

# 动态网络环境下的链路层拓扑发现算法<sup>①</sup>

李祝红<sup>1</sup>, 赵灿明<sup>1</sup>, 石 滚<sup>2</sup>, 田 野<sup>2</sup>

<sup>1</sup>(国网安徽省电力公司芜湖供电公司 信通公司, 芜湖 241000)

<sup>2</sup>(中国科学技术大学 计算机科学与技术学院, 合肥 230027)

**摘 要:** 完整准确的网络拓扑是网络管理的基础. 由于链路层设备的多样性、透明性、动态性, 因此如何快速、准确地发现链路层拓扑成为近年来研究的热点和难点. 现有的链路层拓扑发现研究均基于静态网络, 然而随着无线网络的普及, 网络中的移动设备越来越多, 静态网络的假设不再成立. 鉴于此, 本文提出了动态链路层拓扑发现算法, 通过在网路中部署有限的 agent 节点, 发现链路层拓扑. ns-3 模拟实验以及对实际运行网络的测量结果表明, 该算法能有效排除移动设备的影响, 快速准确地发现链路层拓扑结构.

**关键词:** 链路层拓扑; 拓扑发现; 地址转发表; 简单网络管理协议(SNMP)

## Link-Layer Topology Discovery Algorithm under Dynamic Networking Environments

LI Zhu-Hong<sup>1</sup>, ZHAO Can-Ming<sup>1</sup>, SHI Gun<sup>2</sup>, TIAN Ye<sup>2</sup>

<sup>1</sup>(Information & Telecommunication Branch, State Grid Wuhu Power Supply Company, Wuhu 241000, China)

<sup>2</sup>(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

**Abstract:** Complete and accurate network topology is the basis for the network management. With the increasing diversity, transparency, dynamics of the link layer devices, discovering the link-layer network topology becomes increasing challenging. Existing works on link-layer topology discovery is based on the assumption that the network is static without mobile devices. However with popularity of wireless access, there are more and more mobile devices, making the network more and more dynamic. In this paper, we propose an algorithm for discovering the link-layer topology under dynamic networking environments. Simulation experiments on ns-3 and the real-world networking topology measurement results indicate that our algorithm could exclude the impact of mobile devices and discover the linker-layer topologies efficiently and accurately.

**Key words:** link-layer topology; topology discovery; address forwarding table; simple network management protocol (SNMP)

## 0 引言

随着计算机网络规模的不断扩大, 网络的结构日趋复杂, 对网络的管理变得日益重要. 网络的故障管理、性能管理和网络升级都需要借助完整准确的网络拓扑结构信息. 同时网络拓扑也为网络的研究提供了较有价值的参考. 因此怎样获得完整准确的网络拓扑一直是计算机网络研究中的一个热点和难点.

目前已经有很多链路层拓扑的研究成果. Breitbart

等<sup>[1]</sup>基于标准的 SNMP 协议, 提出了一种在多子网和异构网络条件下的链路层拓扑发现算法, 并基于该算法开发了 NetInventory 系统. 但文献[1]中的算法依赖网络中所有交换机上 FDB(Forwarding database)的完备性, 这在大规模网络中并不现实. Lowekamp 等<sup>[2]</sup>在 Breitbart 算法的基础上, 针对大规模网络中普遍存在的 FDB 不完备, 提出了一种拓扑发现算法, 并基于该算法开发了 REMOS 系统. Black 等<sup>[3]</sup>考虑在缺乏网络

① 基金项目:国家自然科学基金(61202405,61103228)

