

基于 SNMP 的链路层拓扑发现算法

薛珊珊, 李俊, 金涛

(南京航空航天大学信息科学与技术学院, 南京 210016)

摘要: 提出一种基于地址转发表的网络拓扑发现算法, 利用简单网络管理协议获得网桥 MIB 中的地址转发表信息, 从而推导出连接关系。与现有方法相比, 它不要求各个网桥转发表的信息是完备的, 也无须进行大量比较。仿真实验结果表明, 该算法能够准确地计算出整个被管网络的二层拓扑结构。

关键词: 拓扑发现; 简单网络管理协议; 地址转发表

Link Layer Topology Discovery Algorithm Based on SNMP

XUE Shan-shan, LI Jun, JIN Tao

(College of Information Science & Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 210016)

【Abstract】 A network topology algorithm based on Address Forwarding Table(AFT) is presented, which gets the AFT information from the bridge MIB by Simple Network Management Protocol(SNMP), and deduces the indirect connective relationship. Compared with current algorithms, it does not require the completeness of the AFT information and a mass of compare. Simulation experimental results show this algorithm can calculate topology graph of 2 layer accurately.

【Key words】 topology discovery; Simple Network Management Protocol(SNMP); Address Forwarding Table(AFT)

1 概述

拓扑发现是指发现网络节点, 并发现网络节点之间的连接关系。它是网络管理、通信瓶颈分析、网络故障分析定位的关键与前提。在认识到链路层拓扑发现的重要意义后, IEEE 于 2002 年起草的标准 IEEE 802.1AB 中定义了 LLDP(Link Layer Discovery Protocol)协议, 能够发现临近的设备, 并将这些信息保存在标准 MIB 中。通过简单网络管理协议(Simple Network Management Protocol, SNMP)读取该 MIB, 可以发现链路层的网络拓扑, 该标准目前仍处于草案阶段^[1]。

目前, 运用较多的拓扑发现算法主要有: (1)基于生成树协议的方法^[2], 利用生成树信息构造网桥之间连接关系。该方法的优点是: 不用发送大量的 ICMP, 可以发现备份链路; 缺点是: 不能发现交换机与主机之间的连接。(2)基于地址转发表的物理拓扑发现算法^[3-4], 该方法的优点是: 可发现网络层连接; 缺点是: 要求转发表或者下行端口转发表完整性。文献[4]选择在网络中注入附加的流量保证地址转发表的完整, 增加了网络负荷, 从而使算法的可操作性受到影响。

鉴于每个交换机都具备反向学习地址的能力, 从通用性考虑, 本文算法仍是以地址转发表中的信息为主要推导依据, 因为在不完整的地址转发表中仍存在着一些连接信息, 只要利用一定的推导算法也可以建立拓扑关系, 并针对校园网这个多种类设备构成的以太网环境, 实现基于地址转发表的快速网络拓扑发现算法, 同时加以验证。

2 算法基础

2.1 相关符号定义

在介绍拓扑发现算法前, 先定义算法所涉及的字符以及术语。

(1) V_k : 表示交换机 V 的第 k 个端口(V 交换机集合)。

(2) $F(V, k)$: 表示网桥 V 的第 k 个端口学习到的地址转发表条目集合(V 交换机集合)。

(3)四元谓词 $R\langle SwitchA, PortX, PortY, SwitchB \rangle$: 表示 A_x 和 B_y 之间有一条路径。若 $PortX$ 与 $PortY$ 都已知, 则称为四元关系, 否则称为类三元关系。

2.2 算法理论依据

根据生成树协议可知, 在局域网中, 任意一对网络节点之间有且只有 1 条包含 2 层网络设备的路径, 包括以下 2 种情况:

(1)直接路径: x, y 端口不经过其他交换机, 通过一条物理线路连接在一起;

(2)间接路径: x, y 端口之间存在 N 个交换机($N > 0$)。本算法的主要思想是通过地址转发表信息建立起间接连接关系, 再进一步生成直接连接关系。

文献[5]提出一种基本推理规则(Basic Reasoning Rule, BRR)。

(1)对称律: $Link(A, i, B, j) \Leftrightarrow Link(B, j, A, i)$;

