软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(6):2865−2877 [doi: 10.13328/j.cnki.jos.006494]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

SDN 中基于图分割的自适应带内网络遥测探测路径配置*

原鹏翼[1,2], 王 淼[2], 王凌豪[1,2], 张玉军[1,2], 周继华[3]

[1](中国科学院大学, 北京 100049)

[2](中国科学院 计算技术研究所, 北京 100190)

[3](金美通信, 重庆 400030)

通信作者: 张玉军, E-mail: zhmj@ict.ac.cn

摘 要: 软件定义网络 (SDN) 是一种将控制与转发平面分离的新型网络架构, 可以基于全局信息进行网络资源的调度和优化, 而精确的调度需要对全网信息 (包括网络中所有交换设备状态及拓扑中所有链路信息) 进行准确的测量. 带内网络遥测可以在转发数据包的同时实现相关信息的采集, 其中配置全网覆盖的探测路径是带内网络遥测需要解决的关键问题之一. 但现有 SDN 网络中全网覆盖的带内网络遥测路径配置方案存在以下问题: (1) 需要提前部署大量探测节点导致维护开销增大; (2) 探测路径过长导致探测分组长度超过网络中的 MTU 值; (3) 冗余的探测路径导致测量引入的流量负荷在网络整体流量中占比过大; (4) 动态变化拓扑下探测路径调整恢复时间长等. 为解决上述问题, 提出了 SDN 中基于图分割的自适应带内网络遥测探测路径配置 (ACGS) 方法, 其基本思想是: 利用图分割对网络拓扑图进行划分, 通过控制拓扑规模来限制探测路径长度; 在分割后的子图中求解欧拉回路得到只遍历子图中有向边一次的探测路径, 以避免探测节点数量过多、探测路径冗余度高的问题; 并利用局部调整与整体调整相结合的方式解决拓扑动态变化时探测路径恢复时间长的问题. 实验结果证明 ACGS 方法能够在 SDN 网络环境下, 实现探测路径长度适中、探测节点数量较少、探测路径冗余程度更低的全网覆盖带内网络遥测探测路径配置, 并实现其在拓扑动态变化后更快速的调整.

关键词: 软件定义网络; 带内网络遥测; 图分割; 欧拉回路; 动态拓扑

中图法分类号: TP393

中文引用格式: 原鹏翼, 王淼, 王凌豪, 张玉军, 周继华. SDN中基于图分割的自适应带内网络遥测探测路径配置. 软件学报, 2023, 34(6): 2865–2877. http://www.jos.org.cn/1000-9825/6494.htm

英文引用格式: Yuan PY, Wang M, Wang LH, Zhang YJ, Zhou JH. Adaptive Detection Path Configuration for In-band Network Telemetry in SDN Based on Graph Segmentation. Ruan Jian Xue Bao/Journal of Software, 2023, 34(6): 2865–2877 (in Chinese). http://www.jos.org.cn/1000-9825/6494.htm

Adaptive Detection Path Configuration for In-band Network Telemetry in SDN Based on Graph Segmentation

YUAN Peng-Yi[1,2], WANG Miao[2], WANG Ling-Hao[1,2], ZHANG Yu-Jun[1,2], ZHOU Ji-Hua[3]

[1](University of Chinese Academy of Sciences, Beijing 100049, China)

[2](Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

[3](Jin Mei Communication, Chongqing 400030, China)

Abstract: Software-defined network (SDN) is a new network architecture that separates the control and forwarding planes. It can schedule and optimize network resources based on global information. Nevertheless, precise scheduling requires accurate measurement of

* 基金项目: 国家重点研发计划 (2018YFB1800403); 网络计算创新研究院课题 (E061010003); 国家自然科学基金 (61902382, 61972381); 中国科学院战略性先导科技专项 (XDC02030500)
收稿时间: 2021-05-17; 修改时间: 2021-07-07, 2021-09-02; 采用时间: 2021-09-18; jos 在线出版时间: 2022-10-14
CNKI 网络首发时间: 2022-11-16

information on the entire network (including the status of all switching devices in the network and all link information in the topology). In-band network telemetry (INT) can realize the collection of relevant information while forwarding data packets, and configuration of detection paths which cover the entire network is one of the key issues to be solved for INT. However, existing detection path configuration methods for INT have the following problems. (1) The deployment of a large number of detection nodes is required in advance, which leads to increased maintenance overhead. (2) The detection path is too long, which results in the length of detection packet exceeding the MTU value in the network. (3) The redundant detection paths cause the traffic load introduced by the measurement to account for too much of the overall network traffic. (4) The recovery time of the detection path adjustment under the dynamically changing topology is too long. In order to solve the above problems, an adaptive detection path configuration method for in-band network telemetry in SDN based on graph segmentation (ACGS) is proposed. The basic idea is to divide the network topology with the graph segmentation to restrict the length of detection path by controlling the topology scale, solve the Euler circuit in the divided subgraph to obtain a detection path that only traverses the directed edges in the subgraph once, to avoid the problems of too many detection nodes and high detection path redundancy; and use the combination of local adjustment and global adjustment to solve the problem of long recovery time of the detection path when the topology changes dynamically. The experimental results prove that the ACGS method can realize the INT detection path configuration in SDN with moderate detection path length, fewer detection nodes, lower detection path redundancy, and faster adjustment under the dynamically changing topology.

Key words: software-defined network (SDN); in-band network telemetry (INT); graph segmentation; Euler circuit; dynamic network

软件定义网络 (software defined network, SDN)[1]作为一种新型网络架构, 将传统网络的控制平面与数据转发平面分离, 控制平面负责集中的逻辑控制, 转发平面负责数据转发. 这种架构拥有全局可控的控制平面和高效的转发平面, 可以基于全局信息进行网络资源的调度和优化[2]. 而精确的调度需要对全网信息 (包括网络中的所有交换设备状态以及拓扑中的所有链路信息) 进行准确的测量. 带内网络遥测[3]就是近年来在网络测量领域中备受关注的一种全网信息测量方法.

带内网络遥测是指探测路径上的节点在转发数据包的同时, 将设备状态和网络链路信息插入数据包, 以完成相关信息的采集. 如图 1 所示, 带内网络遥测的探测路径上的节点按功能分为一个源节点 (source)、若干个转发节点 (transit) 以及一个汇聚节点 (sink), 其中, 源节点负责向原始数据包头部插入遥测指令, 指定需要采集的网络状态和设备信息, 并根据遥测指令的要求将设备状态、网络链路信息以及时间戳以元数据 (INT metadata) 的形式插入到数据包头部, 然后转发数据包; 中间的转发节点收到带有遥测指令的数据包后, 根据遥测指令的要求将设备状态、网络链路信息以及时间戳以元数据的形式插入到数据包头部, 然后转发数据包; 汇聚节点收到带有遥测指令的数据包后, 同样根据遥测指令要求将设备状态、网络链路信息以及时间戳以元数据的形式插入到数据包头部, 然后将数据包中的所有遥测元数据提取出来, 发送至控制器, 并将剔除了遥测元数据后的数据包转发至目的主机. 相较于传统网络测量方法, 带内网络遥测方法能够对网络设备状态和网络链路信息进行更细粒度的测量, 其中设计覆盖全部网络链路的探测路径是关键, 包括确定各个节点的功能 (源节点、转发节点、汇聚节点) 以及规划从源节点到汇聚节点的探测路径.