收稿时间:2015-02-02;收到修改稿时间:2015-03-12

作系统中. Li 等<sup>[9]</sup>提出了一种基于 VLAN 发现异构链路层拓扑发现的方法, 该方法首先通过分析地址转发表(Address Forwarding Table, AFT)获得交换机的连接, 然后再通过查询MIB获得不同设备的VLAN拓扑信息, 最后建立拓扑. 然而该方法依赖于完整的AFT和MIB, 一般难以满足. 文献[15]-[18]基于AFT和STP发现链路层拓扑. 孙延涛等<sup>[10]</sup>提出了由网络节点的直系关系和旁系关系判断交换机之间的连接关系, 该算法能够利用不完整的地址转发表构造出网络拓扑图. 陈松等<sup>[11]</sup>采用树表示交换机拓扑连接关系, 利用生成树集合演绎树的变化过程来完成拓扑发现. Xiong 等<sup>[12]</sup>提出了一种基于SNMP MIB的链路层拓扑推算的方法, 但是在大型网络中获得完整的MIB是比较困难的. Gobjuka 等<sup>[13]</sup>证明了基于不完整的信息发现链路层拓扑是一个NP问题, 然后提出了Half-Complete-RSs, Connection等启发式算法来生成唯一的链路层拓扑. Bejerano<sup>[14]</sup>基于交换机的MIB信息利用文中的skeleton-tree creation算法推出多子网络的骨架拓扑, 然后再根据文中的拓扑发现方法细化每个子网的内部结构, 最后形成完整的链路层拓扑. 上述工作均假设在运行拓扑发现算法时网络是静态的, 并不考虑网络中存在移动节点的情况. 此外, Nassu 等<sup>[21]</sup>提出了一种基于移动的测量agent和群体智能发现动态网络拓扑的方法; 文献[22][23]提出基于移动代理发现无线网络的动态拓扑. 这些工作尽管针对包含移动节点的网络,

但脱离具体的链路层协议, 仅仅从图论角度讨论拓扑发现问题, 所提出的方法难以工程化.

近年来, 一些企业相继开发了链路层网络拓扑发现技术. 例如思科公司提出了思科发现协议(Cisco Discovery Protocol, CDP)<sup>[4]</sup>, 并在思科的路由器、交换机、网桥等产品中集成了该协议. 微软公司提出了链路层拓扑发现协议(Link Layer Topology Discovery Protocol, LLTD)<sup>[5]</sup>, 并将其集成到Windows操作系统中. 惠普公司的OpenView系统<sup>[6]</sup>, IBM公司的Tivoli网络管理系统<sup>[7]</sup>, 上海北塔软件公司的BTNM网络管理系统<sup>[8]</sup>均提供以太网拓扑发现功能. 然而, 这些系统均基于私有的技术方案, 其准确性、可靠性特别是可验证性难以保障.

目前已知的所有方法均假设在运行拓扑发现算法时, 网络是静态的. 然而随着无线技术的普及, 网络中移动终端越来越多, 静态网络的假设不再被满足. 大量的移动终端使网络上很多地址时刻处于迁移之中, 导致在执行拓扑发现算法时, 从不同交换机上获得的FDB反映的是网络在不同时刻的视图, 从而算法无法获得正确的拓扑结构. 例如在图1中, 当拓扑发现程序请求交换机B在端口j上的FDB时, 获得 $F_B^j = \{X, Z\}$ , 此时移动终端Z接入交换机A; 而当拓扑发现程序请求交换机A在端口i上的FDB时, 由于Z已经发生移动, 接入交换机B, 因此 $F_A^i = \{Y, Z\}$ . 显然, 现有的算法都无法发现A<sub>i</sub>与B<sub>j</sub>的连接.

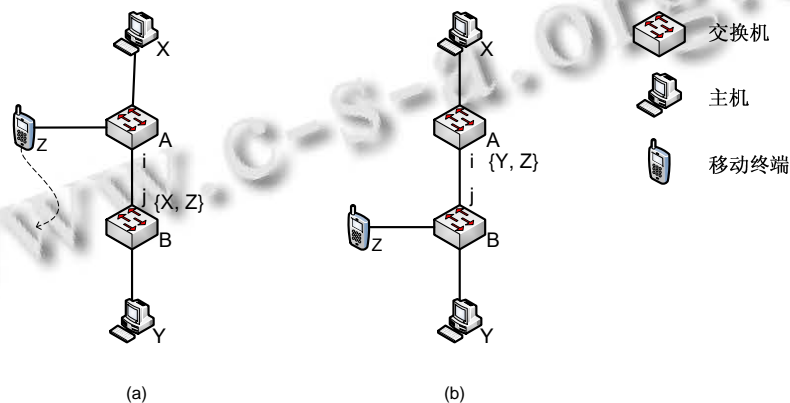


图1 移动终端示例

鉴于此, 本文提出了动态拓扑发现算法, 并且使用ns-3<sup>[19]</sup>进行模拟实验和对实际运行网络链路层拓扑的测量, 实验结果表明该算法能够排除设备移动的影响, 较快地发现完整的链路层拓扑. 基于实际应用环

境, 本文针对以下网络开展以太网拓扑发现算法的研究:

①具备网络支持: 本文假设网络中所有设备支持SNMP标准协议, 提供MIB标准所定义的各类信息,

网络中没有非合作节点. 这一假设符合大多数企业网络的实际情况.

