

网络拓扑发现新算法及其实现

陈 福^{1,2}, 杨家海^{2,3}, 杨 扬¹

(1. 北京科技大学计算机科学与技术系, 北京 100083; 2. 清华大学信息网络工程研究中心, 北京 100084;
3. 清华大学信息与科学国家实验室, 北京 100084)

摘 要: 本文首先提出一种面向 IP 主干网的启发式网络拓扑发现算法. 该算法基于启发式、贪心法的思想, 具有 IP 地址的自动推理和查找范围的自我调整, 因而具备较强的自适应性. 进而提出了一个数据链路层拓扑发现算法. 该算法与已有的算法相比重点解决了无生成树协议工作下数据链路层拓扑发现、主机的拓扑构造以及集线器或无 IP 地址的交换机等所谓的哑设备发现问题. 同时阐述了一种新的数据结构类型并给出其数学定义、专有操作及该数据结构在算法中的应用. 算法已经实现并应用到我们自己开发的软件中, 同时在清华大学网络研究中心实验网络环境中进行了实际测试和验证, 达到了很好的效果.

关键词: 拓扑发现; 树型图; 网络层; 数据链路层

中图分类号: TP301 文献标识码: A 文章编号: 0372-2112 (2008) 08-1620-06

New Algorithms on IP Network Topology Discovery and Its Implement

CHEN Fu^{1,2}, YANG Jia hai^{2,3}, YANG Yang¹

(1. Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing 100083, China;
2. The Network Research Center of Tsinghua University, Tsinghua University, Beijing 100084, China;
3. Tsinghua University National Laboratory of Information Science and Technology, Beijing 10008, China)

Abstract: In this paper, we present two novel algorithms, one for discovery logical topology, the other one for discovering physical topology in heterogeneous IP networks. The algorithm for logical network, which adopt heurism \ greedy algorithms, can be applied to large scale IP network. For the algorithm can self deduce IP addresses, adjust the search domain, it has strong adaptabilities. Another novel algorithm is presented for physical topology which includes switches, hosts and hubs. At the same time, a new data structure named Treelike- Graphics for topology discovery is presented with precise mathematic definition, relationship set and proprietary operation on this data structure. The algorithm, which we present in this paper, does no base on the STP protocol to get the information of topology. So, there are few preconditions to the algorithm. We have implemented the algorithm presented in this paper in Perl in Tsinghua University own research network. The experimental results clearly validate our algorithm, demonstrating that our algorithm can consistently discover the accurate physical network topology with reasonably small running time requirements.

Key words: topology discovery; treelike graphics; network; data link

1 引言

网络管理中的配置管理主要功能为发现网络拓扑结构、管理网络设备. 而管理网络设备的基础是已知网络设备及其拓扑结构. 因此, 网络拓扑发现是网络管理的基础. 网络拓扑发现目的在于获取网络节点的存在性信息及其之间的拓扑关系, 并籍此绘制拓扑结构. 网管人员通过拓扑结构定位网络故障、发现网络瓶颈, 从而优化网络运行. 对于小型网络, 网管人员可以手工管理网络拓扑. 但大型网络设备之间的拓扑关系异常复杂, 无法手工管理网络拓扑. 自动网络拓扑发现是指计算机程序自动发现网络逻辑或物理拓扑、网络节点间拓扑、探测目标网络、发现网元设备, 并绘出网络拓扑图. IP 网络逻辑层包括路由器、子网拓扑, 物理拓扑主要针对

交换机、集线器、主机工作站等. 目前对 IP 网络层拓扑发现研究较多, 该领域算法相对比较成熟^[1~3]. 但这些算法均基于路由转发表, 而转发表数据量极大, 据此发现效率很低. 与网络层拓扑发现研究相比, 针对数据链路层拓扑发现研究相对较少. 由于网络层拓扑信息不能反映子网内的网络单元物理连接关系从而使网管人员对物理拓扑信息知之甚少, 这极大削弱了网管人员查找、排除网络故障的能力. 因此对物理拓扑发现的研究具有重要意义.

