问题的研究工作主要有路由不稳定预测与路由不稳定溯源 2 种思路[11]。路由不稳定预测通常采用动态调整或统一约束各节点的出入站选路策略的方式以减小 BGP 的收敛时延，并不关注导致路由不稳定的源头[12-13]，不仅无法判断与追溯 BGP 不稳定的类型及其源头，且牺牲了一定的策略自治性。路由不稳定溯源主要通过增加 BGP 传递属性追溯导致路由不稳定的源头，从而限制冗余路径探索和消除不一致状态。Zhang 等[14]针对持续路由振荡提出 update chain，通过构造路由更新链标识以检测路由振荡，但其判断发生路由振荡的充分条件只是导致路由振荡的特殊情况，有失一般性。Li 等[15]针对非单一类型的路由不稳定溯源问题提出 stable BGP，通过增加路由变更原因 root cause 传递属性过滤相关路由，提高 BGP 的收敛速度，但持续增加 root cause 信息会为 BGP 通信带来额外开销，且溯源检测时间受限于 BGP 更新报文传播时延。此外，用于路由不稳定溯源的传递属性中携带的信息在 BGP 更新报文传递过程中可被任意篡改，无法保证溯源信息的安全性与一致性。

针对上述问题，本文提出一种基于路由状态因果链的域间路由不稳定溯源检测方法 RSCTchain。RSCTchain 的主要思想是将自治域状态变更信息与转移过程以交易的模式发布并存储于区块链，构建能够反映自治域间路由状态变更因果关系的路由状态因果链，从而通过追溯 RSCTchain 检测是否存在导致路由不稳定的失效链路或策略冲突 AS 序列。本文的研究工作主要包括：1)基于拓扑与策略变化等触发路由状态变更的不同类型及因果关系，设计 RSCTchain 路由状态变更标识生成算法，生成能够反映路由状态变更的唯一标识并在区块链上进行同步；2)通过分析短时路由抖动及持续路由振荡的机理，设计 RSCTchain 路由不稳定溯源检测算法，判断路由不稳定的类型，并定位失效链路或策略冲突 AS 序列；3)从理论上证明了 RSCTchain 用于 BGP 不稳定溯源检测的正确性，并基于 Quagga 软件路由器及区块链数据库 BigchainDB 对 RSCTchain 的有效性及可扩展性进行验证。

2　问题描述

2.1　符号与假设

本文将 AS 组成的域间路由系统表示为一个时间相关的无向图 $G=(V,E)$，其中时间由非负的离散序列 $t=[0,\infty)$ 表示，每个节点 $v\in V$ 对应一个 AS，边 $(u,v)^t\in E$ 代表 t 时刻 AS v 与 AS u 之间的 BGP 链路可达，$(u,v)^t\notin E$ 代表 t 时刻 AS u 与 AS v 之间的 BGP 链路失效。选路过程中，每个节点根据本地路径偏好排序从到达特定目的节点的路径候选集中选择最优路径。下面给出路径偏好排序与路径的形式化描述。

定义 1　路径偏好排序 $\succ$。对于任意 AS 节点 $u\in V$，其路径偏好排序指 u 对本地路径集合的全序关系，表示为 $\succ_u$，若节点 u 在路径 P 和路径 Q 中更偏向于选择路径 P，则 $P\succ_u Q$。

定义 2　路径 P。在任意时刻 t，AS 节点 $u_0\in V$ 到达目的节点 d 的路径 P 用路径分配函数 π 表示为 $P=\pi(u,t)$，路径 P 途经的节点可使用序列形式表示为 $P=\langle u_0u_1\cdots u_nd\rangle$，$u_0$ 的邻居节点 u 通过路径 P 到达目的节点 d 可用节点与路径连接表示为 $\langle u,P\rangle$，

空路径表示为$\langle\varnothing\rangle$，如果路径 P 是被禁止的，则 $\langle\varnothing\rangle \succ P$。

为了表述清晰，本文做出如下假设：1)所有 AS 节点发出的路由宣告 update 报文均通往单个目的前缀地址；2)每个 AS 节点由单个 BGP 路由器代表。

2.2 域间路由稳定收敛条件

当域间路由在 t 时刻稳定收敛时，其中每一个 $v \in V$ 都被分配了通往目的节点 d 的路径 $\pi(v,t)$，所有节点到达 d 的路径构成路径分配集合 $\mathbb{P}_t$，且 $\mathbb{P}_t$ 需是一致、稳定、安全的[16]。下面，给出路径分配集合一致性、稳定性与安全性的形式化描述。

定义 3 一致性。对于在 t 时刻到达目的节点 d 的路径分配集合 $\mathbb{P}_t$，当所有路径 $P \in \mathbb{P}_t$ 构成一个以 d 为根的转发树时，则路径分配 $\mathbb{P}_t$ 是一致的，即对于任意 $v,u \in V$，如果 $\pi(v) = vuP$，则 $\pi(u) = uP$。

定义 4 稳定性。对于在 t 时刻到达目的节点 d 的路径分配集合 $\mathbb{P}_t$，其中任意 AS 节点 $v \in V$ 的路径 $\pi(v,t)$ 是其根据本地路由策略的优选路径，则路径分配 $\mathbb{P}_t$ 是稳定的，即对于任意 $v,u,w \in V$，如果 $\pi(v) = v\pi(u)$，不存在其他节点 w 使 $v\pi(w) \succ_v \pi(v)$。

定义 5 安全性。对于在 t 时刻所有节点 $v_i \in V(i = 1,2,\cdots,n)$ 到达目的节点 d 的路径分配集合 $\mathbb{P}_t = \{P_1, P_2, \cdots, P_n\}$ 发生更新时，存在有限时间 T，当 $t' < T$ 时 $\mathbb{P}_{t'} = \{P_1', P_2', \cdots, P_n'\}$ 且路径可达，当 $t' \geqslant T$ 时 $\mathbb{P}_{t'} = \mathbb{P}_T$，则路径分配 $\mathbb{P}_t$ 是安全的。

域间路由系统中各自治域均根据本地自主策略进行选路，在所有时刻保证路径分配的一致性、稳定性和安全性十分困难。Griffin 等[6]证明了判断域间路由系统稳定收敛的充分条件为在系统中不存在争议轮。一个大小为 k 的争议轮 $W = (V,Q,R)$ 如图 1 所示，由节点集合 $V = \{v_0, v_0, \cdots, v_{k-1}\}$、辐边路径集合 $Q = \{Q_0, Q_1, \cdots, Q_{k-1}\}$ 及弧边路径集合 $R = \{R_0, R_1, \cdots, R_{k-1}\}$ 组成，其中对于每个 $0 \leqslant i \leqslant k$，$R_i \in R$ 为连接 v_i 与 v_{i-1} 的边；$Q_i \in Q$ 为从 v_i 到 d 的边；且对于任意节点 v_i，$R_i v_i Q_{i+1} \succ_{v_i} Q_i$。

2.3 域间路由不稳定类型

域间路由不稳定主要包括 2 种类型：持续路由振荡及短时路由抖动。图 2 分别给出了持续路由振荡与短时路由抖动的示例。

持续路由振荡指一些 AS 节点在选路过程中始终无法稳定，使路由更新持续在这些 AS 节点中交换传递的现象。导致持续路由振荡的原因通常是各 AS 自主配置的本地优先属性配置不合理，路由策略产生了冲突。图 2(a)为策略冲突导致持续路由振荡的例子，AS_0 为其他 AS 节点的目的节点，每个节点旁边标注的是本地路径偏好列表，最上面的路径为根据本地偏好的最优路径。例如 AS_1 相较于路径 10，更希望选择路径 1430 到达 AS_0。当 AS_2 选择 210 时，AS_3 由于最优路由 320 不可达变更路径为 30，继而 AS_4 更改路径为 430，AS_1 更改路径为 1430，导致 AS_2 由于 120 路径不可达重新更改路径为 20，使 AS_3 重新选择 320……AS_0～AS_4 构成了一个争议轮，路由状态会因策略冲突周而复始地切换。