②具备有限的终端支持: 本文假设可以控制网络中的一些主机, 从这些主机上发起探测; 然而不能主动地在网络上特定位置部署受控主机. 这一假设也符合大多数企业网络的实际情况.

③大规模异构网络: 网络规模较大, 使得交换机 FDB 不具有完备性; 而在异构网络上, 网络设备由多家设备商提供, 拓扑发现算法仅能够依赖所有设备都支持的标准 SNMP 协议和 MIB 信息库, 以及网络节点的标准化行为.

④移动网络环境: 网络中具有较多的移动终端, 因此拓扑发现算法必须避免地址迁移带来的 FDB 不一致性.

### 1 动态拓扑发现算法

#### 1.1 探测节点部署算法

本文提出的以太网拓扑发现算法基于一个管理 agent 和多个探测 agent, 这两类 agent 均运行在主机上. 不失一般性, 本文假设管理 agent 连接以太网树状拓扑的根节点交换机, 而多个探测 agent 位于网络的其它位置. 本文不对探测 agent 的部署范围做任何假设. 例如在图 2 中, *M* 代表管理 agent, *A-F* 共 6 个探测 agent 分别连接网络上不同的交换机节点, 而网络中多个交换机节点没有连接任何探测 agent. 其中管理 agent 的主要功能包括: 从本地发起主动探测; 指示探测 agent 发起主动探测; 对网络节点发起 SNMP 查询, 获得查询结果; 运行拓扑发现算法, 输出所发现的网络拓扑. 探测 agent 的主要功能是接收管理 agent 指令, 发起主动探测.

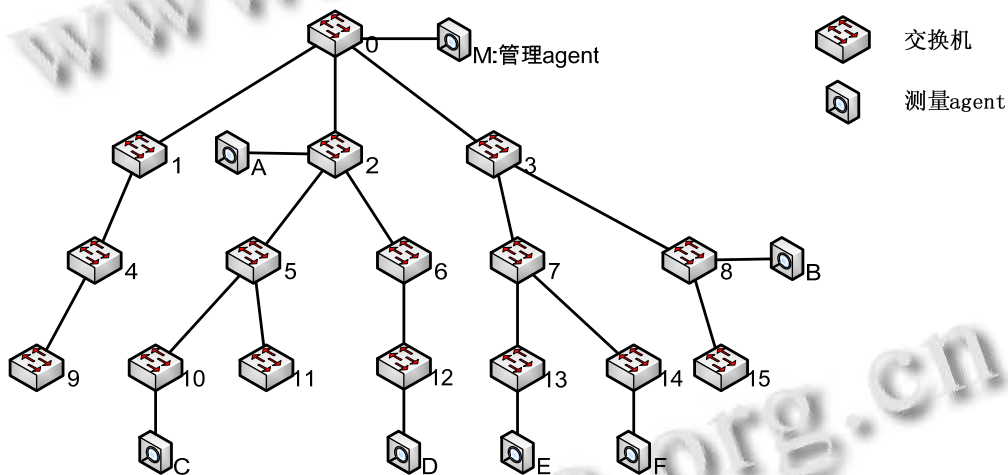


图 2 测量系统部署示意图

#### 1.2 路由器和交换机发现算法

本文使用以下方法发现所有路由器和路由器形成的第三层拓扑: 维持一个“未访问路由器集合”*R*, 开始时 *R* 中仅包含初始的路由器 *R*; 从 *R* 开始, 查询 ipRouteTable<sup>[20]</sup> 变量获取其路由表, 对每一条路由, 如果其下一跳地址对应的路由器 *R<sub>i</sub>* 没有被访问过, 则将 *R<sub>i</sub>* 加入到 *R*; 完成对 *R* 的路由表的访问后, 将 *R* 从 *R* 移除; 从 *R* 中选取一个未访问的路由器, 重复上述过程; 直到完成对目标网络上所有路由器的访问, 即 *R* 成为空集.