目前, 研究者提出了一些带内网络遥测探测路径配置方法[4−8], 但存在着以下问题: (1) 探测节点 (包括源节点和汇聚节点) 数量过多, 由于控制器需要向源节点下发遥测指令、收集来自汇聚节点的遥测结果, 所以过多的探测节点会导致控制器维护开销增大; (2) 单条探测路径长度过长, 导致探测包大小超过网络 MTU, 从而引起探测包的丢包; (3) 探测路径冗余, 即一条链路多次被探测路径覆盖, 这会导致测量引入的流量负荷在网络整体流量中占比过大; (4) 网络拓扑动态变化时, 重新部署探测路径的时间长.

为了解决上述问题, 提出了一种基于图分割的自适应带内网络遥测路径配置方法 (adaptive detection path configuration method for in-band network telemetry in SDN based on graph segmentation, ACGS): 首先, 通过图分割的方式把网络拓扑划分为多个子图, 通过限制拓扑的规模从而解决单条探测路径长度过长的问题; 然后, 基于欧拉回路的思想在子图中规划得到覆盖拓扑中有向边仅一次的探测路径, 以解决探测节点数量和探测路径冗余问题; 并给出在动态拓扑下, 自适应使用局部调整与整体调整相结合的方式, 解决了网络拓扑动态变化时重新部署探测路径时间长的问题. 实验结果表明, ACGS 方法能够在控制探测节点数量的同时, 降低探测路径平均长度, 降低探测路径冗余程度, 并在动态拓扑下保证调整速度更快.

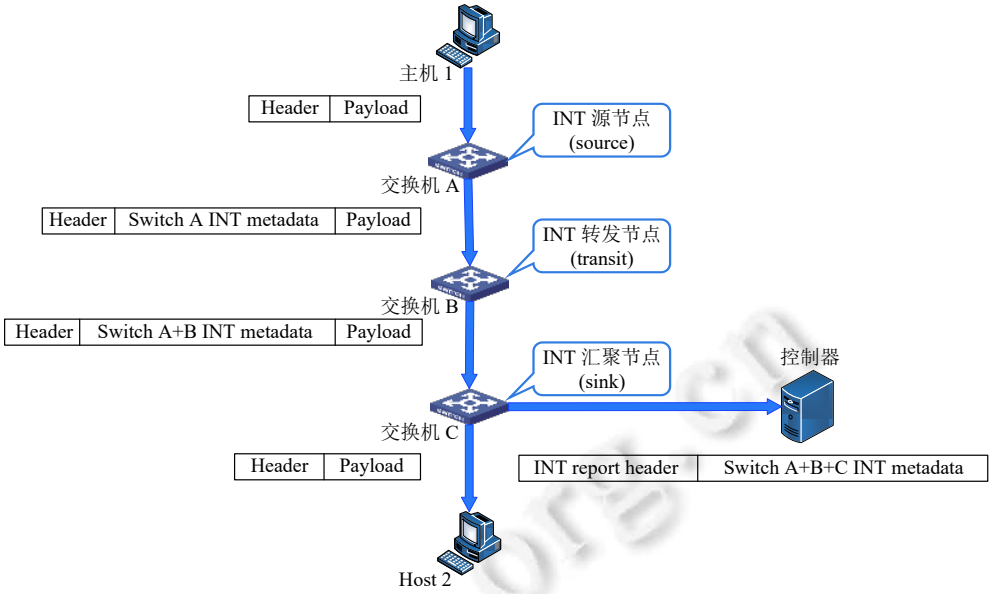


图 1　带内网络遥测测量过程示例

1　相关研究

软件定义网络中带内网络遥测能够持续监测网络状态, 动态高效地测量网络参数, 使控制器能够获得全网网络状态, 统筹管理全网流量[9], 这个过程中配置覆盖全网的探测路径是关键.

文献 [4] 提出了一种基于最短路径的生成树算法的路径覆盖机制, 首先, 配置多个探测节点; 然后, 以各个探测节点为生成树的根, 根据最短路径策略规划出多个生成树以覆盖整个网络. 该方法中同一条链路存在极大可能被多个生成树覆盖, 存在探测路径冗余的问题, 使网络测量效率下降, 另外, 额外配置探测节点会导致维护开销增大.

文献 [5,6] 提出了一种基于欧拉回路的路径覆盖策略, 先将无向图扩展为有向图; 根据欧拉定理[10,11], 一定存在这样一条路径, 即源节点和汇聚节点为同一个节点, 路径覆盖拓扑图中的所有有向边和所有节点, 且这条路径保证每条有向边只遍历一次. 这种方法解决了基于最短路径的生成树算法中测量路径冗余的问题, 但是对于较大规模的网络, 会出现探测链路过长导致探测分组长度超过网络中的 MTU 值, 从而引起丢包.

文献 [7] 中提出了一种基于 Hierholzer 算法的路径覆盖机制, 通过 Hierholzer 算法求解多条欧拉回路, 以完成全网的路径覆盖. 由于 Hierholzer 算法生成的多个欧拉路径的长度是无法限制的, 所以因探测路径过长而导致探测分组长度超过网络中的 MTU 值的问题并未解决, 并且每条探测路径长度也可能存在较大差距, 可能引起网络资源分配不均匀的问题.

文献 [8,12] 中提出了一种基于启发式贪心算法的图覆盖算法, 先将网络拓扑分为多个子集; 然后, 随机选取一条链路, 尽可能向这条路径中添加更多的链路, 合并子集以获得探测路径最少的贪心策略. 这种路径覆盖算法同样存在着探测分组长度超过网络中的 MTU 值的问题.

综上所述, 现有全网覆盖的带内网络遥测探测路径配置方法存在的问题包括: 额外配置的探测节点会带来更大的维护开销; 因为探测路径上的转发节点需要插入遥测数据, 所以长度过长的探测路径会引起探测包大小超过网络 MTU, 从而导致探测包丢包率增加; 而存在大量冗余的探测路径会导致测量引入的流量负荷在网络整体流量中占比过大.

同时, 现有方法当网络拓扑动态变化时就重新部署探测路径, 探测路径恢复速度慢的问题, 所以在动态变化的网络拓扑中, 如何减少探测路径配置的维护难度, 提高拓扑变化后部署测量路径的效率, 也是软件定义网络环境下带内网络遥测待解决的问题.

2 ACGS 方法

为解决上述问题, 提出了 ACGS 方法, 包括基于图分割的探测路径部署机制和探测路径自适应调整机制, 方法思路如下.

基于图分割的探测路径部署机制负责规划出覆盖全网的带内网络遥测探测路径. 首先, 通过图分割算法将拓扑图分割为多个拓扑子图, 以解决探测路径过长导致的数据包长度超过 MTU, 引起丢包的问题; 然后, 通过在拓扑子图中求解欧拉回路获得遍历子图中所有边的探测路径, 解决探测节点数量过多以及探测路径冗余的问题; 最后, 控制器将探测路径下发至交换设备, 以完成全网覆盖的带内网络遥测探测路径配置.

探测路径自适应调整机制是为了解决当网络拓扑发生变化时, 重新部署探测路径面临的探测路径恢复速度慢的问题, 使用探测路径局部调整与整体调整相结合的方式调整带内网络遥测的探测路径. 由于探测路径的规划过程中将网络拓扑分割为多个子图, 对拓扑变化的子图中的探测路径进行局部调整, 有效降低了探测路径的调整范围, 减少了修复探测路径的时间; 由于多次局部调整会引起探测路径冗余程度增高, 所以当变化的拓扑子图超过一定比例后, 需要整体调整探测路径, 重新调用基于图分割的探测路径部署机制以完成探测路径的规划.