本文提出的 IP 网络交换层拓扑发现算法基于启发式和贪心法的思想, 通过源点逐步向外围拓展, 并在条件具备时通过 BGP 和 ARP 协议进一步提高发现效率, 具有很强的自适应性. 本算法突破了基于路由表拓扑发现耗时、低效的缺点, 极大提高了发现效率.

收稿日期: 2007-06-13; 修回日期: 2008-03-11

基金项目: 国家自然科学基金 (No. 60673160, No. 90412012, No. 60473083); 国家 863 高技术研究发展计划 (No. 2007AA01Z234, No. 2005AA1031102)

目前物理拓扑发现集中于具有生成树协议(STP)工作的交换机之间的连接关系. 文献[4]使用 SNMP 公有 MIB 变量在单子网、多子网和存在 VLAN 的情况下的链路层拓扑发现问题. 文献[5]采用基本相同的方法发现单子网、多子网、主机和路由器的连接关系, 但增加了对网络服务拓扑的查询. 2000 年 IETF 曾提出针对物理拓扑的 RFC, 定义了拓扑信息的 MIB 变量^[6]. 文献[7, 8]提出的针对链路层的算法都是基于 STP 协议、地址信息完全情况下的解决方法. 文献[9]将拓扑发现转换为数学问题并利用数学工具发现, 解决了多子网转发表转发信息不完整情况下的拓扑发现问题. 根据 STP 协议确定交换机之间、交换机与主机之间以及交换机、集线器、主机之间的连接关系, 在很多文献中均有阐述. 而实际网络环境中未启动 STP 协议的情况非常普遍, 因此仅根据 STP 协议进行拓扑发现必然不是完整的链路层拓扑图. 由于这些设备大多与终端用户非常接近, 从而对用户网络服务质量具有重要影响. 本文提出的算法重点解决无 STP 协议工作的情况下数据链路层拓扑发现、主机拓扑构造以及集线器或没有 IP 地址的交换机等所谓的哑设备发现问题. 该算法填补了无 STP 协议工作的数据链路层拓扑发现的空白, 是非常有意义的工作.

2 基于地址猜测的启发式交换层拓扑发现算法

2.1 算法的基础理论

引理 1 路由器的标识确定: 由于路由器具有多个接口, 必须唯一标识路由器. 如图 1(a) 所示, 通过路由器 A、B 均可以发现路由器 C. 但却是 C 的两个不同接口. 因此需要将 C1、C2 两个接口合并标识为同一设备. 本文首先使用路由器的 Loopback 地址, 其次采用所有接口中数值最大的接口作为路由器的标识接口地址.

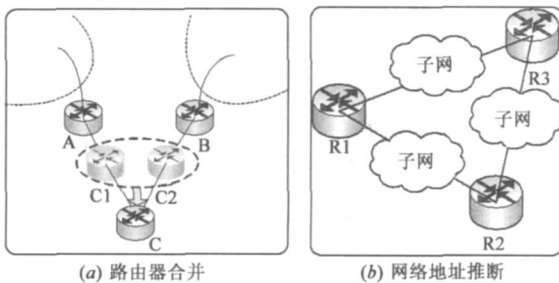


图 1 多址路由器合并与网络地址推断

引理 2 子网地址猜测: 网络层拓扑发现算法目的在于发现路由器及其子网的拓扑连接. 图 1(b) 中 R1、R2、R3 为路由器, 由图 1(b) 可以看出, 每个路由器可能带若干子网, 路由器之间相互连接. 但直接相连的两台路由器之间客观上也形成了只有两个 IP 地址的子网.