对每一个路由器上的每一个接口, 查询 ipAdEntAddr<sup>[20]</sup> 变量获取其连接的子网, 对子网中所

有的 IP 地址, 查询 system.sysServices<sup>[20]</sup> 变量和 ipForwarding<sup>[20]</sup> 变量判定该设备是交换机还是路由器.

对于子网中每一个交换机 IP 地址, 查询变量 ipNetToMediaTable<sup>[20]</sup> 获取其 MAC 地址、系统名称、端口数等信息, 获得交换机集合 *S*. 对于子网中每一个主机 IP 地址, 查询变量 ipNetToMediaTable<sup>[20]</sup> 获取其 MAC 地址, 获得主机集合 *H*.

#### 1.3 框架拓扑发现算法

拓扑发现系统中的 agent 通过主动发送探测报文, 获取拓扑结构信息. 以下, 本文使用标记 *A: X → B* 表示 agent *A* 以 *X* 为源 MAC 地址, 以 *B* 为目标 MAC 地址发送探测报文. 显然, 如果 *A ≠ M*, 则 agent *A* 需

要伪造探测报文。

管理 agent  $M$  向集合  $\mathbf{H}$  中每台主机  $H$  发送探测报文,  $M : M \rightarrow H, H \in \mathbf{H}$ ; 查询交换机集合  $\mathbf{S}$  中每台交换机的 FDB. 显然, 对任意交换机  $S \in \mathbf{S}$ , 仅存在一个端口  $i, M \in F_S^i$ , 则将该端口成为交换机  $S$  的**根端口**, 其余端口称为交换机的**指派端口**. 框架拓扑发现算法如算法 1 所示.

算法 1: 框架拓扑发现算法

---

Input: switch set ( $\mathbf{S}$ ), probe agent set ( $P$ )  
 Define:  $cover\_dict = \{\}, frame\_topo = \{\}$   
 Steps:

1. for each probe agent  $pa$  in  $P$ ://初始化  $cover\_dict$
2.      $cover\_dict[pa] = \{\}$
3. end for
4. for each switch in  $\mathbf{S}$ ://查询每个交换机的 FDB, 计算每个探测 agent 的覆盖集
5.     for each entry  $item$  in FDB of  $switch$ :
6.          $pa = \text{get mac of probe agent learned of } item$
7.          $cover\_dict[pa].insert(switch)$
8.     end for
9. end for
10. while intersection of cover set of any two probe agents is not empty://计算所有探测 agent 的覆盖集的
11.      $min\_intersection = \text{compute minimum intersection of each probe agent}$  //最小交集, 直到没有交集
12.     add  $min\_intersection$  into  $frame\_topo$  //将最小交集集中的节点视为一个交换机加入框架拓扑中
13.     for each probe agent  $pa$  in  $cover\_dict$ ://删除探测 agent 的覆盖集中最小交集包含的节点
14.         delete elements from  $pa$ 's cover set which both in  $min\_intersection$  and cover set
15.     end for
16. end while
17. draw frame topology based on  $frame\_topo$

---

每个探测 agent 使用独特的地址  $X$  向管理 agent  $M$  发送报文, 即对探测 agent  $A, A : X_A \rightarrow M$ .  $X_A$  是不存在与网络中的探测专用地址, 并且每个探测 agent 有不同的探测专用地址. 探测 agent  $A$  发送报文后,  $M$  查询所有交换机 FDB. 如果在某交换机  $S$  的 FDB 中发现地址  $X$ , 则称该  $S$  被  $A$  覆盖, 且  $X$  所在

的端口  $A$  的**面向端口**. 对 agent  $A$ , 所有它所覆盖的交换机及其面向端口形成一个集合, 被称为  $A$  的覆盖集  $S_A$ . 例如在图 2 中,  $S_A = \{0, 2\}$ 、 $S_B = \{0, 3, 8\}$ 、 $S_C = \{0, 2, 5, 10\}$ 、 $S_D = \{0, 2, 6, 12\}$ 、 $S_E = \{0, 3, 7, 13\}$ 、 $S_F = \{0, 3, 7, 14\}$ .