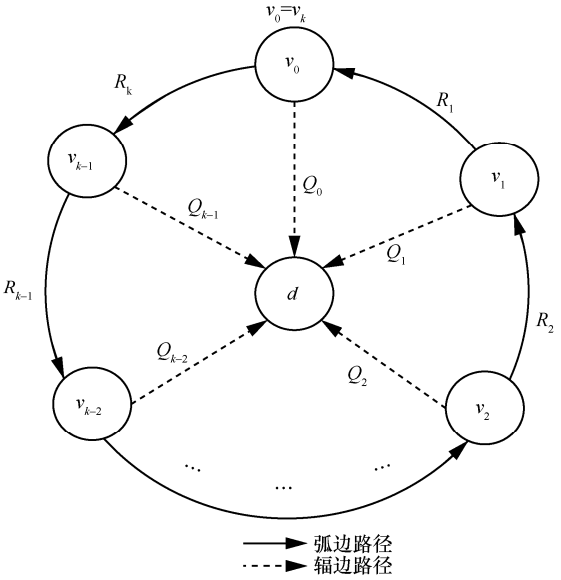
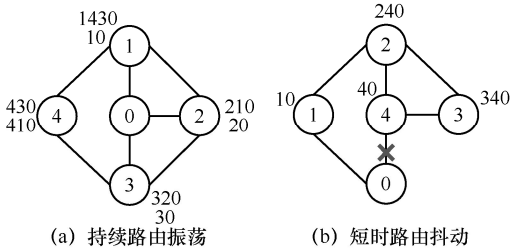


图 1　一个大小为 k 的争议轮



(a) 持续路由振荡　　(b) 短时路由抖动

图 2　持续路由振荡与短时路由抖动示例

短时路由抖动指 AS 节点错误选取并通告含有失效链路的不可达路径，导致路径冗余探索及收敛时延的现象。图 2(b)为链路失效导致短时路由抖动的例子，当链路 AS_4 与 AS_0 之间的链路失效时，AS_4 会向 AS_2 与 AS_3 发送撤回消息。然而由于 AS_2 与 AS_3 分别有备选路径 340 与 230，导致沿备选路径互相发送数据包。由于最小路由宣告间隔（MRAI, minimum route advertisement interval）计时器的限

制，即使 AS_2 和 AS_3 已经开始互相转发流量，但还无法发出新路径的更新报文（MRAI 计时器建议取值为 30 s[11]）。最终当 MRAI 计时器到期，AS_2 与 AS_3 才会发现通过 AS_1 到达 AS_0 的备选路径。

因此，需要一种路由不稳定溯源检测方法，能够在图 2(a)中的路由状态开始反复变更时检测出策略冲突的 AS 序列 2→3→4→1→2，使相关 AS 可调整本地策略消除冲突；且能够在图 2(b)中的链路失效后使相关 AS 获知 AS_0 与 AS_4 间的链路已失效，不再向包含链路 40 的路径发送数据包或通告该路径，避免路径的冗余探索。

综上，本文要解决的问题是针对当前路由不稳定溯源检测方法中检测时间受限于路由更新时延、溯源信息可能被篡改的问题，在不改变现行 BGP 的前提下，检测 BGP 不稳定现象并判断其类型（拓扑变化导致的短时路由抖动/策略冲突导致的持续路由振荡），追溯导致 BGP 不稳定的具体失效链路/策略冲突 AS 序列，同时保证检测过程可追溯、防篡改。

3　RSCTchain

3.1　RSCTchain 概述

针对上述问题，本文提出一种基于路由状态因果链的域间路由不稳定溯源检测方法 RSCTchain。为了能够以可追溯、防篡改的方式对非单一类型的 BGP 路由不稳定现象进行溯源检测，采用区块链对反映路由状态变更因果关系的信息进行分布式存储；通过检索本地区块链数据库，从而与 BGP 并行工作完成 BGP 不稳定溯源检测；基于区块链共识机制保证数据存储的安全性与一致性。

图 3 展示了 RSCTchain 的整体架构与工作流程。在 RSCTchain 中，每个参与自治域需要运行一个代表本自治域的 RSCTchain 区块链节点，在不引起混淆的情况下，下文简称其为 RSCT 节点。每个 RSCT 节点都持有一个公钥−私钥对，其公钥的哈希值作为该节点的唯一身份标识。所有参与 RSCTchain 的 AS 节点共同构成逻辑上的信任覆盖网络，基于 RSCTchain 采用的共识算法发布并一致地存储路由状态变更标识，协同检测路由不稳定并追溯其源头。RSCTchain 的工作流主要由记录与检索两类组成。记录工作流指当 AS 更新本地路由状态时，其 RSCT 节点生成相应的路由状态变更标识 RSCT，将 RSCT 以交易的形式经签名后发布于区块链网络，在完成共识确认后存储于每个 RSCT 节点。RSCT 记录对应状态更新的类型、变更路径、触发节点等信息，在逻辑上按照 AS 间导致路由状态变更的因果关系依次相连，构成路由状态因果链。检索工作流指 RSCT 节点在对应 AS 做出路由决策前，通过追溯路由状态因果链检测是否存在导致路由不稳定的失效链路或策略冲突 AS 序列，并将检测结果返回对应 AS 以为其路由决策提供参考。

RSCTchain 中分布式存储的路由状态因果链的安全性与一致性取决于 RSCTchain 采用的共识算法。RSCTchain 中的自治域参与节点不同于公链参与节点的完全竞争状态，各 RSCT 节点间是在有限

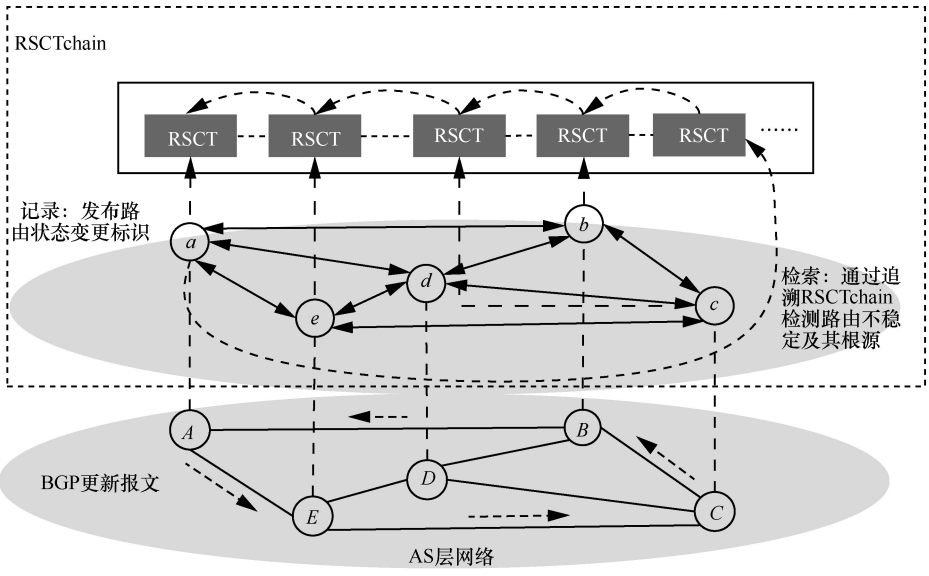


图 3　RSCTchain 的整体架构与工作流程

竞争的基础上共同寻求整体利益的最大化（如路由的稳定一致）。因此联盟链共识机制更适用于RSCTchain 的实际需求。以 Tendermint[17]共识算法为例，由于 Tendermint 是拜占庭容错的，当 RSCT 节点中恶意节点比例不超过 1/3 时，能够避免路由状态因果链被篡改，诚实节点仍达成一致的路由状态因果链最终状态，从而正常完成 BGP 不稳定溯源检测。因此 RSCTchain 可以使参与自治域在去中心化信任的基础上，实现可追溯、防篡改的路由不稳定检测，提供可靠溯源证据。