假设以 R1 作为种子 IP 开始拓扑发现, 可从 R1 的管理信息库(MIB)中取出每个交换机的接口地址和子

网掩码, 并据此计算出各个子网的子网地址. 设从 MIB 中取出的路由器的接口地址集合为

$$IP_SET = \{x | x \in \text{路由器的接口地址}\}$$

从 MIB 中取出的子网掩码地址集合为:

$$Mask_SET = \{y | y \in \text{路由器的子网掩码}\}$$

由于一个接口地址对应一个子网掩码, 而且接口地址与其对应的子网掩码两者逻辑相与可以得到网络号. 即, $\forall X_i \in IP_SET$ 可以唯一确定一个 $Y_i \in Mask_SET$, 即 $X_i | \rightarrow Y_i$.

令: $Z_i = X_i \& Y_i$, 则有 Z_i 构成一新的集合, 为子网地址集合, 设该集合为 Sub_Net.

根据 IP 地址规范, 可以从子网掩码推测可能的 IP 地址. 综上, 得出计算可能 IP 地址的方法:

(1) $\forall X_i \in IP_SET$ 唯一确定 $Y_i \in Mask_SET$, 即 $X_i | \rightarrow Y_i$.

(2) 将两者相与, 即 $Z_i = X_i \& Y_i$.

(3) 根据子网掩码得到非 0 部分的可能值与 Z_i 相加. 就可以得到一可能 IP 地址集合, 设该集合为 Gusee_IP_Set.

引理 3 BGP 协议端点路由器的确定: 如启动了 BGP 协议, 由于基于该协议通信的设备必然是路由器. 本算法突出之处其一就是如路由器启动了 BGP 协议则通过该协议取得和路由器直接交互的端到端结点信息. 通过该信息可直接确定设备的类型、连接关系.

引理 4 子网范围的自适应调整: 如两个路由器直接相连接, 子网掩码范围一般为 255.255.255.252, 但由于各种原因可能将其配置为 248、240、224、192 或 128. 本算法具有一定自适应性, 如果为 252 未能发现路由器设备, 程序将自动调整为 248、240、224、192 或 128.

2.2 网络层拓扑发现算法描述

```
TopoDisver(IP_Add IP, char * Community, int depth) {
    judgeDeviceType(gatewayIP, gateCommunity, );
    if(IP! = 路由器) {
        Return
    }
}
```

将种子 IP 放入发现队列;

判断 COMMUNITY;

While(发现深度小于规定深度 or 发现队列! = null) {

从发现队列中取得待发现地址;

将该设备所有接口地址放入到接口集合中;

取得待发现路由器端口地址、子网掩码;

SubAdd= gatewayIP \cap MaskAdd;

推断可能 IP 地址;

If(启动 BGP 协议){

通过 BGP 获得该路由器连接端点集;

}

```
// 判断该地址对应设备的存在状态和设备类型.
If(路由器) {
    判断该路由器是否已经发现;
    If(该路由器没有被发现){
        将该地址存入到待发现队列;
        将待发现地址和设备作为一个连接放入结果集;
    }
}
}
}
返回发现结果集;
```

3 数据链路层拓扑发现新算法基础理论

3.1 相关概念

根据交换机工作原理和 STP 协议, 并充分比较已有方法基础上, 本文提出无 STP 协议工作数据链路层拓扑发现算法, 借鉴 STP 协议思想, 根据各个交换设备到网关的距离判断子网内各个交换机的拓扑连接关系. 为便于阐述给出下述定义:

- 根交换机: 直接与网关相连接的交换机.
- 交换域: 与一个网关相连接的主机、集线器、交换机等组成的网络设备.
- 交换机上行口: 与网关连接距离最近的端口.
- 交换机下行口: 除了上行端口外, 与下级的级联交换机、集线器或主机相连接的端口.
- 下行完全集: 某一交换机以下所有交换机构成的集合.

- 上行完全集: 在上行交换域中交换机的集合.
- 下行交换域: 由某交换机及其以下各个设备组成的数据链路层网络.

- 上行交换域: 在整个交换域中除了本交换机和本交换机下行交换域之外的交换域.
- 上行交换机: 通过本交换机的上行端口与本交换机直接相连的交换机.
- 下行交换机: 通过本交换机的下行端口与本交换机直接相连的交换机.