2.1 基于图分割的探测路径部署机制

基于图分割的探测路径部署机制负责根据网络拓扑规划出覆盖全网的带内网络遥测探测路径, 其主要流程如图 2 所示, 首先, 利用图分割算法将输入的拓扑图分为多个符合阈值大小限制的子图; 然后, 在子图中基于欧拉回路的思想规划探测路径; 最后控制器向可编程交换机下发探测路径, 输出遥测指令.

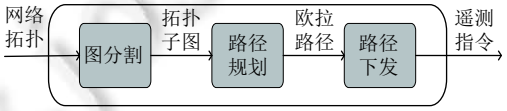


图 2 基于图分割的探测路径部署机制流程

2.1.1 图分割

图分割作为基于图分割的探测路径部署机制的预处理部分, 作用是将较大的拓扑子图分为多个符合拓扑分割阈值大小的拓扑子图, 并保证子图之间的割边尽量少.

图分割涉及到如下两个概念[13]: 1) 无向图的割指的是, 作为无向图 $G=(V, E)$, 设 C 为图 G 中一些弧的集合, 若从 G 中删去 C 中的所有弧能使图 G 不是连通图, 称 C 为图 G 中的一个割, C 中的边成为割边; 2) S-T 割是使顶点 S 与顶点 T 不再连通的割.

依据 S-T 割概念, 一个无向图求割的过程中, 两点 A 和 B 只可能存在两种情况: 1) A、B 不在同一子图中, 即 S-T 割; 2) A、B 在同一子图中, 即非 S-T 割. 那么当求一个无向图割的过程中, 对于其中两点, 只需要先求其 S-T 割, 再将两点整合为一个点, 同时将与这两点相连的边权值重新计算, 对更新后的拓扑图继续寻找新的 S-T 割, 那么就可以覆盖 A 和 B 两点的所有分割情况.

通过在 Stoer-Wagner 算法[14]的流程中, 保存每次计算出的割的中间结果, 同时限制子图阈值大小, 选择一个符合阈值大小且割的权值最小的子图作为一个的分割结果. 再对除去这个子图之后的拓扑图重新运行算法, 获得下一个子图, 直到将整个拓扑图分割成多个符合阈值的子图.

算法主要流程如下.

- Step 1. 根据 Stoer-Wagner 算法, 在图 G 中找出任意 S-T 最小割.
- Step 2. 记录分割结果, 合并 S、T, 重复执行 Step 1 直到图 G 只剩下一个顶点.
- Step 3. 选择记录结果中符合子图阈值大小且割的权值最小的第 1 个结果作为一个子图.
- Step 4. 删除分割好的子图, 更新整体拓扑的节点集合和边集合.
- Step 5. 对新的子图重新执行 Step 1, 直到所有子图符合阈值大小.

伪代码如算法 1.

算法 1. 图分割.

输入: *matrix* 拓扑矩阵, *devices* 交换机数组;
输出: *res* 割.

```
1. function NumRestrictMinCut(matrix, devices)
2. while len(devices) > threshold do
3.    weights devs 数组用于记录中间结果
4.    while len(devices) > 1 do
5.       minCut ← INT_MAX
6.       dev ← 0
7.       for device in devices do
8.          计算使 device 与图不再连通的割的权值 tempCut
9.          if tempCut < minCut then
10.            minCut ← tempCut
11.            dev ← device
12.         end if
13.      end for
14.      weights.append(minCut)
15.      devs.append(dev)
16.      找到与 dev 邻接的边里权值最大的边 edge, 及 edge 的另一个端点 dev0
17.      dev 与 dev0 组成节点集合{dev, dev0}, 以该集合作为一个新的节点更新 matrix、devices
18.   end while
19.   选择数组 weights、devs 中, weight 最小且 dev 小于 threshold 的作为一个子图 subGraph
20.   删除 matrix 和 devices 中, 子图 subGraph 包含的节点和边
21.   res.append(subGraph)
22. end while
23. return res
24. end function
```

图 3 所示为分割一次子图的示例, 每个步骤对应的 Cut 表示当前拓扑对应最小割的权值, 如图 3 第 (4) 步中, 节点 4-7-8 与其他节点的割是当前最小割, 所以 Cut 值为 3. 当分割子图大小符合子图阈值时, 即为一次成功的子图分割. 例如, 当子图阈值大小为 3 时, 图 3 第 (4) 步和第 (7) 步都是符合子图阈值且割的权值最小的条件的, 可以随机选择其中一个作为一次子图分割, 删除子图后重新执行算法 Step 1, 获得下一个符合要求的子图.

关于子图阈值大小的选择, 子图的欧拉回路路径长度 L 应该符合公式 (1) 的条件, 公式 (1) 中 $Length_{\mathrm{MTU}}$ 代表网络 MTU 大小; $Length_{\mathrm{Packet}}$ 代表探测数据包长度; $Length_{\mathrm{IH}}$ 代表带内网络遥测头部长度; $Length_{\mathrm{IM}}$ 代表带内网络遥测元数据长度.

$$L < \frac{Length_{\mathrm{MTU}} - Length_{\mathrm{Packet}} - Length_{\mathrm{IH}}}{Length_{\mathrm{IM}}} \tag{1}$$

在传统网络中, 以太网数据包大小在 64–1 500 B 之间; 带内网络遥测的头部长度为 12 B, Metadata 长度为 16 B; 网络中 MTU 通常为 1 500 B. 将上述理论值带入公式 (1), 得到子图的欧拉回路路径长度 L 的理论范围在 0–89 之间, 但是文献 [15] 中指出, 网络中超过 60% 的流量数据包大小不超过 1 KB, 所以为了满足超过 60% 的流量在插入带内网络遥测数据后, 仍然能够符合 MTU 大小, 那么子图的欧拉回路路径长度应小于 30, 同时为了

保证由无向图扩展为有向图后, 路径里边的个数仍小于 30, 拓扑子图中的无向边个数应小于 15, 所以子图阈值节点数量应该控制在 15 以下. 实际场景中, 在选择子图阈值时, 除了需要考虑 MTU 大小的约束外, 还需要考虑子图的规模过小会导致子图数量增加, 进而导致探测节点数量增加, 系统维护开销增大. 经第 3.2.1 节的实验测试, 子图阈值范围应该控制在 10–15 之间.