为了优化路由不稳定检测与溯源时间，RSCTchain 在实现中可将每个 RSCT 节点运行在与 BGP 路由器物理上独立的专用服务器中，并对应一个运行在 BGP 路由器中的本地验证节点。路由状态标识发布时，RSCT 节点负责在区块链网络中完成同步与存储。本地验证节点不参与区块链中的共识同步过程，只负责定期接收 RSCT 节点下发的区块并在缓存中存储本地相关的路由状态变更标识以供查询，即对于任意的 AS，可将通往同一目的网络且具有连锁因果关系的路由状态变更标识按序存储于本地路由验证缓存中。

3.2 路由状态变更标识

在 RSCTchain 中，使用路由状态变更标识记录各自治域路由状态变更信息与转移过程。本节给出路由状态变更标识的定义及其生成算法。

文献[18-19]给出了由拓扑与策略变化触发的可能影响收敛过程的四类路由状态变更事件（Tup, Tdown, Tshort, Tlong），统称为 T 事件，并在此基础上给出路由状态变更类别的定义。

定义 6 路由状态变更类别。对于任意的自治域节点在 $t+1$ 时刻将 t 时刻的路径 $r_o=\pi(u,t)$ 更改为 $r_n=\pi(u,t+1)$ 的路由状态变更事件 e，其对应的路由状态变更类别指导致该事件的原因类别，具体可表示为 RSCT.type(e)，RSCT.type(e)$\in$ {0+,0−,1+,1−}，其中，“0+”代表由 AS u 与某邻接 AS 间的链路在 $t+1$ 时刻恢复导致，且 AS u 在 $t+1$ 时刻变更的新路径 r_n 的本地优先级高于 t 时刻的当前路径 r_o，即 $r_n\succ r_o$ 的路由状态变更事件；“0−”代表由 AS u 与某邻接 AS 间的链路在 $t+1$ 时刻失效导致，且 AS u 在 $t+1$ 时刻变更的新路径 r_n 的本地优先级低于 t 时刻的当前路径 r_o，即 $r_n\prec r_o$ 的路由状态变更事件（当 $r_n=\langle\varnothing\rangle$ 时即对 r_o 进行撤回操作）；“1+”代表由 AS u 本地路由策略变更或收到某邻接 AS 的路径更新通告导致，且 AS u 在 $t+1$ 时刻变更的新路径 r_n 的本地优先级高于 t 时刻的当前路径 r_o，即 $r_n\succ r_o$ 的路由状态变更事件；“1−”代表由 AS u 收到某邻接 AS 的路径更新通告导致，且 AS u 在 $t+1$ 时刻变更的新路径 r_n 的本地优先级低于 t 时刻的当前路径 r_o，即 $r_n\prec r_o$ 的路由状态变更事件。

为叙述方便，下文使用新路径 r_n 和旧路径 r_o 分别代表 $t+1$ 时刻的更新路径和 t 时刻的当前路径。下面，给出路由状态变更标识 RSCT 的定义。

定义 7 路由状态变更标识。对于任意一个路由状态变更事件，路由状态变更标识 RSCT 指唯一对应该事件的一个十元组，RSCT = (Key, PT, SC, UN, CN, OP, NP, TS, Sig, Pk)，其中，“Key”代表该路由状态变更标识的索引，这里 Key = hash(UN,NP,TS)；“PT(previous token)”代表该路由状态变更标识的上一个标识的索引，即导致当前 AS 路由状态变更的路由状态变更标识，在当前 AS 路由状态变更是由于本地原因（如链路失效、策略更改）的情况下，PT = null；“SC(state change type)”代表路由状态变更的类别，SC$\in${0+,0−,1+,1−}；“UN(update node)”代表当前路由状态变更的 AS 对应节点，即发布该路由状态变更标识的节点；“CN(cause node)”代表导致当前 AS 路由状态变更的触发 AS，在当前 AS 仅根据本地因素变更路由状态时，CN = null；“OP(old path)”代表变更节点的当前待更改的路径；“NP(new path)”代表变更节点要采用的新路径，当没有可达路径时，NP = null；“TS(time stamp)”代表当前路由状态变更的时间戳；“Sig(signature)”代表当前路由状态变更节点使用本地私钥对发布 RSCT 内容摘要进行的数字签名 $\mathrm{DSK_{UN}}$(hash(Key, PT, SC, UN, CN, OP, NP, TS))，用于验证 RSCT 中的信息未被篡改过；“Pk(public key)”代表当前路由状态变更节点的公钥，用于检查数字签名的真实性。

路由状态变更标识的生成算法如算法 1 所示。

算法 1 路由状态变更标识生成 RSCTgeneration

输入 在 t 时刻 AS u 的路径 r_o 被撤回或被替换为 r_n（因链路失效/恢复、本地策略更改或收到邻接 AS v 在 t' 时刻通告路径 r_v 的路由更新 U_v）

输出 路由状态变更标识 RSCT

1) if link(u, r_o.nexthop) 链路失效 and $r_o\succ r_n$

2) RSCT=(hash(u,r_n,t),null,0−,u,null,r_0,r_n,t, Sig_u, Pk_u)

3) else if link(u,r_n.nexthop) 链路恢复 and $r_o \prec r_n$

4) RSCT=(hash(u,r_n,t),null,0+,u,null,r_0,r_n,t, Sig_u, Pk_u)

5) else if u 更改本地优先级策略 and $r_o \prec r_n$

6) RSCT=(hash(u,r_n,t),null,1+,u,null,r_0,r_n,t, Sig_u, Pk_u)

7) else if u 在接收到U_v后将要更改本地路由状态

8) if $r_o \prec r_n$ and r_n.nexthop = v

9) RSCT=(hash(u,r_n,t),hash(v,r_v,t'),1+,u,v,r_o, r_n,t,Sig_u, Pk_u)

10) if $r_o \succ r_n$ and r_n.nexthop! = v

11) RSCT=(hash(u,r_n,t),hash(v,r_v,t'),1−,u,v,r_o, r_n,t, Sig_u,Pk_u)

12) return RSCT

生成路由状态变更标识分为 4 种情况。

1) 当 AS u 与路径r_o的下一跳 AS 之间的链路失效，AS u 将r_o撤回（即$r_n = \langle\varnothing\rangle$）或将其替换为一条次优路径$r_n$时，生成(hash($u$, r_n, t), null, 0−, u, null, r_o, r_n, t, Sig_u, Pk_u)作为该路由状态变更的标识 RSCT（第 1)～2)行）。