3.2 前提假设

可假设一个交换机下行完全集是以树型结构组织的. 依据: 一个交换域交换机之间的连接关系只可能是树或图型结构, 如是图型关系则在此交换域中必启动了 STP 协议, 否则将形成回路. 而若启动 STP 协议, 其拓扑可由 STP 协议得到, 具体方法参见文献[4~9]. 本算法是在无 STP 协议工作情况下获取网络拓扑的方法. 因此, 本算法假设交换机之间的连接关系是树型结构.

3.3 无 STP 协议工作的链路层拓扑发现引理

引理 1. 无 STP 协议工作的链路层以下交换机上

行端口、下行端口的确定: 在无 STP 协议工作的链路层交换机转发表中目的地址为网关的表项所对应的端口即为该交换机的上行端口. 该交换机其余端口为下行接口, 该结论不言自明, 毋用证明.

引理 2 交换域根交换机的确定: 根据前述假设, 交换机下行完全集为树型结构. 在该树型结构中, 任一交换机通过其上行端口转发的转发表数目一定大于其下行交换机上行端口转发的转发表数目. 且在该交换域中的根交换机一定是通过本交换机上行端口转发的转发条目最少的那个交换机.

证明:

如图 2 所示, 设在网关的下行交换域中存在任意两个交换机, 其中交换机 S_1 为交换机 S_2 的上行交换机, S_2 为交换机 S_1 的下行交换机. 由图 2 可看出, 交换机 S_1 的交换域 = $S_{11} + S_{12} + S_{13}$, 其中 S_{13} 为 S_1 的上行完全集为, $S_{11} + S_{12}$ 为 S_1 的下行完全集. S_2 的交换域 = $S_{21} + S_{22}$. 其中 S_{21} 为 S_2 上行完全集, S_{22} 为 S_2 的下行完全集.

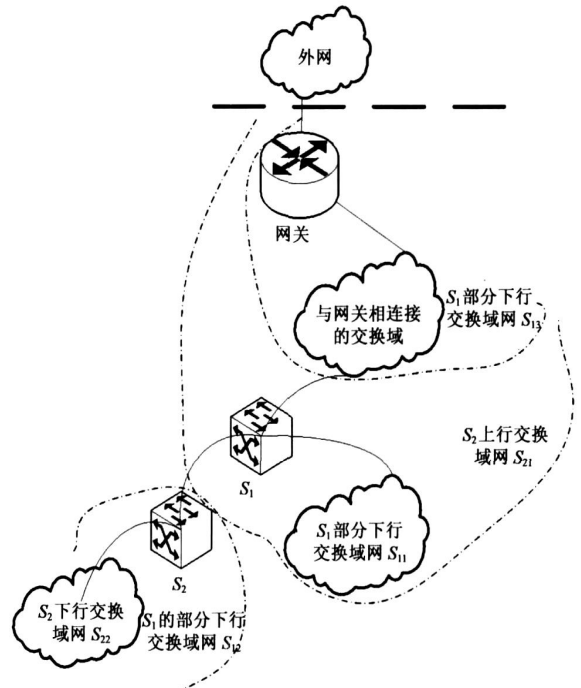


图 2 网关交换域

$$S_{22} + S_{21} + S_{22} = S_{11} + S_{111} + S_{112} + S_{13} \quad (1)$$

其中, S_{22} 和 S_{12} 只相差交换机 S_2 本身, 因此可以认为 $S_{22} = S_{12}$. 同时我们也忽略交换机 S_1 、 S_2 . 因此由式 1 得到:

$$S_{21} = S_{11} + S_{13} \quad (2)$$

因此, 可以推导出:

$$S_{21} \geq S_{13}$$

引理 3 交换域中的交换机拓扑生成树的确定: 任一交换机下行完全集与该交换机下行完全集中所有交

交换机下行完全集之和的差就是该交换机的下行交换机集合. 如果已知根交换机, 则可以通过根交换机的转发表求得该交换机域的下行完全集, 进而求得该交换机下行完全集中所有交换机下行完全集, 因此可以得到根交换机的下行交换机, 通过递归使用本结论即可得到该交换域的交换机拓扑信息.

