第 38 卷 第 2 期
Vol.38 No.2

计 算 机 工 程
Computer Engineering

2012 年 1 月
January 2012

• 网络与通信 •

文章编号：1000—3428(2012)02—0103—03 文献标识码：A 中图分类号：TP393

一种基于 SNMP 的链路层拓扑发现算法

潘 楠[a]，王 勇[b]，陶晓玲[c]

(桂林电子科技大学 a. 计算机科学与工程学院；b. CSIP 广西分中心；c. 信息与通信学院，广西 桂林 541004)

摘 要：为提高链路层网络拓扑发现效率，提出一种基于简单网络管理协议的拓扑发现算法。将交换机间的连接网络用树形结构表示，自顶向下逐层确定每个交换机的连接关系。通过修改连接关系的判定条件，并结合线程池和哈希查找技术，提高拓扑发现的效率。实验结果表明，该算法能快速准确地获得完整的网络拓扑结构。

关键词：链路层；拓扑发现；简单网络管理协议；地址转发表；线程池；哈希查找

Link Layer Topology Discovery Algorithm Based on Simple Network Management Protocol

PAN Nan[a], WANG Yong[b], TAO Xiao-ling[c]

(a. College of Computer Science and Engineering; b. CSIP Guangxi Center; c. College of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China)

【Abstract】In order to improve the network topology discovery efficiency of link layer, this paper proposes a discovery algorithm for link layer of topology based on Simple Network Management Protocol(SNMP). By describing the connections between switches as a tree, the connection relationship of each switch is established for every layer according to top-down manner. By improving the conditions of the connection between switches, combined with the thread pool and hash search to improve the efficiency of topology discovery. Experimental result indicates that the algorithm can discover link layer topology rapidly and completely, and the possible network elements can be discovered.

【Key words】link layer; topology discovery; Simple Network Management Protocol(SNMP); Address Forwarding Table(AFT); thread pool; hash search

DOI: 10.3969/j.issn.1000-3428.2012.02.033

1 概述

链路层拓扑发现是确定子网内部网络设备之间的连接关系，其中最主要的是确定交换机之间的连接关系。目前，基于简单网络管理协议(Simple Network Management Protocol, SNMP)的网络拓扑发现算法在国内外得到了广泛的研究[1-3]。文献[4-5]提出了基于地址转发表(Address Forwarding Table, AFT)的物理拓扑发现算法，但该算法要求所有交换机的地址转发表是完整的。文献[6]在此基础上提出了改进算法，充分利用不完整的 AFT 信息判断网络设备的连接关系，并能发现可能存在的“哑设备”，但该算法运算量较大。文献[7]通过将交换机端口分为上行端口和下行端口，降低了对 AFT 的要求，并提出了一条判定上行端口和下行端口直连的引理。本文在上述研究工作的基础上，提出一种新的链路层网络拓扑发现算法，通过改进交换机连接关系的判定条件，并借助于线程池和哈希查找等技术，提高拓扑发现的效率。

2 拓扑发现算法理论基础

本文将子网中连接在一台路由器下的所有交换机的连接结构看作一棵“树”，设与路由器直接相连的交换机为树的第 1 层节点，与第 1 层交换机直接连接的交换机为第 2 层节点，以此类推，将交换机的连接结构划分为多个层，每个层有若干个节点。每个交换机节点又会连接一个或多个主机。这种结构方式清楚地反映了网络的拓扑结构。图 1 是一种常见的网络拓扑结构。在交换机的结构树中，连接关系包括直接连接和间接连接。直接连接是指处于相邻位置的交换机相连接；而间接连接是指 2 个交换机之间存在着一条连接路径。直接连接实际上是间接连接的特殊形式。例如在图 1 中，Switch1 和 Switch2 属于直接连接关系，Switch1 与 Switch4 则是间接连接关系。

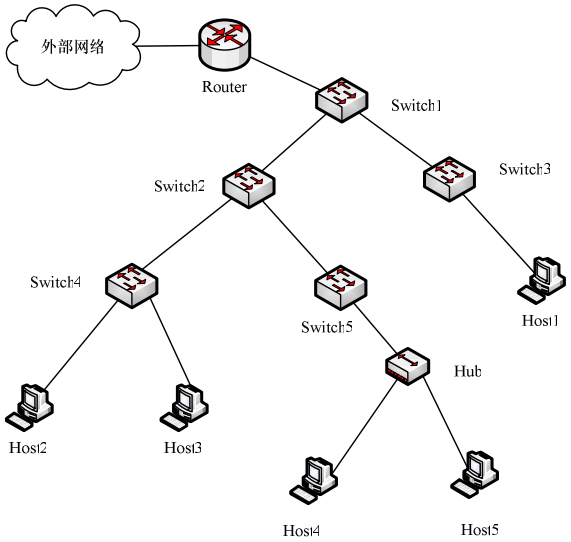

图 1 常见的网络拓扑结构

基金项目：国家自然科学基金资助项目(60872022)；广西研究生创新基金资助项目(2010105950812M21)