2) 当 AS u 与路径r_n的下一跳 AS 之间的链路在失效后恢复，AS u 将当前次优路径r_o替换为本地偏好值更高的r_n路径时，生成(hash(u, r_n, t),null, 0+,u,null,r_o,r_n,t,Sig_u,Pk_u)作为该路由状态变更的标识 RSCT（第 3)～4)行）。

3) 当 AS u 的本地优先级策略发生更改，选取更改后优先级更高的r_n替换当前路径r_o时，生成(hash(u, r_n, t), null, 1+,u,null,r_o,r_n,t,Sig_u,Pk_u)作为该路由状态变更的标识 RSCT（第 5)～6)行）。

4) 当 AS u 由于接收到邻接 AS v 在t'时刻通告路径r_v的路由更新U_v要将当前路径r_o替换为r_n时（第 7)行）时，需再分为 2 种情况考虑，第一种情况是 AS u 将 AS v 作为下一跳并通过r_v的路径本地优先级高于当前路径r_o(即$r_n = \langle uv, r_v\rangle$)，此时 AS u 将 (hash(u,r_n,t),hash(v,r_v,t'),1+,u,v,r_o,r_n,t,Sig_u,Pk_u) 作为其路由状态变更的标识 RSCT（第 8)～9)行）；第二种情况是 AS u 不选择 AS v 作为其下一跳 AS，且因为收到U_v不得不将当前路径r_o替换为优先级更低的新路径r_n时，将(hash(u,r_n,t), hash(v, r_v, t'),1−,u,v,r_o, r_n, t, Sig_u, Pk_u)作为其路由状态变更标识 RSCT（第 10)～11)行）；最后返回生成的路由状态变更标识 RSCT（第 12)行）。

图 4 给出了 RSCTchain 中一个路由状态变更标识生成的例子，在图 4(a)的初始稳定状态 t_0 时刻，节点 a,b,c 到达 d 的路径形成一个稳定的转发树 $\mathbb{P}_0 = \{ad, bd, cad\}$；在 t_1 时刻，节点 a 变更了本地策略，如图 4(b)，将路径 abd 的优先级调整为高于路径 ad 的优先级，进而将到达 d 的路径由 ad 变更为 abd 并向邻居节点转发路由更新报文，同时生成对应该状态变更的 RSCT，即图 4 右侧的 RSCT（Key①）；在 t_2 时刻，节点 c 由于收到节点 a 的路由更新，不得不将当前路径 cad 更改为 cd 并向邻居节点转发路由更新报文，同时生成 RSCT（Key②）；在 t_3 时刻，节点 b 由于收到节点 a 的路由更新，在获取路径 cd 可达后将本地路径更新为优先级更高的 bcd 并向邻居节点转发路由更新报文，同时生成对应该状态变更的 RSCT（Key③）；在 t_4 时刻，节点 a 又因为收到 b 的路由更新，获知路径 bd 不可达后重新将路径变更为 ad，生成 RSCT（Key④）。在没有检测机制的干预下，节点 a,b,c 的路由状态会循环往复地变更，RSCTchain 也会不断增长。在每个 RSCT 中都包含了对应状态变更的必要信息，且能够通过 RSCT 中的 PT 值获取引发当前路由状态变更的变更事件及节点信息，从而支持对整个路由状

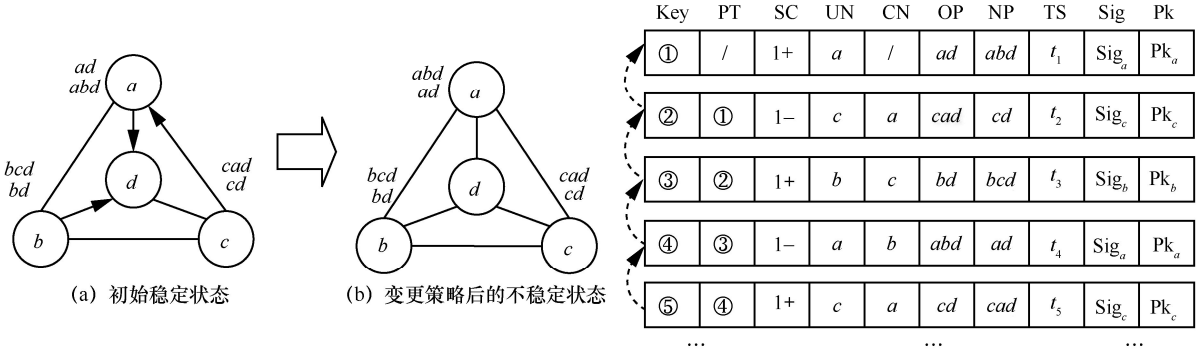


图 4　RSCTchain 路由状态变更标识生成示例

态变更过程的追溯。

3.3 路由不稳定溯源检测

RSCTchain 中的路由不稳定溯源检测算法不仅需要判断路由不稳定类型，还需要定位失效链路或策略冲突 AS 序列。本节给出 RSCTchain 的路由不稳定溯源检测算法。

路由不稳定溯源检测算法如算法 2 所示。首先对参数进行初始化（第 1)～6)行）。然后在指定检测时间范围内，从当前 RSCT 开始向前追溯（第 7)～27)行），检查 RSCT 中数字签名的正确性，若数字签名有误，报告对应路由状态变更标识的索引值并返回（第 8)～9)行）。当检测到“0−”类型路由状态变更事件时，则返回失效链路（第 10)～11)行），当前 AS 可对路由表中包含失效链路的路径进行过滤；当追溯的 RSCT 中的 UN 节点与当前 AS 节点相同时，即发生了路由状态因果环路。在此情况下，第一步根据 curRSCT 的前一个标识 nxtRSCT 的事件类型 SC 来决定 Path_{in} 的取值，当 nxtRSCT.SC 为“1+”或“0+”时，Path_{in} 为 curRSCT 中的新路径，当 nxtRSCT.SC 为“1−”时则为 curRSCT 中的旧路径（第 13)～16)行）；第二步根据 AS u 根据 U_v 要在本地变更的新路径 r_n 与要替换的旧路径 r_o 来决定 Path_{out} 取值，当 $r_o \prec r_n$ 且 r_n 的下一跳为 v 时，Path_{out} 为 AS u 选取的新路径 r_n，当 $r_o \succ r_n$ 且 r_n 的下一跳不是 v 时，Path_{out} 为 AS u 当前的旧路径 r_o（第 17)～20)行）；第三步通过比较 Path_{in} 与 Path_{out} 在 AS u 的优先级，判断是否发生了策略冲突，若存在策略冲突，将检测起始时间 T 作为当前发生冲突的时间 t 并返回策略冲突 AS 序列 causeChain，报告策略冲突争议轮（第 21)～23)行）。若不存在策略冲突，则将 nxtRSCT 赋为 curRSCT 的索引，curRSCT 赋为其上一个索引标识 curRSCT.PT，并将当前路由状态更新自治域节点追加至 causeChain，继续向前回溯（第 24)～26)行）。最后若在指定检测时间范围内未发现异常情况，返回 True（第 27)行）。

算法 2 路由不稳定溯源检测 RITdetection

输入 AS u 收到 AS v 在 t 时刻通告路径 r_v 的路由更新 U_v；AS u 根据 U_v 要在本地变更的新路径 r_n 与要替换的旧路径 r_o；检测起始时间 T，路由状态变更链 RSCTchain

输出 检测正常/异常（异常时返回故障源：失效故障链路或策略冲突 AS 序列）

1) $\text{Path}_{\text{in}} \leftarrow$ null

2) $\text{Path}_{\text{out}} \leftarrow$ null

3) RSCTkey $\leftarrow$ hash(v, r_v, t)