图 3 是实际网络的模型, 交换机 S_1 的下行完全集为 S_{11} , 交换机 S_1 的下行交换机集合为 $\{S_2, S_3, S_4, \dots\}$. 交换机 S_2 的下行完全集为 S_{21} , S_3 的下行完全集为 S_{31} , S_4 的下行完全集为 S_{41} , \dots, S_n 的下行完全集为 S_{n1} . 我们可以看出交换机 S_1 的下行完全集 $S_{11} = S_2 + S_3 + S_4 + \dots + S_n + S_{21} + S_{31} + S_{41} + \dots + S_{n1}$.

因此交换机 S_1 的下行交换机集合 $S_2 + S_3 + S_4 +$

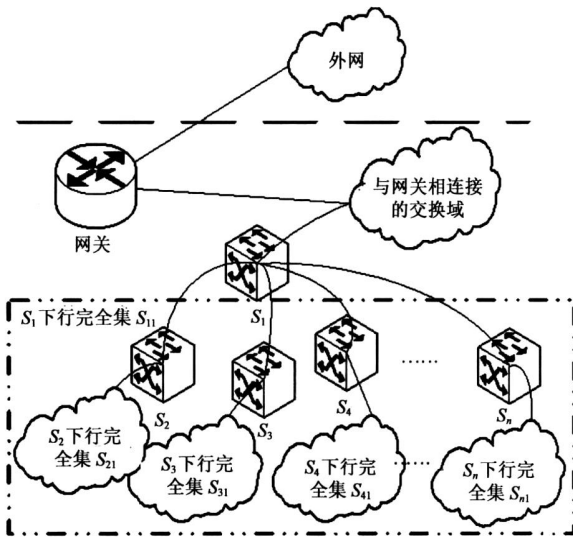


图 3 交换机拓扑生成图

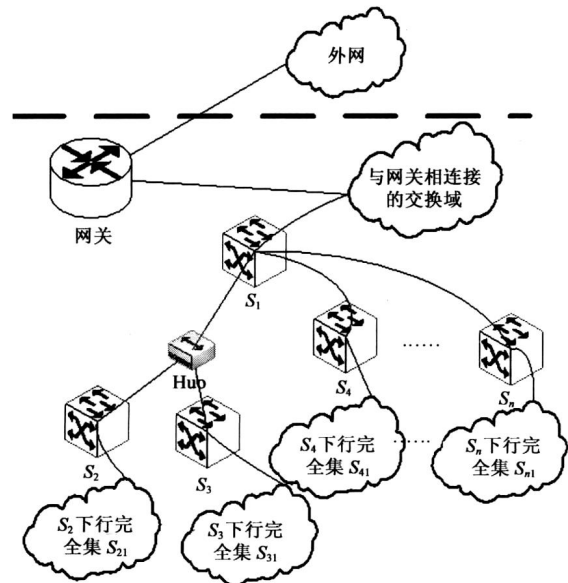


图 4 哑设备发现示意图

$$\dots + S_n = S_{11} - \{S_{21} + S_{31} + S_{41} + \dots + S_{n1}\}$$

引理 4 集线器或没有 IP 地址交换机的发现: 通过结论 2 能确定若干交换机处于交换机拓扑生成树的同一层次而且具有相同的上行交换机, 则在这些交换机之上必然存在一个集线器或没有 IP 地址交换机设备.

证明:

应用公理 2, 可以从图 4 中得到和图 3 完全相同的交换机拓扑生成树, 并且也可以知道交换机 S_1 的下行交换机的转发端口. 因此由图 4 可以明显看出若干交换机如果通过集线器接入到上行交换机, 则在其上行交换机上必然具有相同转发端口.