迭代地, 从所有探测 agent 覆盖集中寻找最小交集, 这个交集被称为一个可能包含多个实际交换机的“虚拟交换机”; 从各覆盖集中移除虚拟交换机包含的节点, 得到的剩余覆盖集记为  $S_A^{(i)}$ , 其中  $i$  是当前的迭代次数. 反复迭代, 直到没有交集产生, 将剩下的剩余覆盖集也视为“虚拟交换机”, 并连接相应的探测 agent. 例如, 对图 2 所示的网络, 每一轮迭代得到的虚拟交换机和剩余覆盖集如图 3 所示. 最终得到的目标网络拓扑称为“框架拓扑”, 框架拓扑存在两个缺陷: 虚拟交换机可能包含多个物理交换机节点, 如图 3 中 (5,10)、(6,12), 需要确定虚拟交换机内部的结构; 没有被任何探测 agent 覆盖的那些交换机节点及其拓扑结构, 如图 3 中交换机 1、4、9、11、15, 仍然有待确定. 框架拓扑发现算法的时间复杂度主要取决于探测 agent 数 ( $n$ ) 和网络中交换机的数目 ( $m$ ), 时间复杂度为  $O(mn+n^2)$ . 一般情况下, 一个以太网内的交换机数不会很大, 同时第二节的实验结果也说明本文的算法也不需要较多探测 agent, 因此算法的效率是比较高的.

#### 1.4 判别共享网段上的探测 agent

在上一节的框架拓扑发现算法中, 通过若干探测 agent 获得框架拓扑. 然而如果某个探测 agent 通过共享网段(如 hub)接入网络, 则算法不再有效. 例如图 4 中, agent  $B$  的覆盖集合为  $S_A = \{0, 1, 2, 3, 4\}$ , 执行框架拓扑发现算法不能获得正确的结果.

本文提出一种简单的方法判别通过共享网段接入网络的探测 agent. 过程为: 探测 agent  $A$  向管理 agent  $M$  发送探测报文后,  $M$  向该探测  $A$  发送探测报文,  $M : X_M \rightarrow A$ , 其中  $X_M$  是  $M$  专用的特殊地址. 管理 agent 查询获得 FDB 中包含地址  $X_M$  集合, 如果该集合与  $S_A$  不同, 则判定  $A$  通过共享网段接入网络. 例如在图 4 中, 从  $M$  向  $A$  发送探测报文, 则包含地址  $X_M$  节点集合可能为  $\{1, 3\}$  或者  $\{2, 4\}$ , 显然此时两个集合不同. 对所有通过共享网段接入网络的探测 agent, 将其从 agent 集合中移出, 并不在框架拓扑发现算法中使用它们.