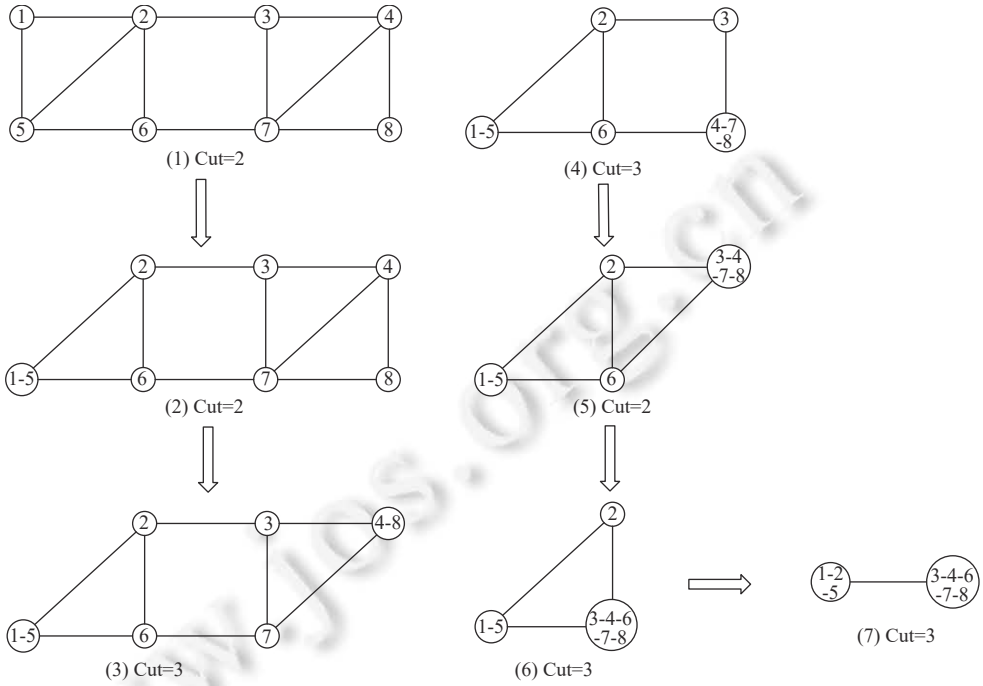


图 3 图分割示例流程

2.1.2 路径规划

路径规划作为基于图分割的探测路径部署机制的主要部分, 负责在划分好的拓扑子图中求解一条覆盖子图所有有向边仅一次的带内网络遥测探测路径.

欧拉路径是指图 G 中的一个路径包括每个边恰好一次的有向路径. 如果一个回路是欧拉路径, 则称为欧拉回路 (Euler circuit). 一个图 G 存在欧拉回路的充要条件是 G 为有向图, G 的基图连通, 并且所有顶点的出度与入度都相等; 或者除两个顶点外, 其余顶点的出度与入度都相等, 而这两个顶点中一个顶点的出度与入度之差为 1, 另一个顶点的出度与入度之差为−1. 根据上述图 G 存在欧拉回路充要条件中的第 1 条, 将无向图扩展为有向图, 所有顶点的出度与入度都相等, 则一定存在一条可以覆盖整个拓扑子图的欧拉路径.

欧拉回路的求解是利用欧拉定理判断出一个图存在欧拉回路或欧拉通路后, 随机选择一个正确的起始顶点, 用 DFS 算法遍历所有的边 (每一条边只遍历一次), 遇到走不通就回退. 在搜索前进方向上将遍历过的边按顺序记录下来. 这组边的排列就组成了一条欧拉通路或回路. 由于拓扑图是由无向图扩展得来的, 而求解欧拉路径的目的是覆盖无向图的所有边, 由一条无向边扩展来的两条边只需要覆盖一条即可, 所以上述计算过程是可以提前结束的, 通过维护两个数组 undirctedVisited 和 dirctedVisited, 分别表示无向图边的访问情况和有向图边的访问情况, 当 undirctedVisited 全覆盖时, 即无向图边全覆盖, 就可以结束算法, 回到起始点. 通过遍历选取子图各个节点作为源节点, 选择欧拉回路最短的起点作为最终的源节点.

2.1.3 路径下发

路径的下发作为基于图分割的探测路径部署机制的最后一步, 负责将规划好的探测路径下发至交换设备, 其功能主要通过 P4 Runtime 实现.

P4 Runtime 是一套基于 Protobuf 以及 gRPC 框架上的协议, 通过 P4 Runtime, SDN 控制器可以控制 P4 的设备,

并获得设备基本信息. 与传统 SDN 南向协议 OpenFlow 不同, 除了具备高度弹性的信息格式以外, 控制器与设备之间连接的顺序也不同, 以往 OpenFlow 是需要控制器开启特定的接口, 然后设备才能连上控制器, P4 Runtime 则是在设备上开启 RPC server, 由控制器联系设备, 因此设备上会有一个 Agent 负责处理由控制器发来的请求, 完成与控制器互连. 上述路径规划求解后的路径通过 P4 Runtime[16]下发至 BMv2[17]交换机, 交换机按照规则修改并转发探测包.

2.2 探测路径的自适应调整机制

探测路径自适应调整机制主要负责在拓扑发生变化时, 及时根据拓扑变化情况, 在局部或整体上调整探测路径, 以降低因拓扑改变导致的探测路径调整时间. 主要包括网络状态动态监测、探测路径局部调整、探测路径整体调整 3 部分.

2.2.1 网络状态动态检测

网络动态检测实时监控网络拓扑, 并在拓扑发生变化时, 及时上报控制器, 主要由两部分功能组成: 实时链路状态收集和链路状态上传汇报. 其中, 链路状态收集主要通过 LLDP 协议实现, 链路状态上传汇报主要通过 P4 Runtime 来完成.

LLDP[18]定义在 802.1ab 中, 提供了一种标准的链路层发现方式. LLDP 协议使得接入网络的一台设备可以将其管理地址、设备标识、接口标识等信息发送给接入同一个局域网络的其他设备. 当一个设备从网络中接收到其他设备的这些信息时, 它就将这些信息以 MIB 的形式存储起来. 这些 MIB 信息可用于发现设备的物理拓扑结构以及管理配置信息.

网络设备通过 LLDP 获取网络中其他设备的 MIB 信息, 并将相关信息通过 P4 Runtime 上传至控制器, 从而收集到网络拓扑, 以及网络设备之间的实时链路状态.

2.2.2 探测路径局部调整

探测路径局部调整是在网络设备或网络链路发生故障, 且已发生故障的子图数量占子图总数的比例低于一定阈值 (此阈值将在后续实验中探究) 时, 从局部调整恢复带内网络遥测探测路径.

由于带内网络遥测探测路径固定, 一旦一条路径中的一台交换机出现故障 (如图 4), 会导致路径中其他正常交换机也无法完成带内网络遥测. 而直接重新计算并部署探测路径, 会给网络带来很大的开销. 考虑到我们对网络拓扑进行了子图分割, 单独的一个或几个无法正常工作的交换机可能只影响某几个子图中的探测路径, 对受影响的子图中的探测路径进行局部调整, 从而减少探测路径的恢复时间.

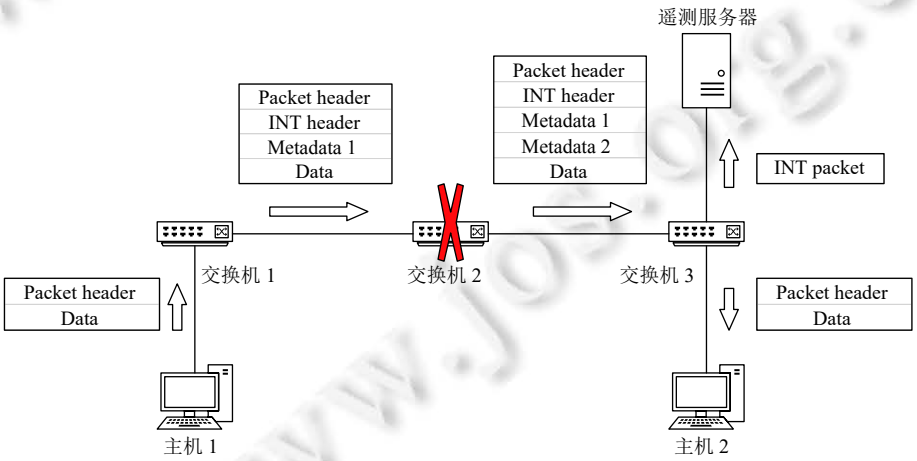


图 4 带内网络遥测故障示例

具体来说, 对于发生变化的拓扑, 考虑其影响会集中于少数几个拓扑子图中, 局部利用生成树算法调整探测路径, 即在拓扑变化的子图中, 若源节点未发生故障, 则以已有源节点为生成树的根, 采用最小生成树算法, 重新计算探测路径; 若源节点发生故障, 则随机选择一个交换设备作为生成树的根, 利用最小生成树算法, 重新计算探测路

径. 探测路径局部调整会减轻由于交换机节点在测量中角色发生变化带来的开销.