4 一种新型数据结构—树型图的提出

4.1 树型图逻辑特点及定义

在进行拓扑发现过程中, 我们发现在交换机交换域的下行端口转发信息中存在一个类似与树、但实质却是图的一种数据结构, 类似于树的依据:

(1) 本算法解决在没有 STP 协议工作的情况下二层拓扑发现问题. 因此该拓扑一定是一树型结构, 因为如果不是树型结构则必然存在回路, 如果存在回路又没有 STP 协议消除冗余路线则必然造成转发死循环, 因此实际物理连接一定是树结构.

(2) 在交换域各个交换机的下行端口转发表中一定不含其兄弟和祖先结点的转发信息. 因为这样的转发信息一定通过上行端口转发. 而且在该生成树深度相同的各个结点的交换机的向下转发表一定不包含其兄弟节点的信息从而具有树型结构的典型特点.

(3) 无论是真实的物理连接还是下行端口转发表均存在唯一的根结点.

因为交换域中任一交换机的下行端口集合必然包含所有下行交换域交换机的物理地址. 因此我们说具有图的特点. 简言之, 该数据结构是具有树的特点的图. 下面给出树型图数学定义.

树型图(Treelike_Graphics) 是一个包含 n 个结点的有限集, 在任意一棵非空树型图中:

- (1) 有且仅有一个特定的称为根的结点(Root).
- (2) 当 $n > 1$ 时, 其余结点可分为 m ($m > 0$) 个互不相交的有限集 T_1, T_2, \dots, T_m , 其中每一个集合本身又符合本定义, 并且称为根的子树型图(Sub Treelike_Graphics).
- (3) 任意一个结点都有指向其任意一个子孙的信息. 这此一点与普通的树型结构截然不同.

ADT Treelike_Graphics {

数据对象 D : D 是具有相同特性的数据元素的集和, 称为顶点集.

数据关系 R :

若 D 为空集, 则称为空树型图;

若 D 仅含一个数据元素, 则 R 为空集, 否则 $R = \{H\}$, H 是如下二元关系:

(1) 在 D 中存在唯一的称为根的数据元素 $root$, 它在关系 H 下无前驱;

(2) 若 $D - \{root\} \neq \Phi$, 则存在 $D - \{root\}$ 的一个划分 $D_1, D_2, \dots, D_m (m > 0)$, 对任意 $j \neq k (1 \leq j, k \leq m)$ 有 $D_j \cap D_k = \Phi$, 且对任意的 $i (1 \leq i \leq m)$, 唯一存在数据元素 $x_i \in D_i$, 有 $\langle root, x_i \rangle \in H$;

(3) 对应于 $D - \{root\}$ 的划分, $H - \{\langle root, x_1 \rangle, \dots, \langle root, x_m \rangle\}$ 有唯一的一个划分 $H_1, H_2, \dots, H_m (m > 0)$, 对任意 $j \neq k (1 \leq j, k \leq m)$ 有 $H_j \cap H_k = \Phi$ 且对任意的 $i (1 \leq i \leq m)$, H_i 是 D_i 上的二元关系, $(D_i, \{H_i\})$ 是一棵符合本定义的树型图, 称为根 $root$ 的子树型图.

(4) $\forall v \in D, v$ 的子孙集合 $W, \forall w \in W$, 则必然存在关系 $VR = \{\langle v, w \rangle \mid v \in D, w \in W \text{ 且 } P(v, w), \langle v, w \rangle \text{ 表示从 } V \text{ 到 } W \text{ 的弧, 谓词 } P(v, w) \text{ 定义了弧 } \langle v, w \rangle \text{ 的意义或信息}\}$.

基本操作 P :

对树和图的基本操作均可移植到对树型图的操作, 但本文重点阐述对我们有意义的操作—将该数据结构转换为树结构 $TGtoTree(TG, &T)$. 该操作实质上就完成了拓扑发现的工作, 即无生成树协议工作情况下数据链路层拓扑信息的获得.

}

4.2 树型图基本操作 $TGtoTree(TG, &T)$ 算法描述.