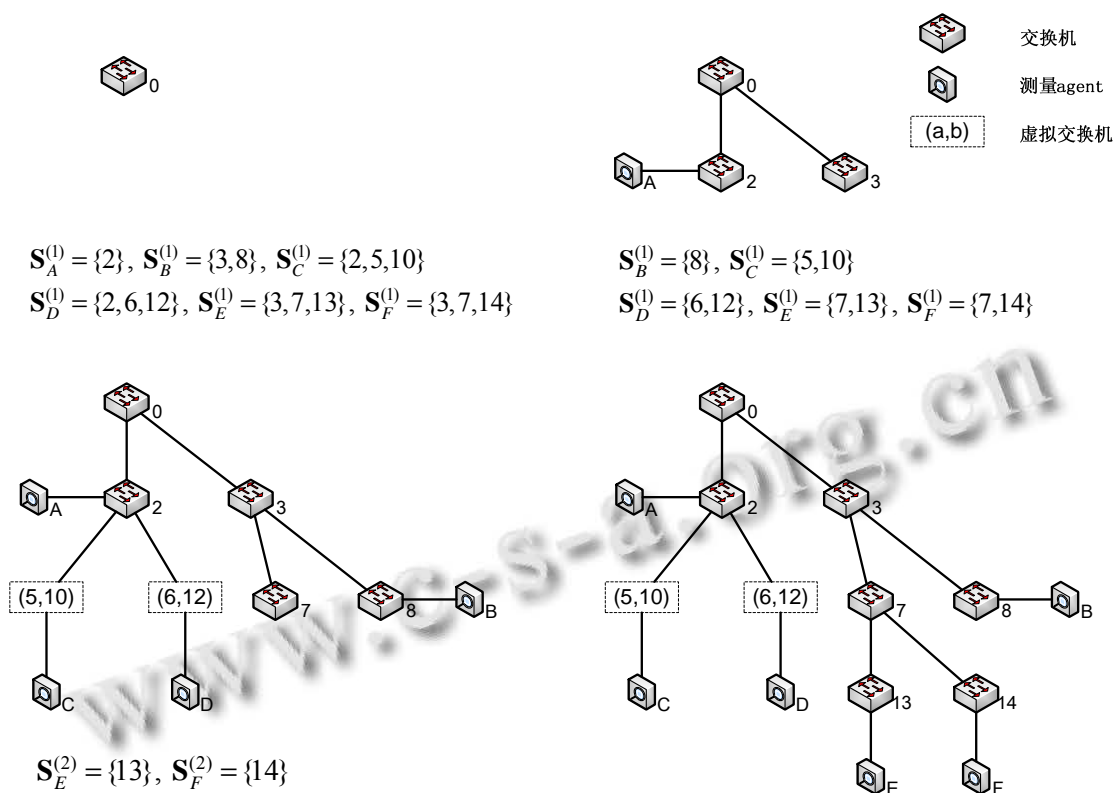


图 3 算法执行示意图

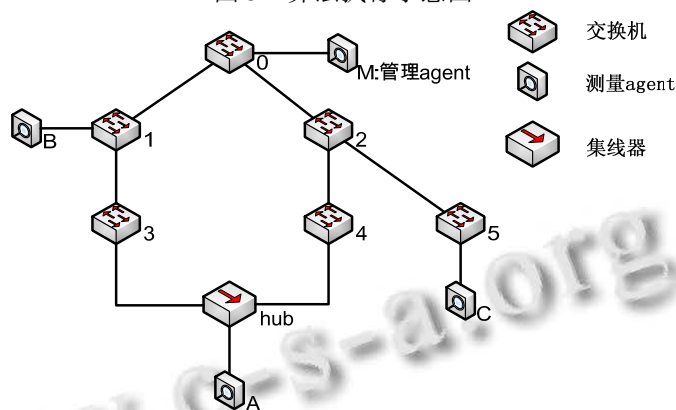


图 4 通过共享网段接入网络的探测 agent

### 1.5 判别移动主机

如前所述, 移动主机的地址随时间迁移, 难以直接应用文献[1][2][3]等算法获取以太网的拓扑结构. 基于已获得的框架拓扑结构, 可根据以下的规则发现存在迁移现象的地址.

规则 1: 在框架拓扑中, 假设  $U$  是  $V$  的下游节点, 如果一个地址  $X$  出现在虚拟交换机  $U$  某个指派端口的 FDB 中, 同时又出现在另一虚拟交换机  $V$  某个非面向  $U$  端口的 FDB 中, 则  $X$  是移动主机地址.

规则 2: 如果一个地址  $X$  同时出现在虚拟交换机  $U$  和  $V$  的指派端口的 FDB 中, 且  $U$  和  $V$  是兄弟节点

关系(存在共同祖先), 则  $X$  是移动主机地址.

规则 3: 如果一个地址  $X$  同时出现在虚拟交换机  $U$  和框架拓扑以外某个交换机节点  $V$  的指派端口的 FDB 中, 则  $X$  是移动主机地址.

将通过上述三个规则发现的移动地址从所有交换机的 FDB 中移除, 以去除移动主机地址对拓扑发现算法的影响.

### 1.6 完整的拓扑发现

在框架拓扑的基础上得到完整的拓扑, 需要: 确定虚拟交换机内部的线性结构; 确定框架拓扑之外的拓扑结构. 可以证明: 如果一个虚拟交换机包含多个

物理交换机节点,那么这些节点形成线性结构;如果一个交换机不在框架拓扑上,那么它最多只能和一个属于框架拓扑的交换机相连.因此对于包含  $n$  台物理交换机的虚拟交换机,可以将其视为一个小型的线性网络,仅需要发现正确的线性顺序  $V_1-V_2-\dots-V_n$ ,因此可以使用文献[2]中的算法,获取拓扑结构.对于框架拓扑之外的网络节点,首先寻找所有通过同一框架拓扑上交换机和端口连接到网络的交换机节点,将这些节点视为一棵子树,然后使用文献[2]中的算法,判定子树的拓扑结构.