2.2.3 探测路径整体调整

探测路径整体调整是在网络设备或网络链路发生故障, 且已发生故障的子图数量占子图总数的比例超过一定阈值 (此阈值将在后续实验中探究) 时, 从整体调整恢复带内网络遥测探测路径.

由于探测路径局部调整并不能保证子图内的探测路径是冗余度低且路径长度短的最优解, 当拓扑改变过大时, 将导致存在很多非最优的探测路径, 网络状态测量效率下降. 所以在超过一定比例的子图发生改变后, 需要重新对探测路径进行整体调整, 即重新执行基于图分割的探测路径部署机制. 至于何时触发探测路径的整体调整, 需要在实验中进行尝试来寻找合适的值.

3 实验分析与验证

3.1 实验环境

实验所用环境如图 5 所示, 控制器使用开源控制器 ONOS[19], 负责探测路径计算与下发, 配置为 Ubuntu 16.04, 4 GB 内存; 交换机采用使用 Mininet[20]环境中行为模型 BMv2 模拟虚拟交换机, 负责完成带内网络遥测中不同的节点功能 (源节点、转发节点、汇聚节点); 遥测结果计算工具使用开源项目 INTCollector[21], 负责根据汇聚节点上传的包含所有交换设备元数据的 INT Report 计算出最终的遥测结果, 上传至数据库, 并提供给控制器上层应用使用. 实验中主要采用的拓扑图为数据中心中常用的 FatTree[22]拓扑, 如图 6 所示. 部分实验中为了更直观地对比评价指标相对交换设备数量的变化趋势, 通过水平扩展其中边缘层、核心层以及汇聚层的节点, 来增加拓扑中的交换设备数量. 对于需要改变交换设备数量的实验, 交换机数量取 10–100 之间; 对于不需要变化交换机数量的实验, 使用含有 50 个交换设备的 FatTree 拓扑. 对比方案选用生成树算法[4]、欧拉回路算法[5].

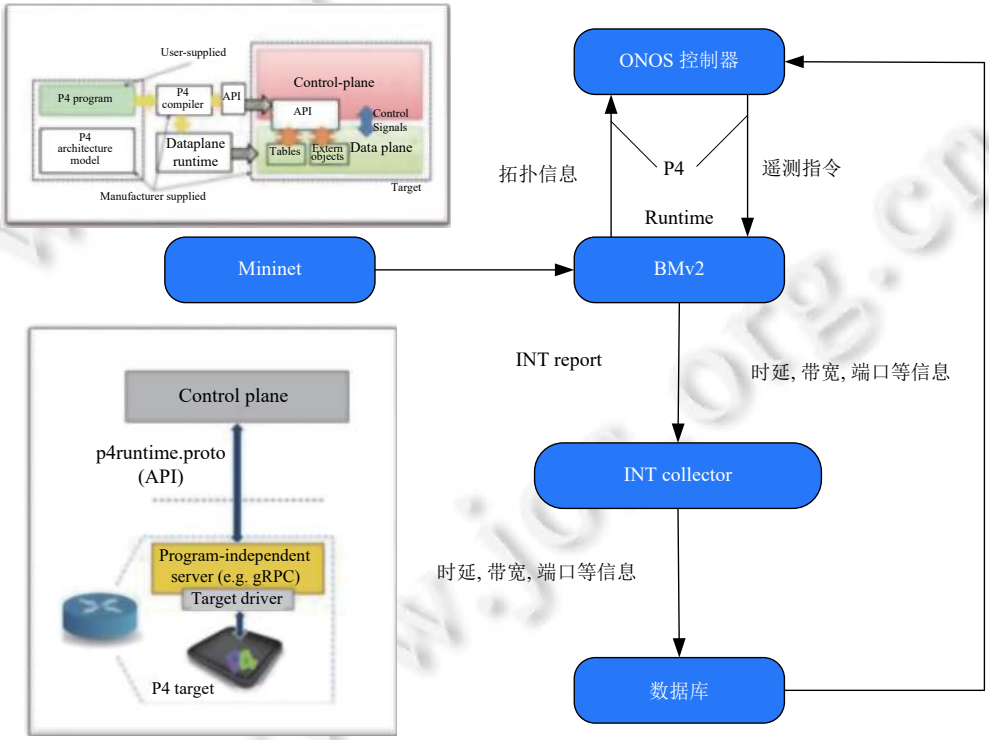


图 5 全网覆盖的带内网络遥测实验环境

实验参数包括: INT 探测节点数量、INT 探测路径平均长度、INT 报文丢包率、INT 元数据流量占比[23]、INT 探测路径冗余度, 以及探测路径调整时间等.

● INT 探测节点数量包括带内网络遥测的源节点和汇聚节点的总数.

● INT 探测路径平均长度为探测路径总长度除以探测路径数量.

● INT 报文丢包率是网络中丢失的 INT 探测包个数总和与 INT 探测数据包个数总和的比值.

● INT 元数据流量占比的计算方式如公式 (2) 所示, 分子为一段时间内网络中所有流量中的 Metadata 部分的长度总和, 分母为这段时间内所有流量的长度总和, 计算出来的百分比可以代表带内网络遥测为网络流量带来的负荷, 此百分比越大, 代表带内网络遥测为整体网络流量带来的负荷越大.

● INT 探测路径冗余度的计算方式如公式 (3) 所示, 分子为探测路径总长度, 即所有子图中的探测路径长度总和, 分母为实验所用网络拓扑中的链路个数, 此参数代表了探测路径的冗余程度. 同一拓扑下链路总数一定, 即公式中的分母一定, 而拓扑图中的链路被重复测量的次数越多, 分子越大, INT 探测路径冗余度越高. INT 探测路径冗余度的最优值为 1, 表示探测路径不存在冗余.

● 探测路径调整时间为链路改变的时间与经过调整后带内网络遥测重新开始工作时间的差值.

$$\text{INT元数据流量占比} = \frac{\sum Length_{\text{IM}}}{\sum Length_{\text{Packet}}} \tag{2}$$

$$\text{INT探测路径冗余度} = \frac{\text{探测路径总长度}}{\text{拓扑链路数}} \tag{3}$$

上述参数中 INT 探测节点数量主要反映了带内网络遥测中数量过多的探测节点带来的维护开销问题; INT 探测路径平均长度和 INT 报文丢包率反映了带内网络遥测中长度过长的探测路径会引起探测包大小超过网络 MTU, 导致探测包丢包的问题; INT 元数据流量占比和 INT 路径冗余度反映了带内网络遥测中大量冗余的探测路径会导致测量引入的流量负荷在网络整体流量中占比过大的问题; 探测路径调整时间反映了链路发生改变后探测路径的恢复速度.