4) curRSCT $\leftarrow$ RSCTchain.get(RSCTkey)

5) nxtRSCT $\leftarrow$ null

6) causeChain $\leftarrow$ null

7)while curRSCT.TS > T and curRSCT.PT ≠ null

8)if curRSCT.Sig 签名验证不正确

9) return RSCTkey //返回签名验证错误的 RSCT 索引 RSCTkey

10) if curRSCT.SC == 0−

11) return link(curRSCT.UN,curRSCT.OP.nexthop) //返回失效链路

12) else if curRSCT.UN == u and nxtRSCT ≠ null

13) if nxtRSCT.SC == 1+or nxtRSCT. SC == 0+

14) $\text{Path}_{\text{in}} \leftarrow$ curRSCT.NP

15) else

16) $\text{Path}_{\text{in}} \leftarrow$ curRSCT.OP

17) if $r_o \prec r_n$ and r_n.nexthop == v

18) $\text{Path}_{\text{out}} \leftarrow r_n$

19) if $r_o \succ r_n$ and r_n.nexthop ≠ v

20) $\text{Path}_{\text{out}} \leftarrow r_o$

21) if $\text{Path}_{\text{in}} \prec \text{Path}_{\text{out}}$

22) $T \leftarrow t$ //更新检测起始时间

23) return (u, Path_{in}, Path_{out}, disputeList) //返回争议轮中的策略冲突 AS 序列

24) nxtRSCT $\leftarrow$ curRSCT

25) curRSCT $\leftarrow$ RSCTchain.get(curRSCT.PT)

26) causeChain.append(curRSCT.UN)

27)return True

以图 4 为例，在 t_4 时刻，当 a 收到来自 b 宣告路径 bcd 时，需把当前路径 abd 更新为 ad，此时为了检测是否存在失效链路或策略冲突，a 在 RSCTchain 中向前追溯，直到发现在 t_1 时刻的 RSCT（Key①）中的 UN 为 a，构成因果环路。首先根据 RSCT（Key②）中的 SC 类型，确定 $\text{Path}_{\text{in}} = ad$；然后根据路径 abd 与 ad 的优先级顺序，确定 $\text{Path}_{\text{out}} = abd$；最后通过比较 Path_{in} 与 Path_{out} 的优先级，发现存在策略冲突，并报告构成策略冲突的 AS 序列 $\{a,b,c,a\}$ 与冲突路径 $\{abd, ad\}$。

基于 RSCTchain 进行路由不稳定溯源检测的时间复杂度主要体现在该更新报文在每一次被自治域节点接收和转发时在路由状态标识链中根据导致路由状态变更的因果关系向前回溯的过程。假设

链上存储的到达不同目的前缀的数量为 c，待验证的更新报文 U 中的 AS-path 长度为 n，RSCTchain 路由不稳定溯源检测在最坏情况下的时间复杂度为 $O(cn^2)$。根据 2021 年 1 月发布的 BGP 测量报告，从 2012 年到 2021 年，BGP 更新报文中的平均路径长度一直稳定在 5.7，在 2021 年还呈下降趋势。截至 2021 年 1 月，路由表中宣告的目的 IP 网络前缀数量为 860 000，且数量近年来均保持较稳定，没有出现大规模增长。

4　正确性证明

本节给出 RSCTchain 的正确性证明。基于第 2 节中描述的域间路由模型及路由收敛的充分条件，针对 RSCTchain 在链路失效导致的短时路由抖动及策略冲突导致的持续路由振荡 2 种情况下的溯源检测方法，相应地给出定理 1 和定理 2，并进行证明。

定理 1 给定由去往同一目的节点 d 的自治域节点组成的无向图 $G=(V,E)$，若所有自治域节点 $v\in V$ 均为 RSCTchain 的参与节点，则在链路失效时，对于任意自治域节点 $v\in V$，v 能够检测出可导致短时路由抖动的失效链路。

证明　无效路由的冗余传播是导致短时路由抖动的必要条件[20]。假设在无向图 $G=(V,E)$ 中存在失效链路 P，定理 1 的证明等同于证明 RSCTchain 的参与节点使用 RITdetection 算法可避免继续选择或传播包含失效链路 P 的无效路由。

不失一般性地，假设有无向图 G。链路失效导致的短时路由抖动如图 5 所示，节点 v_0 通往目的节点 d 的优选路由为 $P_0=\langle v_0 d\rangle$；节点 v_k 通往目的节点 d 的优选路由为 $P_k=\langle v_k v_0 d\rangle$；节点 $v_i(0<i<k)$ 通往目的节点 d 的优选路由 $P_i=\langle v_i v_{i-1} P_{i-1}\rangle$ 均包含链路 P_0。若不使用 RSCTchain 进行溯源检测，当链路 P_0 失效后，v_k 在收到 v_0 对 P_0 的撤回通告后会选用路由表中的备选路径 $P_k'=\langle v_k P_1\rangle$；在收到 v_1 对 P_1 的撤回通告后会选用备选路径 $P_k''=\langle v_k P_2\rangle$，不断重复此过程，直到收到 v_{k-1} 对 P_{k-1} 的撤回通告后才发现去往目的节点 d 不可达。由于链路 P_0 的失效导致冗余更新报文的传播，造成短时路由抖动。

当 $G=(V,E)$ 中所有自治域节点 $v\in V$ 均为 RSCTchain 的参与节点时，根据 RSCTgeneration 算法，v_0 在对 P_0 进行撤回通告的同时会生成类型为“0−”的路由状态变更标识，同步至 RSCTchain 区块链网络，并由其他各 RSCT 节点存至本地 RSCTchain 存储模块。根据路由不稳定溯源检测 RITdetection 算法，对于任意节点 $v_i\in V$，不管是否收到邻接节点发送的路由更新报文，均可检测导致短时路由抖动的失效链路 P_0，并在本地路由表中过滤包含 P_0 的路由，避免无效路由的冗余传播。证毕。

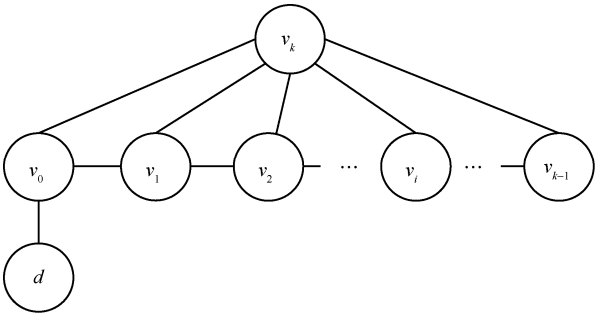


图 5　链路失效导致的短时路由抖动

定理 2　给定由去往同一目的节点 d 的自治域节点组成的无向图 $G=(V,E)$，若所有自治域节点 $v\in V$ 均为 RSCTchain 的参与节点，则对于任意自治域节点 $v_0\in V$，v_0 能够检测出是否在本地节点发生路由持续振荡并追溯相应的策略冲突自治域序列。

证明　假设在 $G=(V,E)$ 中存在引起路由持续振荡的策略冲突相邻自治域序列 $S=\{v_0,v_1,v_2,\cdots,v_k\}$，则 S 中对于任意整数 $i\in[0,k]$，节点 v_{i-1} 的路由状态变更引发节点 v_i 的路由状态变更且 $v_0=v_k$。对定理 2 的证明即等同于证明 v_0 使用 RITdetection 算法可检测出 G 中存在争议轮。