## 2 实验结果

### 2.1 ns-3 模拟实验

本文使用 ns-3<sup>[19]</sup>进行拓扑模拟实验,模拟拓扑中主要包括交换机、无线网络节点、有线网络节点.表1显示的是每种拓扑4轮测量的平均测量周期,由此可以观察到本文的测量算法可以较高效地测量大型以太网链路层拓扑(拓扑中只有1个管理agent,表1中显示的是探测agent数).

表1 ns-3 模拟网络拓扑测量周期

交换机数	10	20	40	80	160
探测 agent 数	3	6	9	18	36
测量周期(秒)	5.7	13.6	78.7	461.1	1361.9

本文将动态拓扑发现算法发现的拓扑链路 with 预先定义的拓扑中相同的比率定义为拓扑准确率.为了验证本文测量算法的准确性,针对包含40和80台交换机的以太网,用15个探测agent和1个管理agent进行探测,然后逐渐增加网络中的无线节点,并且这些无线节点随机移动,观察移动节点对拓扑准确率的影响,其结果如图5所示.从图5可以观察到,本文的测量算法能够较好地排除移动节点的影响,可以得到比较准

确的拓扑.图6展示了探测agent数与拓扑准确率的关系,以太网中包含160个移动节点,从图6可以观察到,本文的测量算法在探测agent数较少的情况下也可以探测到较准确的拓扑.

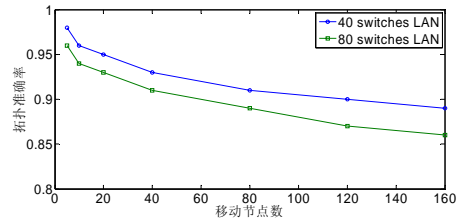


图5 移动节点数与拓扑准确率的关系

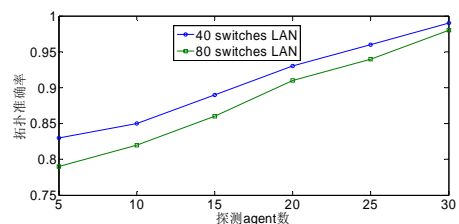


图6 探测agent数与拓扑准确率的关系

### 2.2 对实际运行网络链路层拓扑结构的测量发现

我们将所提出的方法应用到对芜湖市国家电网公司计算机网络的拓扑发现与管理中.芜湖市国家电网的内部以太网覆盖安徽省芜湖市的所有电网网点,骨干网络由40多台交换机组成,为整个城市的供电管理提供了通信基础设施.基于本文提出的动态网络拓扑发现算法和10个测量agent(1个管理agent,9个探测agent),我们开发了动态网络拓扑发现模块,并集成到公司的网络管理综合系统中.拓扑发现的功能界面如图7所示.我们使用所开发的模块开展了7轮拓扑测量(测量日期:2014/11/12-2014/11/22),7轮拓扑测量的测量周期和拓扑准确率如表2所示.实验结果表明系统测量具有较高的效率和准确率.

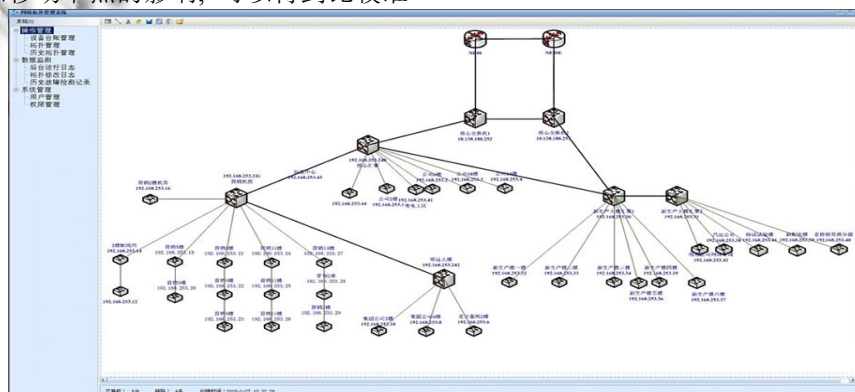


图7 网络链路层拓扑发现模块界面

表2 芜湖市国家电网网络拓扑测量周期和拓扑准确率

测量日期	11/12	11/14	11/17	11/18	11/22
拓扑准确率	0.83	0.85	0.86	0.84	0.84
测量周期(秒)	124	130	128	129	126