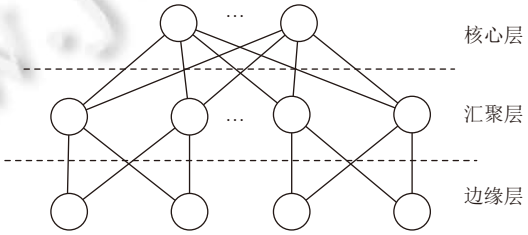


图 6 FatTree 拓扑示例

3.2 实验结果分析

3.2.1 基于图分割的探测路径部署机制

子图分割阈值的选择会影响部署时 INT 探测流量在网络整体流量中的占比和所需探测节点的数量. 图 7 分析不同子图分割阈值时 INT Metadata 网络中流量占比和 INT 探测节点数量的变化情况. 随着子图分割阈值的增加, INT Metadata 网络中流量占比持续增加, 而 INT 探测节点数量下降速度逐渐减缓. 在子图阈值为 10 时, 继续增加子图分割阈值对 INT 探测节点数量影响不大, 但会增加 INT Metadata 网络中流量占比, 所以在实验环境下, 子图阈值选为 10 即能保证 INT 探测节点数量较少, 又使得 INT Metadata 网络中流量占比较小. 后续实验将选取 10 (记为图分割 (10)) 作为子图分割的阈值, 同时为了证明在不同子图分割阈值下 ACGS 方法的表现, 还将选取 5 (记为图分割 (5)) 作为子图分割阈值作为对照实验.

图 8 显示了 INT 探测节点数量随交换机数量变化的情况. 如图 8 所示, 欧拉回路算法[5]在偶数个交换机的 FatTree 拓扑中始终是一个探测节点, 而在奇数个交换机的拓扑中探测节点数量非常多, 这种现象是欧拉回路算法导致的: 根据欧拉回路存在的充要条件, 偶数个交换机的 FatTree 拓扑一定存在一条能够覆盖所有边的欧拉回路; 而奇数个交换机的 FatTree 拓扑中利用 Hierholzer 算法求解出的覆盖全网的欧拉回路有多条, 所以这个算法在控制探测节点数量方面较为不稳定. 生成树算法[4]的 INT 探测节点数量随着交换机数量逐渐升高, 维护开销逐渐增大. ACGS 方法在以 5 作为子图阈值时 INT 探测节点数量与生成树算法相近, 在以 10 作为子图阈值时 INT 探测

节点数量可以维持在一个较小的范围内, 即相对上述两种方法有更小的维护开销. 这是因为 ACGS 方法每个子图中只存在一条探测路径, 且源节点与汇聚节点是同一个交换设备, 所以能够解决带内网络遥测中数量过多的探测节点给控制器带来较大维护开销的问题.

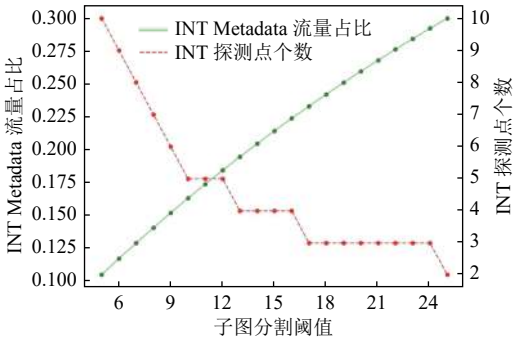


图 7 Metadata 流量占比与探测点个数对比图

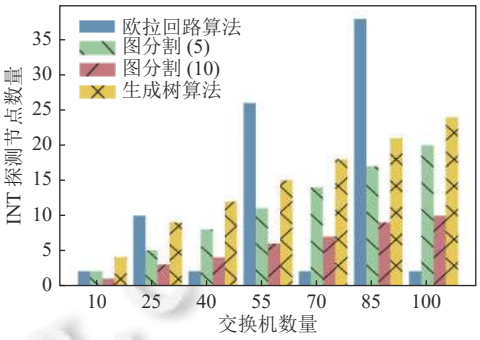


图 8 探测节点数量对比图

图 9、图 10 分析了 INT 探测路径平均长度和 INT 报文丢包率的变化情况. 图 9 显示了 INT 探测路径平均长度随交换机数量变化的情况. 可以明显看出: 生成树算法[4]和 ACGS 方法的 INT 探测路径平均长度基本可以维持在一个稳定的范围内; 而欧拉回路算法[5]的 INT 探测路径平均长度有较大范围的浮动, 其原因还是上述交换机数量与 INT 探测节点数量的关系中, 奇偶探测节点个数导致的; 我们的方法在图分割时, 保证了每个子图的大小维持在一个阈值范围内, 所以其探测路径长度平均值维持在一个合适的范围内. 图 10 显示了 INT 报文丢包率随探测包初始大小变化的情况. 从图中可以看出: 欧拉回路算法[5]随着探测包初始大小的增大, 丢包率始终比其他算法更大; 生成树算法[4]次之; ACGS 方法在以 5 和 10 为子图阈值时都能够保持更小的丢包率. 图 9 和图 10 说明 ACGS 通过在图分割时控制拓扑子图的规模, 从而限制了子图中探测路径长度, 解决了由此引起的探测包大小超过网络 MTU 导致的探测包丢包率增高的问题.

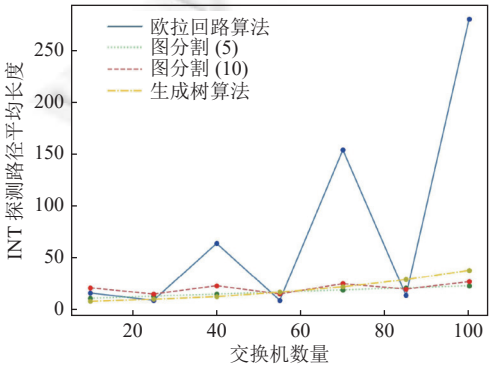


图 9 探测路径平均长度对比图

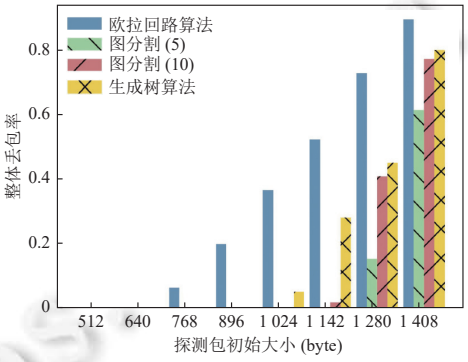


图 10 INT 报文丢包率对比图

图 11、图 12 分析了 INT 路径冗余度和 INT 元数据流量占比的变化情况. 图 11 显示了 INT 路径冗余度随交换机数量变化的情况. 由图 11 可见: 欧拉回路算法[5]因其算法特殊性可以保证 INT 路径冗余度始终为 100%; 而 ACGS 方法在 INT 路径冗余度上优于生成树算法[4]. 分析其原因为我们在拓扑子图中求解欧拉回路得到只遍历子图中有向边一次的探测路径, 解决了全网覆盖的带内网络遥测中大量冗余探测路径的问题. 图 12 显示了 INT 元数据流量占比随探测包初始大小变化的情况, 由图 12 可见: ACGS 方法在子图阈值为 5 时, INT Metadata 在网络整体流量中的占比与生成树算法[4]大小相近; 子图阈值为 10 时, 占比稍高一些; 欧拉回路算法[5]在所有实验中的占比最高. 图 11、图 12 说明了 ACGS 通过欧拉回路规划得到只遍历子图中有向边一次的探测路径, 解决了探测路径冗余以及由此引起的测量引入流量负荷在网络整体流量中占比过大的问题.