作者简介：潘 楠(1985－)，男，硕士研究生，主研方向：网络安全；王 勇，教授；陶晓玲，工程师

收稿日期：2011-05-19 **E-mail：**pn8527@yahoo.com.cn

2.1 基本概念

定义 1 标志节点即出口路由器。

定义 2 S_{ij} 表示第 i 台交换机 S_i 的第 j 个端口。

定义 3 A_{ij} 表示 S_{ij} 包含的 MAC 地址的集合；A_i 表示交换机 S_i 的所有下行端口包含的 MAC 地址的集合。

定义 4 若交换机 S_i 的 A_i 中包含除 S_i 本身以外的所有其他交换机的 MAC 地址，则该交换机为根交换机。

定义 5 交换机端口所对应的地址转发表中出现标志节点 MAC 地址的端口为上行端口，否则，为下行端口。

定义 6 交换机 S_i 的 A_{ij} 中未出现其他交换机 MAC 地址的端口为叶端口，否则，为非叶端口。

2.2 相关定理

定理 1 若交换机 S_i 的 A_{ij} 中包含交换机 S_k 的 MAC 地址 S_k_MAC 和 A_k，则可确定 S_{ij} 与 S_{kl} 有连接关系。

证明：因为交换机 S_i 的 A_{ij} 中包含 S_k_MAC 和 A_k，说明从 S_{ij} 出发必存在一条路径可以到达 S_k，所以 S_{ij} 与 S_{kl} 有连接关系(直接相连或间接相连)。

定理 2 当 S_{ij} 与 S_{kl} 之间存在连接关系时，若 A_{ij} 中 MAC 地址的个数($|A_{ij}|$)等于 A_k 中 MAC 地址个数($|A_k|$)加 1，即 $|A_{ij}|=|A_k|+1$，则 S_{ij} 与 S_{kl} 直接连接。

证明：采用反证法证明。假设定理 2 不成立，即在 S_{ij} 与 S_{kl} 之间存在连接关系且 $|A_{ij}|=|A_k|+1$ 时，S_{ij} 与 S_{kl} 为间接连接。由此可知，A_{ij} 中除了包含 S_k_MAC 和 A_k 之外，还包含其他交换机的 MAC 地址，因此，有 $|A_{ij}|>|A_k|+1$，与假设矛盾，定理 2 得证。

定理 3 如果交换机 S_i 的叶端口 S_{ij} 的 A_{ij} 中仅包含一台主机的 MAC 地址，那么该主机与交换机的 S_{ij} 端口直接相连，若包含多台主机的 MAC 地址，则这些主机通过 Hub 与交换机的 S_{ij} 端口相连。

定理 4 如果交换机 S_i 的叶端口 S_{ij} 的 A_{ij} 中仅包含路由器 R 的 MAC 地址，那么 R 与交换机 S_i 的 S_{ij} 端口直接相连。

3 改进的链路层拓扑发现算法

根据链路层网络的拓扑结构，将拓扑发现分为以下 3 个步骤：

(1)获得活动设备集合

由子网对应的出口路由器的 IP 地址和子网掩码可确定子网的 IP 探测范围，向该范围内的每一个 IP 地址发送 ICMP 回显请求报文，根据是否返回 ICMP 回显响应报文判断设备的活动状态。

(2)确定设备类型

向每个活动设备的指定端口 161 号发送 SNMP 报文，若有回应，则确定是交换机，否则，认为是主机。

(3)确定设备之间的连接关系

网络中主要存在的连接关系包括交换机与交换机的连接、交换机与主机的连接以及交换机与出口路由器的连接。根据定理 1~定理 4 可分别判断以上连接关系，最终得到完整的网络拓扑结构。

在发现设备之间连接关系的操作中，确定交换机与交换机之间的连接关系最复杂也最耗时。传统的确定交换机与交换机之间连接关系的算法如下：

```
for(每个交换机 Si){
for(交换机 Si 的第 j 个端口){
if(Si 已经与别的交换机端口相连)
    continue;
else{
if(Aij=Ak ∪ Sk_MAC)
    Sij 与 Sk 的上行端口直接相连;
}}
}
```