(2)传递律 I: $Link(A, i, C, u) \wedge Link(C, v, B, i) (u \neq v) \Rightarrow Link(A, i, B, j)$;

(3)传递律 II: $Link(A, i, C, u) \wedge Link(B, j, C, v) \Rightarrow \exists x Link(A, i, B, x) \vee \exists y Link(A, y, B, j)$;

(4)互斥律: $Link(A, i, B, j) \Rightarrow \neg (\exists x \exists y (Link(A, x, B, y) \wedge (x \neq i \vee y \neq j)))$;

(5)互斥律推论: $Link(A, i, B, x) \wedge Link(A, y, B, j) \Rightarrow$

基金项目: 国家部委基础科研基金资助项目

作者简介: 薛珊珊(1984-), 女, 硕士研究生, 主研方向: 计算机网络安全; 李俊, 教授; 金涛, 硕士研究生

收稿日期: 2009-01-20 **E-mail:** nuaa_xss@126.com

$Link(A, i, B, j)$ 。

本文的推导算法基于 BRR 规则。在利用 SNMP 获取 MIB 中的信息后,可以根据下面的规则判定网络设备的连接关系。

规则 1 如果 $A \in F(B, i) \wedge B \in F(A, j)$, A, B 交换机集合, 则端口 A_i 和 B_j 之间存在一条间接路径, 且该路径是唯一的, 即四元关系 $R\langle A, i, j, B \rangle$ 成立。

证明: 利用反证法, 假设除 A_i 和 B_j 存在一条间接路径之外, A_x 和 $B_y (x \neq i \text{ 且 } y \neq j)$ 之间为一条不同的路径, 而这与“任意一对网络节点之间有且只有一条包含 2 层网络设备的路径”矛盾, 即本结论成立。

规则 2 设存在类三元关系: $R(A, i, C, x) \wedge R(B, j, C, x)$ 且 $A \in F(B, v)$ 或 $B \in F(A, k)$, 其中, i, j, v, k 是已知端口; x 是未知端口, 可将三元关系更新为四元关系: $R(A, i, B, j)$ 。

证明: 根据 BRR 传递律 I、II 和互斥律, 利用关系的交、并做如下推导:

$$R(A, i, C, x) \wedge R(B, j, C, x) \Rightarrow R(A, i, B, x) \vee R(B, j, A, x) \quad (1)$$

$$\text{式}(1) \wedge R(A, i, B, y) \Rightarrow R(A, i, B, j) \quad // \text{互斥律}$$

证毕。

规则 3 假设存在以下类三元关系: $R(A, i, C, x) \wedge R(B, j, C, x) \wedge R(A, k, D, x) \wedge R(B, m, D, x)$ 且 $A \notin F(B, y) \wedge B \notin F(A, x)$, 其中, i, j 是已知端口; x, y 是未知端口。可将类三元关系更新为四元关系: $R(A, i, B, m) \vee R(A, k, B, j)$ 。

证明: 根据 BRR 传递律 II 和互斥律, 利用关系的交、并做如下推导:

$$R(A, i, C, x) \wedge R(B, j, C, x) \Rightarrow R(A, i, B, x) \vee R(B, j, A, x) \quad (2)$$

//传递律 II

$$R(A, k, D, x) \wedge R(B, m, D, x) \Rightarrow R(A, k, B, x) \vee R(B, m, A, y) \quad (3)$$

//传递律 II

$$\text{式}(2) \text{ 式}(3) \Rightarrow R(A, i, B, m) \vee R(A, k, B, j) \quad // \text{互斥律推论}$$

证毕。

3 算法实现

3.1 主要数据结构

在链路层拓扑发现中, 使用的关键数据结构如下:

```
class AFT //T 转发表信息对象
{int Vertex;
//学习到的 MAC 地址转换成对应的交换设备的编号
int flag; //节点连接标志
}
class Relation //连接对象
{ int SwitchA; //交换机 A
int PortX; //交换机 A 参与连接的端口 X
int PortY; //交换机 B 参与连接的端口 Y
int SwitchB; //交换机 B
}
```

为描述一致, 定义如下: 若 $PortX \neq 0 \wedge PortY \neq 0$, 则称为四元关系, 否则称为类三元关系。

```
AFT AFT[N][N]
//二维数组存储转发表信息, N 为交换设备总数
```