### 3 总结

基于目前无线网络逐渐普及的事实,本文提出了动态网络链路层拓扑发现算法,该算法使用的协议都是目前多数交换机所支持的,因此具有较高的适用性。ns-3 模拟实验和针对实际运行网络的拓扑测量表明该算法能够排除移动设备的影响,高效和准确地发现动态网络环境下的链路层拓扑结构。

#### 参考文献

- Breitbar Y, Garodalakis M, Jai B, et al. Topology discovery in heterogeneous IP networks: the netInventory system. IEEE/ACM Trans. on Networking, 2004, 12(3): 401-414.
- Lowekamp B, O'Hallaron DR, Gross TR. Topology discovery for large ethernet networks. Proc. of SIGCOMM'01. 2001.
- Black R, Donnelly A, Fournet C. Ethernet topology discovery without network assistance. Proc. of ICNP'04. 2004.
- Cisco Discovery Protocol Version 2. <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/cdp/configuration/15-nt/cdp-15-nt-book/nmcdp-discover.html>.
- Microsoft Link Layer Topology Discovery Protocol. [http://en.wikipedia.org/wiki/Link\\_Layer\\_Topology\\_Discovery](http://en.wikipedia.org/wiki/Link_Layer_Topology_Discovery).
- HP OpenView System. <http://h71000.www7.hp.com/>.
- IBM Tivoli system. <http://www-03.ibm.com/software/products/en/tivoli-netview-zos>.
- 上海北塔软件公司 BTNM 系统. <http://www.betasoft.com.cn/product/btnm/index.html>.
- Li DC, Chen M, Han CY, et al. Heterogeneous network topology discovery algorithm based on VLAN. Proc. of CISME'12. 2012.
- 孙延涛,石志强,吴志美.交换式以太网物理拓扑结构的自动发现.计算机研究与发展,2007,44(2):208-215.
- 陈松,王珊,周明天.一种新的物理网络拓扑发现算法.电子与信息学报,2010,32(1):172-177.
- Xiong YT, Gu ZJ, Jin W. Research and implement of an algorithm for physical topology automatic discovery in switched ethernet. Proc. of MPBE'12. 2012.
- Gobjuka H, Breitbart Y. Ethernet topology discovery for networks with incomplete information. IEEE/ACM Trans. on Networking, 2010, 18(4): 1220-1233.
- Bejerano Y. Taking the skeletons out of the closets: a Simple and efficient topology discovery scheme for large ethernet LANs. IEEE/ACM Trans. on Networking, 2009, 17(5): 1385-1398.
- 李琳,李杰.基于 SNMP 的网络拓扑发现算法.计算机工程与设计,2008,29(6):1345-1347.
- 毕艳冰,孙延涛,李涛.一种交换式以太网拓扑结构的发现算法.小型微型计算机系统,2008,29(1):35-40.
- 邓泽林,傅明,刘翌南.基于地址转发表的链路层拓扑发现算法.计算机工程与设计,2009,30(12):2938-2940.
- 陈琼,黄迪明,陈鑫.一种新型工业以太网拓扑结构发现方法.计算机应用研究,2006,23(7):246-248.
- ns-3. <http://www.nsnam.org/>.
- McCloghrie K, Rose M. Management information base for network management of TCP/IP-based internets: MIB-II. IETF RFC 1213. <http://www.ietf.org/rfc/rfc1213.txt>.
- Nassu BT, Nanya T, Duarte Jr EP. Topology discovery in dynamic and decentralized networks with mobile agents and swarm intelligence. Proc. of ISDA'07. 2007.
- 邓勇,王汝传,黄海平,徐喜春.基于移动代理的网络拓扑发现技术的研究.计算机科学,2008,35(10):76-81.
- 牛新征,梁帆,周明天.基于无线传感器的物联网网络拓扑发现算法研究.计算机科学,2012,39(4):118-122.