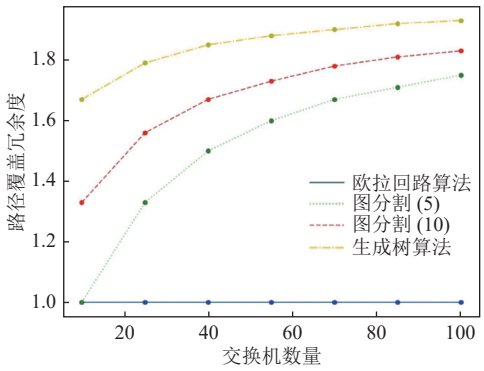


图 11　路径覆盖冗余度对比图

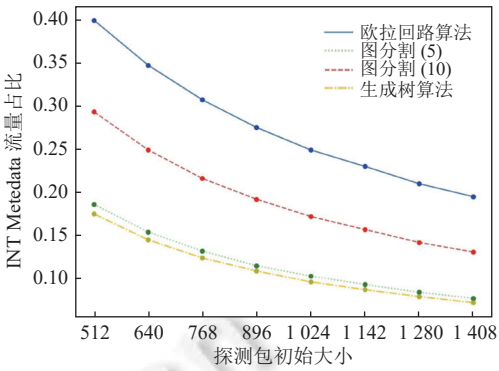


图 12　Metadata 流量占比对比图

3.2.2　探测路径自适应调整机制

图 13 分析了不同子图变化百分比对 INT 路径覆盖冗余度的影响. 实验中 ACGS 方法触发整体调整的子图变化比例分别取 25% (记为图分割 (25%))、50% (记为图分割 (50%))、75% (记为图分割 (75%)). 如图 13 所示: 触发整体调整的子图变化比例在取 25% 时, INT 路径冗余度基本稳定, 但是 25% 低触发值会带来频繁的整体调整, 图中横轴为 25%、50%、75% 时, 图分割 (25%) 发生整体调整, 随着整体调整次数的增加, 用于调整的时间也会随之增加; 当触发整体调整的子图变化比例取 75% 时, 只在图中横轴为 75% 时发生整体调整, 虽然降低了整体调整的次数, 但是会导致在拓扑图中大量子图发生变化时 INT 探测路径冗余度过大, 以及测量引入流量占比增大的问题; 当触发整体调整的子图变化比例为 50% 时, 只在图中横轴为 50% 时发生整体调整, 可以在保证调整次数较低的情况下, 将 INT 探测路径冗余度维持在一个可以接受的范围内. 所以后续实验中都使用 50% 作为触发整体调整的子图变化比例.

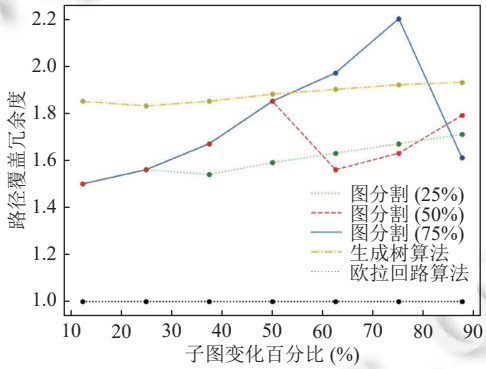


图 13　路径覆盖冗余度对比图

图 14 显示了链路改变后探测路径修复时间随交换机数量变化的情况. 此实验是针对交换机数量不同的网络拓扑, 随机改变一条链路, 即保证触发局部调整而非整体调整, 观察并记录链路变化后探测路径修复时间. 从图 14 可以看出: ACGS 方法在链路发生改变后, 用于调整的时间相较于另外两种算法更稳定, 保持在一个较低的范围内; 而欧拉回路算法[5]与生成树算法[4]的调整时间随着交换机数量的增加而增加, 对于较大规模的拓扑, 会导致调整时间更长. 可见, ACGS 通过局部调整的方式加快了探测路径调整速度.

图 15 显示了链路改变后探测路径调整时间随改变链路数量变化的情况, 可以说明整体调整时间是远大于局部调整时间的. 此实验是针对同一个拓扑, 改变拓扑中的不同链路, 观察并记录链路变化后探测路径修复时间. 需要注意的是, 对于 ACGS 方法, 为了体现子图改变的比例对调整时间的影响, 实验中保证每次改变的链路都属于不同拓扑子图. 从图 15 中可以看出: 欧拉回路算法[5]与生成树算法[4]链路改变后探测路径修复时间基本稳定; 而 ACGS 因局部调整与整体调整的拓扑规模不同, 所以路径调整时间是有较大差异的 (图分割 (5) 改变链路数量为

3 和 6 时是整体调整, 其余为局部调整; 图分割 (10) 改变链路数量为 5 时是整体调整, 其余为局部调整). 相对于欧拉回路算法[5]与生成树算法[4], 本文的局部调整速度更快, 整体调整速度比欧拉回路算法[5]稍慢一些.

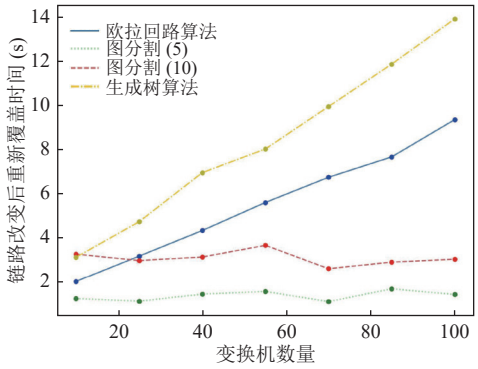


图 14 探测路径调整时间对比图 1

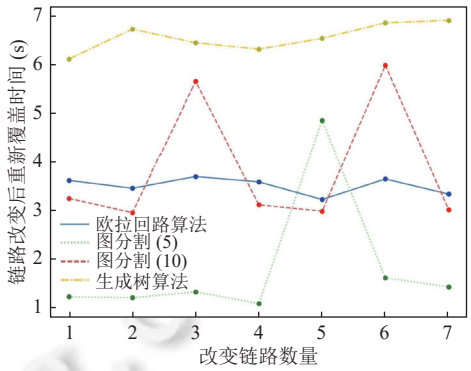


图 15 探测路径调整时间对比图 2

4 总结与展望

本文提出的 ACGS 方法基于图分割的划分方式将拓扑图划分为多个子图; 并基于欧拉回路得到能够覆盖拓扑子图的带内网络遥测探测路径; 结合局部调整与整体调整, 在动态拓扑下进行自适应探测路径调整, 以降低因拓扑改变导致的探测路径调整时间. 实验结果验证了 SDN 网络带内网络遥测探测路径配置方法的有效性, 后续研究主要在实际环境中部署应用 ACGS 方法, 并对其进行改进和优化.