对于上述算法中的判定条件 $A_{ij}=A_k \cup S_k\_MAC$，需要将 A_{ij} 与网络中所有的 $A_k \cup S_k\_MAC(k\neq i)$ 进行判断操作，而 A_k 的计算量很大，导致发现效率不高，根据定理 1 和定理 2，可对上述判定条件作出修改，修改后的拓扑发现算法如下：

```
for(每个交换机 Si){
for(交换机 Si 的第 j 个端口){
if(Si 已经与别的交换机端口相连)
      continue;
else{
  if(Aij 中能查找到 Sk_MAC){
    if(Aij 中能查找到 Ak){
      if(|Aij|=|Ak|+1)
          Sij 与 Sk 的上行端口直接相连;
}}
}}
}
```

上述算法相对于传统算法的优势在于：通过修改交换机与交换机连接关系的判定条件，减少了 A_k 的计算量，缩短了确定连接关系的时间，提高了发现效率。

算法的关键技术包括：

(1)采用线程池技术提高发现效率

本文采用线程池的方式实现算法的前 2 步。与一般的多线程技术相比，线程池适用于单个任务处理时间较短、所需处理任务数量较多的情况，可减少频繁创建和销毁线程花费的时间，节省系统资源开销。采用线程池获得活动设备集合的核心代码如下：

```
ExecutorService pool = Executors.newScheduledThreadPool(50);
ArrayList<Future<String>> futureList =
                                new ArrayList<Future<String>>();
for(int i=0; i<ipSpaceList.size(); i++){
futureList.add(pool.submit(new
PingActiveDeviceCallable(ipSpaceList.get(i)))); }
for(Future<String> fs : futureList){
try {
if(fs.get()!=null)
  activeDeviceList.add(fs.get().toString());
} catch (InterruptedException e) {
e.printStackTrace();
} catch (ExecutionException e) {
e.printStackTrace();
}
}
pool.shutdown();
```

(2)采用哈希查找提高发现效率

在交换机连接关系的判断中，判断交换机端口的 MAC 地址集合中是否包含某一给定 MAC 这一操作的频率很高，但此操作一般需要通过遍历匹配实现，计算量非常大，尤其是在网络中交换机数目较多时，效率很低。因此，本文引入哈希表，将 A_{ij} 存为哈希表，采用哈希查找的方式确定 MAC

地址的包含关系，判断交换机之间是否连接。其中，哈希函数采用折叠法加除留余数法构造：首先把 MAC 地址分为 6 段，把每一段转换为十进制数后再顺叠相加，通过对相加的结果与 p(p 为一个素数，且小于或等于哈希表的大小)进行模运算得到哈希地址。此方法计算简单，得出的哈希地址可使 MAC 地址比较均匀地分布在哈希表中，从而减少查找的次数。当然，任何哈希函数的选择都有可能出现冲突，本文选择拉链法处理冲突。

4　核心算法实现

通过采用线程池技术可快速得到网络中的交换机集合 *switch* 和主机集合 *host*；确定 *switch* 中的根交换机，并设置第 1 层标识 *level*=1，下一层标识 *nextlevel*=*level*+1。在此基础上判定交换机与交换机、主机、路由器的连接关系，从根交换机开始逐层向下判断，类似于广度优先搜索算法，直到所有交换机都已判断，算法结束。基本流程如下：

(1)对交换机 S_i，从 *switch* 中将其删除。

(2)对 S_i 的一个活动端口 S_{ij}，将 A_{ij} 存为哈希表 *hashMac*；根据定义判断 S_{ij} 的类型，若为上行端口，执行步骤(3)；若为下行端口，执行步骤(4)。

(3)判断 *hashMac* 中 MAC 个数是否为 1，若是，则根据定理 4 确定 S_{ij} 与路由器相连；若不是，说明 S_{ij} 为 S_i 的上行端口。

(4)根据定义 6 判断 S_{ij}，若其为叶端口，判断 *hashMac* 中 MAC 个数是否为 1，若是，则根据定理 3 确定 S_{ij} 与一台主机相连；若不是，则由定理 3 确定 S_{ij} 通过 Hub 与多台主机相连；否则，执行步骤(5)。

(5)从 *switch* 中取出交换机 S_k。在 *hashMac* 中查找 S_k_*MAC*，若能查到，再在 *hashMac* 中查找 A_k，若能查到，则根据定理 1 确定 S_{ij} 与 S_{kl} 之间存在连接关系；然后判断 *hashMac* 中 MAC 个数是否等于 A_k 的 MAC 个数再加 1，若相等，则根据定理 2 确定 S_{ij} 与 S_k 的上行端口 S_{kl} 直接相连，将 S_k_*MAC* 存入第 *nextlevel* 层交换机的集合中；如果查不到，则判定 *switch* 中是否还有未判断的交换机，若有，重复此步骤，若无，执行步骤(6)。