不失一般性地，假设无向图 G 为如第 2 节中图 1 所示的争议轮。当相邻自治域序列 $S=\{v_0,v_1,v_2,\cdots,v_k\}$ 中对于任意整数 $i\in[0,k]$，节点 v_{i-1} 的路由状态变更引发节点 v_i 的路由状态变更时，对于任意整数 $i\in[0,k]$，节点 v_{i-1} 的路由状态变更引发节点 v_i 的路由状态变更可分为 2 种情况：第一种情况为 v_i 的路由状态变更类型为“0+”或“1+”，即由于 v_{i-1} 的路由状态变更使 v_i 选择 v_{i-1} 作为新路径的下一跳且新路径的优先级高于旧路径，这代表 v_i 更愿意选择通过弧边 $\langle v_i,v_{i-1}\rangle$ 的路径；第二种情况为 v_i 的路由状态变更类型为“1−”，即由于 v_{i-1} 的路由状态变更使 v_i 不得不放弃当前偏好值更高的路径，该被替换的旧路径下一跳为 v_{i-1}，即优先级高于旧路径，这也代表 v_i 更愿意选择通过弧边 $\langle v_i,v_{i-1}\rangle$ 的路径。因此 S 中自治域节点依次相连可构成争议轮中的弧边。

当 $v_0=v_k$ 时，S 中的自治域节点依次相连构成一个环，这里将 v_0 首次状态变更的时刻记为 t_{in}，v_0 由

v_{k-1} 状态变更导致的状态变更时刻记为 t_{out}，下面分别通过分析 t_{in} 和 t_{out} 时刻的路由状态变化，确定 v_0 节点的辐边与弧边：在 t_{in} 时刻，若 v_0 导致 v_1 的状态变更类型为“0+”或“1+”时，代表 v_1 选取了一条经过 v_0 的新路径 $R_1v_0Q_0$，这说明 v_0 在 t_{in} 时刻的新路径 $Q_0=\pi(v_0,t_{\text{in}})$ 是可达的。因为 v_0 的辐边是唯一的，且在 v_1 节点处 $R_1v_0Q_0$ 的优先级更高，所以 v_0 的辐边为 v_0 在 t_{in} 时刻的新路径 Q_0；若 v_0 导致 v_1 的状态变更类型为“1−”时，代表 v_1 被迫放弃了经过 v_0 的旧路径 $R_1v_0Q_0$，这说明 v_0 在 t_{in} 时刻状态变更前的旧路径 $Q_0=\pi(v_0,t_{\text{in}}-1)$ 在 t_{in} 时刻前是可达的，则 v_0 的辐边为 v_0 在 t_{in} 时刻状态变更时被替换的旧路径 Q_0；同样地，在 t_{out} 时刻，若 v_{k-1} 导致 v_0 状态变更时的新路径优先级高于旧路径，且 v_0 拟选取的新路径下一跳为 v_{k-1}，即代表 v_0 在状态变更后包含弧边 R_k 的新路径 $R_kv_{k-1}Q_{k-1}$ 可达，因此 v_0 包含弧边的路径为在 t_{out} 时刻的新路径 $R_kv_{k-1}Q_{k-1}$；若 v_{k-1} 导致 v_0 状态变更时的新路径优先级低于旧路径，且 v_0 拟选取的新路径下一跳不是 v_{k-1} 时，代表 v_0 状态变更前包含弧边 R_k 的旧路径 $R_kv_{k-1}Q_{k-1}$ 可达，因此 v_0 包含弧边的路径为在 t_{out} 时刻的旧路径 $R_kv_{k-1}Q_{k-1}$。在确定本地辐边与弧边后，v_0 可在本地比较路径 $R_kv_{k-1}Q_{k-1}$ 与路径 Q_0 的优先级，当 $R_kv_{k-1}Q_{k-1}\succ Q_0$ 时，v_0 即可判断 G 中存在争议轮。

当 $G=(V,E)$ 中所有自治域节点均为 RSCTchain 的参与节点时，根据 RSCTgeneration 算法，自治域节点对应的 RSCT 节点会生成描述本地路由状态变更的标识，同步至 RSCTchain 区块链网络，并由各 RSCT 节点存至本地 RSCTchain 存储模块。根据上述分析，当在 v_0 节点开始发生路由持续振荡后，v_0 可使用 RITdetection 算法基于 RSCTchain 本地存储模块追溯路由状态变更标识，从而检测出在本地节点发生的路由持续振荡并追溯出相应的策略冲突自治域序列 $S=\{v_0,v_1,v_2,\cdots,v_k\}$。证毕。

综上，RSCTchain 可以对短时路由抖动（收敛时延）及持续路由振荡 2 种类型的 BGP 不稳定情况进行检测，且可以追溯导致路由不稳定的故障源（失效链路或策略冲突 AS 序列）。

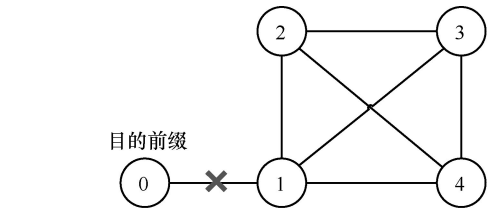
5　仿真分析

为了对 RSCTchain 的有效性和可扩展性进行验证，本文基于 Quagga 软件路由器和 BigchainDB 实现了 RSCTchain 的功能，在经典拓扑中开展仿真实验验证其有效性，并基于 BGP 现网数据对 RSCTchain 的可扩展性进行分析。

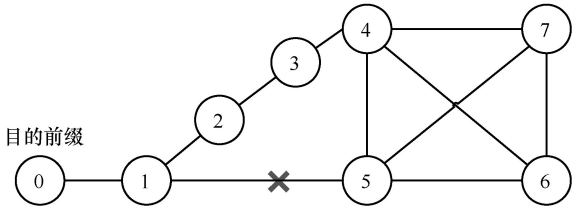
5.1　仿真实验设计

本文实现了 RSCTchain 原型系统。区块链存储基于 BigchainDB 实现。BigchainDB 广泛用于开发并部署区块链概念验证平台和应用程序，可视为一个具备区块链特征的开源分布式存储系统，同时具备区块链块链属性和数据库属性，其数据存储基于 MangoDB 实现，其联网与共识基于拜占庭容错的 Tendermint 实现；域间路由网络拓扑基于软件路由器 Quagga 的 BGPd 模块（即 BGP 守护进程）实现；每个网络节点运行在 Docker 虚拟容器（20.10.5 版本）上，运行在一台 64 GB 内存，Intel® Xeon® Gold 6230 CPU@2.10 GHz 的服务器中的 Ubuntu 16.04 虚拟机（内存 8 GB，硬盘 40 GB，处理器 8）上。

本文选取 Quagga BGPd 模块（等同于使用 BGP）及 stableBGP[15]作为比较对象，分别在策略冲突和链路失效的情况下开展 RSCTchain 的有效性实验。在策略冲突实验情景中，采用文献[6]中给出的策略冲突导致持续路由振荡的 Bad Garget 经典拓扑（如图 2(a)所示），选用更新报文数量用于评估有效性[14-15]。在链路失效实验情景中，采用经常用于路由失效稳定性研究的 Clique 拓扑及 Backup-Clique 拓扑[18,21-22]，如图 6 所示。图 6(a)为 Clique 拓扑，图 6(b)为 8 节点的 Backup-Clique 拓扑。Backup-Clique 拓扑可模拟一个边缘网络通过一条直接路径和一个较长的备选路径连接到核心网络的情景。在链路失效情况下，选用路由收敛时间和更新报文数量评估有效性。