3.2 算法实现

为描述拓扑发现算法, 首先定义如下对象: RelationR3List 队列存放类三元关系; RelationList 队列存放间接四元关系; DirectRelationList 队列存放直接连接四元关系。

算法的核心由 2 个部分组成: 第 1 部分为预处理过程, 发现主机使用指定的 OID(1.3.6.1.2.1.17.1.0)以多线程异步测试待发现网络中的 IP 地址集合是否能够对 SNMP 操作响应。

第 2 部分为连接关系的推断判定, 包括判定交换机间接连接关系、判定直接连接关系、判定交换机与主机间关系。

3.2.1 确定网桥之间的连接

确定网桥之间的连接过程包括 7 个步骤。

(1)读取所有交换机的地址转发表, 填写 AFT 表。

(2)置 $i=0$ 。

(3)对所有的交换机 $j (0 < j < n, n$ 为交换设备总数), 若满足以下 2 个条件:

1) $AFT[j][i].Vertex \neq 0 \wedge AFT[i][j].Vertex \neq 0$;

2) 节点的连接标志位为 0 (记编号 i 的交换机为 A , 编号 j 的交换机为 B)。

根据判定规则 1, 生成四元关系 Relation 对象为

$$R\langle A, AFT[i][j].Vertex, AFT[j][i].Vertex, B \rangle$$

置节点连接标志位为 1。将新连接加入到关系队列 RelationList 中。

(4) $i=i+1$ 。

(5)若 $i < n$, 则转到(3), 否则转到(6)。

(6)两两取交换机推导连接关系, 以计算交换机 B, C 为例, 查找与 B, C 相关的设备即满足:

1) $AFT[B][x].Vertex = 0 \wedge AFT[C][x].Vertex = 0$;

2) 对应的节点连接标志位为 0, 其中 x 为异于 B, C 的交换设备。

根据更新规则 1, 新建类三元关系 Relation 对象为

$$R\langle B, AFT[B][x].Vertex, 0, C \rangle // \text{设为 } R1$$

$$R\langle B, 0, AFT[B][x].Vertex, C \rangle // \text{设为 } R2$$

将 $R1, R2$ 加入待推导类三元关系 RelationR3List 队列中。

(7)重复以上操作, 遍历完所有与 B, C 相关的交换设备, 调用处理类三元对象关系的过程:

```
disposal(B, C, RelationR3List)
```

处理类三元关系的过程 $disposal(B, C, RelationR3List)$ 可以分解成以下步骤:

(1)若 RelationR3List 不为空, 转(2), 否则转(5)结束。

(2)若满足以下 2 个条件之一:

1) $AFT[C][B].Vertex \neq 0$;

2) $AFT[B][C].Vertex \neq 0$ 。

则根据更新规则 1, 更新类三元关系为四元关系。置 A, B 连接标志为 1。判断连接是否已加入到关系队列 RelationList, 清空类三元队列 RelationR3List 转(5)结束, 否则转(3)。

(3)当 RelationR3List 长度小于 2, 清空类三元队列, 转(5)结束, 否则转(4)。

(4)根据更新规则 2, 利用 RelationR3List 中的类三元关系推导构造四元关系:

$$R1 \wedge R2 \wedge R3 \wedge R4 \wedge \dots \Rightarrow R$$

置 A, B 节点连接标志为 1, 完善 AFT 表。将四元关系 R 加入到队列 RelationList, 清空类三元队列, 转(5)结束。

(5)结束。

3.2.2 判定直接连接

通过上述判定, 将 AFT 表基本补充完全。根据地址转发表的生成原理: 如果从某端口收到一个数据帧, 以太网交换机就会分析该数据帧的源 MAC 地址, 并认为目的地址为 source 的报文可以由该端口转发。若 2 个交换设备的一对端口之间有直连关系, 那么这 2 个交换设备可以在对方的转发表中间互相学习到, 并且其转发表交集为空。

判定过程为: 使用上述结论判定 RelationList 队列中的直接连接的四元关系, 并将结果从 RelationList 队列中移出, 保存于 DirectRelationList 队列中。 (下转第 124 页)