$TGtoTree(TG, &T)$ { // 利用树型图生成对应的生成树

Queue TreeQueue; // 定义一个队列

InitQueue(& TreeQueue)

Queue TreeResultQueue; // 定义一个队列

InitQueue(& TreeResultQueue)

TGNodeLinkElem * P;

TGNode treeNode = TG . nodes[0];

P = treeNode. offspring;

While(P != NULL)

EnQueue(&TreeQueue, P);

P = P > next;

}

TreeResultQueue = TreeQueue;

TGNodeLinkElem * TGArc;

TGElemType offspring_ ID; // 子孙 ID

while(TGArc != NULL) {

DeQueue(&TreeQueue, & TGArc);

offspring_ ID = TGArc. offspring_ ID;

TGArc = TG . nodes[offspring_ ID];

If (TGArc 中的某一结点子孙域 = TreeResultQueue

的某一结点子孙域) {

从 TreeResultQueue 删除该节点;

将 TGArc 中的这一结点加入到 TreeResultQueue;

}

}

}

return TreeResultQueue;

}

5 无 STP 协议工作的数据链路层拓扑发现算法

DataLinkTopoDisver(IP_ Add gatewayIP, char * gateCommunity, IP_ Add MaskAdd) {

judgeDeviceType(gatewayIP, gateCommunity,); // 判断是否为路由设备

if(给出的 IP 设备不是路由器) {

返回

}

SubAdd = gatewayIP \cap MaskAdd; // 计算当前网关该端口的网络号

应用 ARP 协议取得当前网关交换域的活动 IP 地址集合 All_IP;

在 IP 地址集合 All_IP 中求得交换机地址集合 SwitchSet;

// 应用公理二在网关交换域中交换机地址集合 SwitchSet 中的根结点交换机

foreach Switch (SwitchSet) { // 通过 foreach 循环寻找通过上行端口转发的表项数目最少的交换机.

temp = getPortNum (Switch, SwitchCommunity);

if(tempMIN > temp) {

tempMIN = temp;

}

}

RootIP = tempSwitch; // 根结点交换机

得到树型图 TG;

利用树型图生成对应的生成树 TGtoTree(TG, &T);

}

6 实验验证与分析

作者已编程实现上述算法 1, 并将其发布为 Web Service. 通过服务发布的内容包括: (1) 发现的路由器集合. (2) 已发现的路由器名称、接口地址、子网掩码和网络号的集合. (3) 拓扑连接关系序偶的集合. 通过在下一代中国教育科研网 (CERNET2) 上测试, 算法完全达到了预期的功能. 发现的 CERNET2 拓扑连接设备与时间如图 5(a) 所示, 同时将该. 算法在中国高速互连研究试验网络 (NSFCNET) 和中国教育科研网 (CERNET) 上实

验, 均验证该算法的有效性和高效率。

由于无 STP 协议工作, 目标网络规模不应该太大, 否则就应该运行 STP 协议消除回路。因此作者在清华大学网络中心搭建了图 5(b) 所示的工作环境, 并采用 Perl 语言实现了原型系统, 通过实验证明该算法的有效性, 在极短的时间内发现了全部设备, 达到了预期的效果。

7 结束语

本文提出的 IP 网络交换层拓扑发现算法突破了基于路由表方法的耗时、低效的缺点, 通过推测、验证逐步从发现起点向外围拓展, 并在条件允许时根据 BGP 和 ARP 协议进一步提高发现效率, 与已有工作相比本算法具有 IP 地址自动推理和查找范围自我调整因而具有很强的自适应性。该算法具有较高效率, 实验结果和实际应用均表明该算法的高效性、实用性。

本文提出另一算法解决了 STP 协议工作数据链路层拓扑发现问题的理论基础并实现了算法。同时引出一种新型数据结构类型 Treelike_Graphics。本文从理论到实现从根本上解决了相应约束条件的拓扑发现问题。

参考文献:

- [1] Benoit Donnet, Philippe Raoult, Timur Friedman. Efficient Algorithms for large scale topology discovery [A]. Proc of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems[C]. ACM Press, 2005. 327- 338.
- [2] Daniel G. Waddington, Fangzhe Chang, et al. Topology discovery for public IPv6 Networks [A]. Proc Of ACM SIGCOMM Computer Communication Review[C]. ACM Press, 2005. 59-68.
- [3] Nick Feamster, Hari Balakrishnan. Towards a logic for wide area Internet routing [A]. Proc of the ACM SIGCOMM Workshop on Future Directions in Network Architecture[C]. ACM Press, 2003. 289- 300.
- [4] Yuri Breitbart, Minos Garofalakis, Ben Jai, Clifton, et al. Topology discovery in heterogeneous IP networks: the NetInventory system[J]. IEEE/ACM Transactions on Networking, 2004, 12(3): 401- 414.
- [5] Narayan P P S, Breitbart Y, Garofalakis M, et al. Physical and

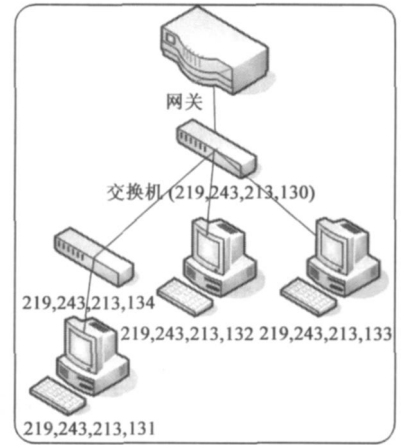
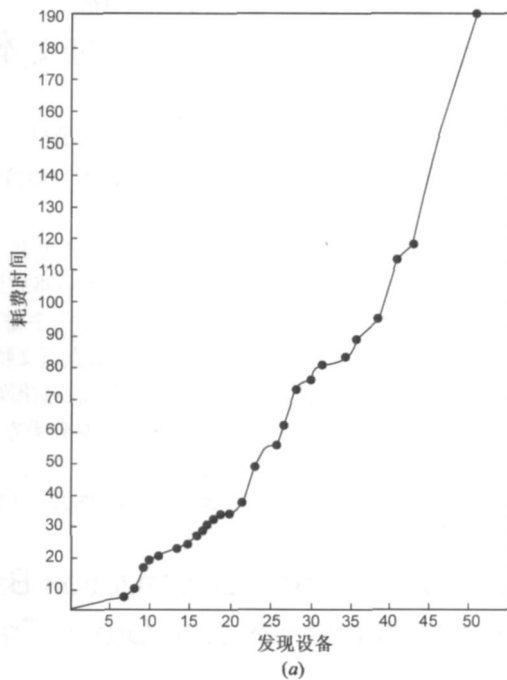


图 5 算法发现的设备与耗费的时间变化图

Service Topology Discovery in Heterogeneous Networks: the Net Inventory System [A]. Telecommunications Network Strategy and Planning Symposium [C]. Vienna: Hermann, 2004. 279 - 284.

- [6] Bierman A, Jones K. Physical Topology MIB. Internet RFC-2922 [S]. September 2000.
- [7] Yuri Breitbart, Minos Garofalakis, Cliff Martin, et al. Topology Discovery in Heterogeneous IP Networks [A]. Proceedings of INFOCOM 2000 [C]. Tel Aviv: IEEE INFOCOM, 2000. 265-274.
- [8] Yuri Breitbart, Minos Garofalakis, et al. Topology discovery in heterogeneous IP Networks: the Net inventory system [J]. IEEE/ACM Transactions on Networking, 2004, 12(3): 401-414.
- [9] Yantao Sun, Zhimei Wu, Zhiqiang Sh. The physical topology discovery for switched ethernet based on connections reasoning technique [J]. Communications and Information Technology, 2005 [C]. IEEE International Symposium 2005, 1(12-14): 44-47.

作者简介:



陈 福 男, 1973 年生于辽宁朝阳, 北京科技大学/清华大学联合培养博士, 研究生, 讲师。研究方向: 服务管理、网络管理、虚拟现实技术、嵌入式系统。

Email: cherfurun@gmail.com