(6)若 S_i 还有未确定的端口，返回步骤(2)；否则，判断第 *level* 层交换机的集合是否非空，若是，取未处理的交换机，并从集合中删除，返回步骤(1)；否则，执行步骤(7)。

(7)*level*=*nextlevel*。判断第 *level* 层交换机的集合是否非空，若是，*nextlevel* 加 1，取未处理的交换机，并从该层交换机集合中删除，返回步骤(1)；否则，算法结束。

5　测试结果与分析

5.1　实验环境

在搭建的实验环境中，有 1 台路由器、3 台交换机、1 台 Hub 和 5 台主机。在测试本文的算法前，需要确保 3 台交换机开启 SNMP 协议。

5.2　实验结果分析

采用本文的改进算法对实验网络进行发现，结果如图 2 所示。图 2 的左侧以树形结构显示了发现网络中的所有交换机和主机的 IP 地址；右下方显示了设备的详细信息；右上方显示了子网拓扑图，借助于不同的网络设备图标，可以直观地反映设备类型。当鼠标移至网络设备或链路时，会显示出相应信息，方便查看。图中虚线表示多台主机通过 Hub 与交换机相连。

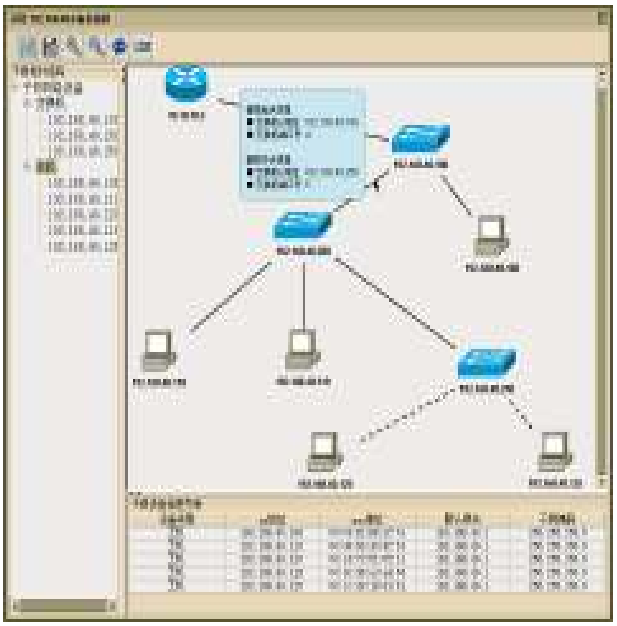

图 2　网络拓扑发现结果

实验网络是一个 C 类子网，而对于 C 类子网，最大存在的主机数为 254 台，采用本文算法创建一个大小为 50 的线程池对象，进行全网活动设备的探测仅需要约 7.2 s，若采用单线程方式，则需要约 123 s；对于确定交换机之间的连接关系这一操作，若采用改进前的算法耗时约 3.9 s，而采用本文的发现算法仅需要约 1.2 s，可见，本文算法的效率更高。

6　结束语

链路层的拓扑能真实地反映子网内部各个设备之间的连接情况。本文为快速实现链路层拓扑的快速发现，提出了一种新的基于 SNMP 的拓扑发现算法，该算法通过修改交换机连接关系的判定条件，并采用线程池和哈希查找等技术，提高了拓扑发现的效率，同时能处理网络中存在的 Hub。本文提出的算法能较好地解决单子网的拓扑发现问题，但无法发现存在 VLAN 的拓扑结构，这将是下一步的研究方向。

参考文献

[1] 邓泽林, 傅　明, 刘翌南. 一种新的异构网络链路层拓扑发现算法[J]. 计算机工程, 2010, 36(2): 119-120.

[2] 孔令文, 谭景信. 数据链路层网络拓扑发现算法研究[J]. 计算机工程与设计, 2008, 29(19): 4941-4944.

[3] 孙　娟. 基于地址过滤的网络拓扑发现算法[J]. 计算机工程, 2010, 36(7): 96-98.

[4] Breitbart Y, Garofalakis M, Martin C, et al. Topology Discovery in Heterogeneous IP Networks[C]//Proc. of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies. [S. l.]: IEEE Press, 2000: 265-274.

[5] Breitbart Y, Garofalakis M, Jai B, et al. Topology Discovery in Heterogeneous IP Networks: The NetInventory System[J]. IEEE/ACM Transactions on Networking, 2004, 12(3): 401-414.

[6] Lowekamp B, O'Hallaron D R, Gross T R. Topology Discovery for Large Ethernet Networks[C]//Proc. of SIGCOMM'01. San Diego, USA: [s. n.], 2001: 237-248.

[7] 郑　海, 张国清. 物理网络拓扑发现算法的研究[J]. 计算机研究与发展, 2002, 39(3): 264-268.

编辑　张　帆