References:

[1] Karakus M, Durresi A. Quality of service (QoS) in software defined networking (SDN): A survey. Journal of Network and Computer Applications, 2017, 80: 200–218. [doi: 10.1016/j.jnca.2016.12.019]

[2] Zhang H, Cai ZP, Li Y. An overview of software-defined network measurement technologies. SCIENTIA SINICA Informationis, 2018, 48(3): 293–314 (in Chinese with English abstract). [doi: 10.1360/N112017-00203]

[3] Kim C, Sivaraman A, Katta N, Bas A, Dixit A, Wobker LJ. In-band network telemetry via programmable dataplanes. ACM SIGCOMM, 2015, 15(1): 1–2.

[4] Atary A, Bremler-Barr A. Efficient round-trip time monitoring in OpenFlow networks. In: Proc. of the 35th Annual IEEE Int'l Conf. on Computer Communications. San Francisco: IEEE, 2016. 1–9. [doi: 10.1109/INFOCOM.2016.7524501]

[5] Liu ZZ, Bi J, Zhou Y, Wang YY, Lin YSX. NetVision: Towards network telemetry as a service. In: Proc. of the 26th IEEE Int'l Conf. on Network Protocols (ICNP). Cambridge: IEEE, 2018. 247–248. [doi: 10.1109/ICNP.2018.00036]

[6] Liu ZZ, Bi J, Zhou Y, Wang YY, Lin YSX. Paradigm for proactive telemetry based on P4. Journal on Communications, 2018, 39(S1): 2018181 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-436x.2018181]

[7] Pan T, Song EG, Bian ZZ, Lin XC, Peng XY, Zhang J, Huang T, Liu B, Liu YJ. INT-path: Towards optimal path planning for in-band network-wide telemetry. In: Proc. of the 2019 IEEE Conf. on Computer Communications. Paris: IEEE, 2019. 487–495. [doi: 10.1109/INFOCOM.2019.8737529]

[8] Bhamare D, Kassler A, Vestin J, Khoshkholghi MA, Taheri J. IntOpt: In-band network telemetry optimization for NFV service chain monitoring. In: Proc. of the 2019 IEEE Int'l Conf. on Communications (ICC). Shanghai: IEEE, 2019. 1–7. [doi: 10.1109/ICC.2019.8761722]

[9] van Tu N, Hyun J, Hong JWK. Towards ONOS-based SDN monitoring using in-band network telemetry. In: Proc. of the 19th Asia-Pacific Network Operations and Management Symp. (APNOMS). Seoul: IEEE, 2017. 76–81. [doi: 10.1109/APNOMS.2017.8094182]

[10] Ford Jr LR, Fulkerson DR. A simple algorithm for finding maximal network flows and an application to the Hitchcock problem. Canadian Journal of Mathematics, 1957, 9: 210–218. [doi: 10.4153/CJM-1957-024-0]

[11] Iranpoor M, Mohammaditabar D. Eulerian trails and tours. In: Farahani RZ, Miandoabchi E, eds. Graph Theory for Operations Research and Management: Applications in Industrial Engineering. IGI Global, 2013. 81–95. [doi: 10.4018/978-1-4666-2661-4.ch007]

[12] Haxhibeqiri J, Moerman I, Hoebeke J. Low overhead, fine-grained end-to-end monitoring of wireless networks using in-band telemetry.

In: Proc. of the 15th Int'l Conf. on Network and Service Management (CNSM). Halifax: IEEE, 2019. 1–5. [doi: 10.23919/CNSM46954.2019.9012678]

[13] Boykov Y, Funka-Lea G. Graph cuts and efficient N-D image segmentation. Int'l Journal of Computer Vision, 2006, 70(2): 109–131. [doi: 10.1007/s11263-006-7934-5]

[14] Queyranne M. Minimizing symmetric submodular functions. Mathematical Programming, 1998, 82(1): 3–12. [doi: 10.1007/BF01585863]

[15] Benson T, Akella A, Maltz DA. Network traffic characteristics of data centers in the wild. In: Proc. of the 10th ACM SIGCOMM Conf. on Internet Measurement. Nice: ACM, 2010. 267–280. [doi: 10.1145/1879141.1879175]

[16] Bosshart P, Daly D, Gibb G, Izzard M, McKeown N, Rexford J, Schlesinger C, Talayco D, Vahdat A, Varghese G, Walker D. P4: Programming protocol-independent packet processors. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 87–95. [doi: 10.1145/2656877.2656890]

[17] Dandavate V, Jinjala J, Keharia H, Madamwar D. Production, partial purification and characterization of organic solvent tolerant lipase from Burkholderia multivorans V2 and its application for ester synthesis. Bioresource Technology, 2009, 100(13): 3374–3381. [doi: 10.1016/j.biortech.2009.02.011]

[18] Alharbi T, Portmann M, Pakzad F. The (in) security of topology discovery in software defined networks. In: Proc.of the 40th IEEE Conf. on Local Computer Networks (LCN). Clearwater Beach: IEEE, 2015. 502–505. [doi: 10.1109/LCN.2015.7366363]

[19] Berde P, Gerola M, Hart J, Higuchi Y, Kobayashi M, Koide T, Lantz B, O'Connor B, Radoslavov P, Snow W, Parulkar G. ONOS: Towards an open, distributed SDN OS. In: Proc. of the 3rd Workshop on Hot Topics in Software Defined Networking. Illinois: ACM, 2014. 1–6. [doi: 10.1145/2620728.2620744]

[20] de Oliveira RLS, Schweitzer CM, Shinoda AA, Prete LR. Using mininet for emulation and prototyping software-defined networks. In: Proc. of the 2014 IEEE Colombian Conf. on Communications and Computing (COLCOM). Bogota: IEEE, 2014. 1–6. [doi: 10.1109/ColComCon.2014.6860404]

[21] van Tu N, Hyun J, Kim GY, Yoo JH, Hong JWK. INTCollector: A high-performance collector for in-band network telemetry. In: Proc. of the 14th Int'l Conf. on Network and Service Management (CNSM). Rome: IEEE, 2018. 10–18.

[22] Leiserson CE. Fat-trees: Universal networks for hardware-efficient supercomputing. IEEE Trans. on Computers, 1985, C-34(10): 892–901. [doi: 10.1109/TC.1985.6312192]

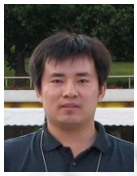
[23] Marques JA, Luizelli MC, Da Costa Filho RIT, Gaspary LP. An optimization-based approach for efficient network monitoring using in-band network telemetry. Journal of Internet Services and Applications, 2019, 10(1): 12. [doi: 10.1186/s13174-019-0112-0]

附中文参考文献:

[2] 张恒, 蔡志平, 李阳. SDN网络测量技术综述. 中国科学: 信息科学, 2018, 48(3): 293–314. [doi: 10.1360/N112017-00203]

[6] 刘争争, 毕军, 周禹, 王旸旸, 林耘森箫. 基于P4的主动网络遥测机制. 通信学报, 2018, 39(S1): 2018181. [doi: 10.11959/j.issn.1000-436x.2018181]

原鹏翼(1996－), 男, 硕士生, 主要研究领域为软件定义网络.



张玉军(1976－), 男, 博士, 研究员, CCF 高级会员, 主要研究领域为计算机网络.

王淼(1975－), 女, 博士, 副研究员, CCF 专业会员, 主要研究领域为未来网络体系结构, 网络智能调度.

周继华(1979－), 男, 博士, 研究员, 主要研究领域为通信网络, 5G/6G.

王凌豪(1996－), 男, 博士生, 主要研究领域为未来网络体系结构, 流量工程.