（a）Clique拓扑



（b）8节点的Backup-Clique拓扑

图 6　链路失效实验网络拓扑

5.2　有效性分析

RSCTchain 能够针对非单一类型的路由不稳定进行检测溯源，区分路由不稳定类型并追溯其源头。因此分别在策略冲突和链路失效的情况下开展有效性实验。

1) 策略冲突实验

图 7 展示了在策略冲突实验情景中，使用 BGP、RSCTchain 及 stableBGP 时，网络中的更新报文总量随 MRAI 时间间隔的变化，这里 MRAI 取值为 30 s。从图 7 中可观察到，在使用 BGP 的情况下，整个网络中的累计更新报文数量随着 MRAI 时间间隔的增多而不断增长，这说明发生了由策略冲突引起的路由振荡，导致路由不收敛；在使用 stableBGP 的情况下，在第三个 MRAI 时间间隔前后检测出策略冲突，在调整策略后的 3 个 MRAI 时间间隔内路由收敛，之后网络中的更新报文数量不再继续增长，其总量保持为 51 个，说明 stableBGP 能够检测出可能导致路由振荡的路由策略冲突，并在及时干预后能够使路由收敛；在使用 RSCTchain 的情况下，也是在第三个 MRAI 时间间隔前后检测出策略冲突，但调整策略后的 2 个 MRAI 时间间隔内路由即收敛，后续网络中的更新报文数量不再继续增长，其总量保持为 45 个。

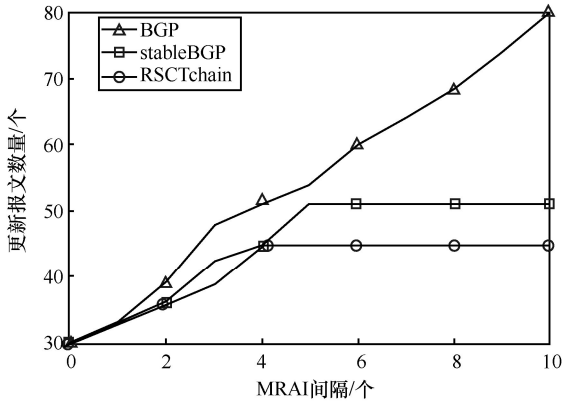


图 7　路由策略冲突情景中 BGP、stableBGP 和 RSCTchain 有效性对比

2) 链路失效实验

图 8 展示了链路失效实验情景中，使用 BGP、RSCTchain 及 stableBGP 时，在不同规模的网络中的收敛时间和更新报文数量。从图 8(a)可以看出，使用 BGP 的网络收敛时间最长；使用 stableBGP 的网络收敛时间次之，比使用 BGP 平均减少了 39.75%；使用 RSCTchain 的网络收敛时间最短，比使用 BGP 平均减少了 50%。相较于使用 BGP，RSCTchain 能够比 stableBGP 平均多缩短 10.25%的网络收敛时间。从图 8(b)可观察到，使用 BGP 情况下路由收敛时网络中总共交换的更新报文数量最多；使用 stabelBGP 的更新报文数量次之，比使用 BGP 平均减少了 16.75%；使用 RSCTchain 的更新报文数量最少，比使用 BGP 平均减少了 27.06%。相较于使用 BGP，RSCTchain 能够比 stableBGP 平均多减少 10.31%的冗余更新报文传播。

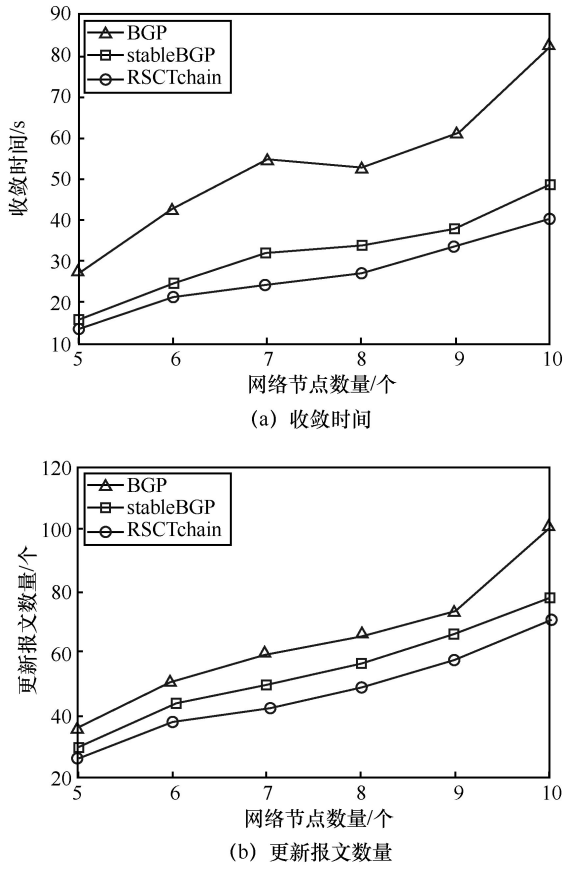


图 8　链路失效情景中 BGP、stableBGP 和 RSCTchain 的有效性对比

由上述实验结果可知，相较于 stableBGP，RSCTchain 能够更及时地检测策略与拓扑动态变化导致的路由不稳定现象并确定其源头，且能抑制更多的冗余更新报文传播，其原因在于：RSCTchain 使用了 BGP 带外路由状态变更同步机制（区块链节点数据同步），而 stableBGP 使用 BGP 带内路由状态变更同步机制（携带于 BGP 更新报文中逐跳传递，其效率取决于 MRAI 设置）。本实验中 MRAI 设置为默认值 30 s，BigchainDB 中采用 Tendermint 的区块时间实测为 3 s 左右，因此 RSCTchain 能够更及时地对路由不稳定现象做出反应。

5.3 可扩展性分析

为了分析 RSCTchain 在实际域间路由部署的可扩展性，基于 Routeviews 项目提供的 BGP 现网数据在 BigchainDB 中模拟路由变更标识的发布与存储过程，统计时间开销和空间开销随着处理更新报文数量的变化，进而根据当前域间路由系统实际需求分析 RSCTchain 的可扩展性。

RSCTchain 原型系统中每个区块最多能包含 400 个路由状态变更标识交易，平均区块时间为 3 s，确认并同步一个路由状态变更标识的时间为 7.5 ms。在处理不同数量 BGP 更新报文的情况下分别生成对应的 RSCTchain 区块链数据库，其所需空间大小及相应路由不稳定溯源的平均时间如图 9 所示。从图 9 可以看出，随着处理 BGP 更新报文数量的增多，RSCTchain 所需存储大小及路由不稳定溯源检测平均时间均呈线性增长趋势。

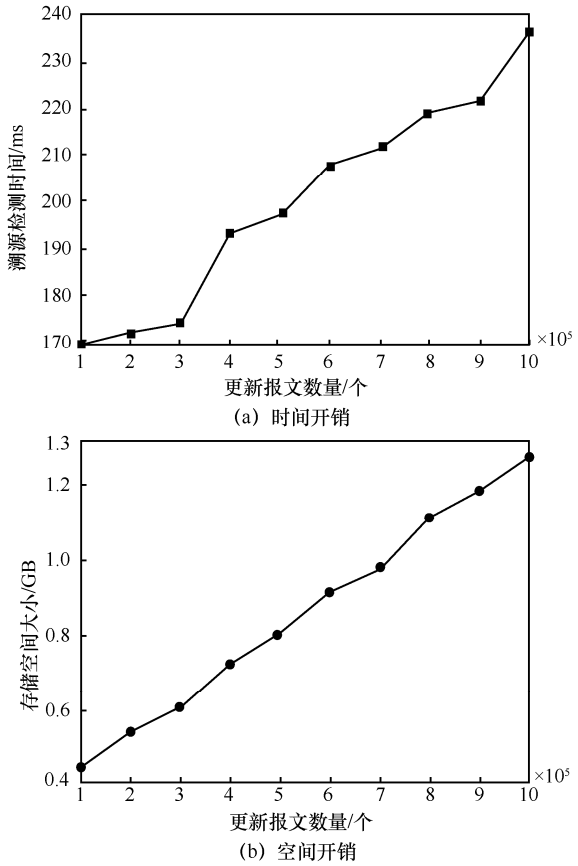


图 9 RSCTchain 可扩展性分析

根据上述实验结果，结合域间路由实际需求对 RSCTchain 的可扩展性进行分析。当前域间路由系统中平均每日的更新报文约为 400 000 个。在时间开销方面，RSCTchain 不稳定溯源检测时间在毫秒级别，对处理 400 000 个更新报文产生的 RSCT 溯源检测检测时间为 193.23 ms，当以天为单位进行溯源检测时可以满足域间路由系统控制平面实际需求；在空间开销方面，RSCTchain 的路由状态变更链大小将以每日约 0.7 GB 的速度增长，一年的路由状态变更链约需 255.5 GB 的存储空间，可以适应当前通用商用硬盘。因此 RSCTchain 应用于实际域间路由系统是可行的。此外，现有的区块链历史交易数据压缩、扩容技术等研究[23]也可进一步提高 RSCTchain 的可行性。

6 结束语

本文提出一种基于路由状态因果链的域间路由不稳定溯源检测方法 RSCTchain，通过将路由状态变更信息与转移过程发布到区块链，构建可反映路由状态变更因果关系的路由状态因果链，使自治域通过追溯本地区块链存储以检测和定位失效链路或策略冲突自治域序列。理论证明与仿真实验结果表明，RSCTchain 能够以合理开销及时检测非单一类型的路由不稳定现象，并定位导致路由不稳定的失效链路或策略冲突自治域序列，可有效抑制冗余路由更新报文、减少路由收敛时间。下一步工作将基于实际域间路由拓扑开展更大规模的仿真实验与性能测试。

参考文献：

[1] REKHTER Y, LI T, HARES S. A border gateway protocol 4 (BGP-4)[R]. RFC Editor, 2006.

[2] LABOVITZ C, AHUJA A, BOSE A, et al. Delayed Internet routing convergence[J]. IEEE/ACM Transactions on Networking, 2001, 9(3): 293-306.

[3] PEI D, ZHANG B C, MASSEY D, et al. An analysis of convergence delay in path vector routing protocols[J]. Computer Networks, 2006, 50(3): 398-421.

[4] VARADHAN K, GOVINDAN R, ESTRIN D. Persistent route oscillations in inter-domain routing[J]. Computer Networks, 2000, 32(1): 1-16.

[5] KUSHMAN N, KANDULA S, KATABI D. Can You hear me now? ![J]. ACM SIGCOMM Computer Communication Review, 2007, 37(2): 75-84.

[6] GRIFFIN T G, SHEPHERD F B, WILFONG G. The stable paths problem and interdomain routing[J]. IEEE/ACM Transactions on Networking, 2002, 10(2): 232-243.

[7] GAO L X, REXFORD J. Stable Internet routing without global coordination[J]. IEEE/ACM Transactions on Networking, 2001, 9(6): 681-692.

[8] GILL P, SCHAPIRA M, GOLDBERG S. A survey of interdomain routing policies[J]. ACM SIGCOMM Computer Communication Re-

view, 2013, 44(1): 28-34.
[9] VILLAMIZAR C, CHANDRA R, GOVINDAN R. BGP route flap damping[R]. RFC Editor, 1998.
[10] MAO Z M, GOVINDAN R, VARGHESE G, et al. Route flap damping exacerbates Internet routing convergence[J]. ACM SIGCOMM Computer Communication Review, 2002, 32(4): 221-233.
[11] DA S R B, SOUZA M E. A survey on approaches to reduce BGP interdomain routing convergence delay on the Internet[J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2949-2984.
[12] GODFREY P B, CAESAR M, HAKEN I, et al. Stabilizing route selection in BGP[J]. IEEE/ACM Transactions on Networking, 2015, 23(1): 282-299.
[13] SOBRINHO J L, FIALHO D, MATEUS P. Stabilizing BGP through distributed elimination of recurrent routing loops[C]//Proceedings of 2017 IEEE 25th International Conference on Network Protocols (ICNP). Piscataway: IEEE Press, 2017: 1-10.
[14] ZHANG J, HU Z Y, ZHANG T. Update chain-based approach for checking route oscillation of BGP[J]. Chinese Journal of Aeronautics, 2011, 24(2): 202-209.
[15] LI Q, XU M W, WU J P, et al. Toward a practical approach for BGP stability with root cause check[J]. Journal of Parallel and Distributed Computing, 2011, 71(8): 1098-1110.
[16] FEAMSTER N, JOHARI R, BALAKRISHNAN H. Implications of autonomy for the expressiveness of policy routing[J]. ACM SIGCOMM Computer Communication Review, 2005, 35(4): 25-36.
[17] KWON J. Tendermint: consensus without mining[R]. 2014.
[18] PEI D, AZUMA M, MASSEY D, et al. BGP-RCN: improving BGP convergence through root cause notification[J]. Computer Networks, 2005, 48(2): 175-194.
[19] AFEK Y, BREMLER-BARR A, SCHWARZ S. Improved BGP convergence via ghost Flushing[J]. IEEE Journal on Selected Areas in Communications, 2004, 22(10): 1933-1948.
[20] OLIVEIRA R, ZHANG B C, PEI D, et al. Quantifying path exploration in the Internet[J]. IEEE/ACM Transactions on Networking, 2009, 17(2): 445-458.
[21] WENHUA W, QINGGUO S, QIN Z. On the relationship between BGP convergence delay and network topology[C]//Proceedings of 2008 11th IEEE International Conference on Communication Technology. Piscataway: IEEE Press, 2008: 546-549.
[22] GÄMPERLI A, KOTRONIS V, DIMITROPOULOS X. Evaluating the effect of centralization on routing convergence on a hybrid BGP-SDN emulation framework[C]//Proceedings of the 2014 ACM Conference on SIGCOMM. New York: ACM Press, 2014: 369-370.
[23] BONEH D, BÜNZ B, FISCH B. Batching techniques for accumulators with applications to IOPs and stateless blockchains[C]//Advances in Cryptology – CRYPTO 2019. Berlin: Springer, 2019: 561-586.

[作者简介]

陈迪（1992− ），女，河南郑州人，信息工程大学博士生，主要研究方向为网络系统安全、区块链技术。

邱菡（1981− ），女，湖北随州人，博士，信息工程大学副教授，主要研究方向为域间路由安全、网络安全模拟与评估。

张万里（1998− ），男，湖南常德人，信息工程大学硕士生，主要研究方向为数据安全、漏洞挖掘。

朱会虎（1992− ），男，河南郑州人，信息工程大学博士生，主要研究方向为域间路由安全。

朱俊虎（1974− ），男，河南郑州人，信息工程大学教授，主要研究方向为网络对抗、网络安全测试与评估。

王清贤（1960− ），男，河南卫辉人，博士，信息工程大学教授、博士生导师，主要研究方向为网络安全。