基于谓词逻辑推理与生成树的优化链路层网络拓扑发现方法

张 良[1],郭延峰[2],何 华[3]

1. 中国航空工业集团公司沈阳发动机设计研究所,沈阳 110015
2. 辽宁工业大学电子与信息工程学院,辽宁锦州 121001
3. 北京启明星辰信息技术股份有限公司,北京 100193

摘要 物理拓扑结构发现的目标是确定网络中的各种设备以及这些设备物理端口之间的链路连接关系,这对于网络性能监测与评估、故障发现与定位、资源分配与管理等一系列维护工作具有重要意义。本文所提供的基于谓词逻辑推理和基于生成树协议的优化拓扑发现算法,在 AFT 数据不完全的情况下,能够有效计算出网络节点信息,从而克服了现有链路层网络拓扑发现方法的不足,提高了获得网络物理拓扑的可能性。

关键词 拓扑;生成树协议;谓词;地址转发表

中图分类号 TP393.02 **文献标识码** A **doi** 10.3981/j.issn.1000-7857.2011.18.007

Algorithm for Data Link Layer Topology Discovery Based on Prediction Logic and Generation Tree

ZHANG Liang[1], GUO Yanfeng[2], HE Hua[3]

1. *Shenyang Aero Engine Research Institute, Aviation Industry Corporation of China, Shenyang 110015, China*
2. *School of Electronics & Information Engineering, Liaoning University of Technology, Jinzhou 121001, Liaoning Province, China*
3. *Beijing Venustech Inc., Beijing 100193, China*

Abstract The objective of the physical topology discovery is to determine the connection relationship between various equipments in the network and the link between physical ports, which is very important for maintenance work, and with which the performance of the network can be monitored and evaluated efficiently and the fault discovery and positioning is also made easy. This paper proposes an optimization algorithm based on the predicate logic reasoning and the spanning tree protocol. It can effectively compute node information under the circumstances that the topology discovery AFT (Address Forwarding Table) data are incomplete. Thus the algorithm can overcome the shortcomings of the existing link layer network topology discovery, to obtain a possible physical topology network. This paper provides a new way to discovery link layer topology, with the link deducation technology, "to repair and revise" the uncomplete root AFT data. The shortcomings of the predicate logic and STP (Spanning Tree Protocol) algorithm are remedied, and the advantages are taken.

Keywords topology; tree protocol; predicate logic; AFT

0 引言

交换式以太网物理拓扑结构发现的目标是确定网络中的各种设备以及这些设备物理端口之间的链路连接关系。网络的物理拓扑信息对于网络性能监测与评估、故障发现与定位、资源分配与管理等一系列维护工作具有重要意义[1]。基于链路层的拓扑发现,目前也有不少算法[2-3],其基本思想是通过 SNMP 协议获取交换机的地址转发表 (Address Forwarding Table,AFT),然后采用直接连接定理,或者采用生成树协议推

收稿日期:2011-03-10;修回日期:2011-05-30

作者简介:张良,高级工程师,研究方向为计算机网络安全,电子信箱:dadi@yeah.net

算出网络物理拓扑连接情况[4]。

在链路层的物理网络拓扑发现技术中,基于连接推理技术的谓词逻辑推理算法对于网络节点信息的依赖性较高,只有在 AFT 数据较为完整的情况下,才能发挥较好的作用。如果 AFT 不完整,基于连接推理技术的谓词逻辑推理算法虽然不能获取全部节点的连接关系,但可以获得部分节点的连接关系,即可以"修补"AFT 数据。而生成树协议算法,优势较为明显,在 AFT 不太完整的情况下,依然可以获得网络拓扑结果,但它要求 AFT 符合一定的分布(逆根 AFT 数据要完整),才能发挥作用[5]。

为了解决上述技术问题,充分利用谓词逻辑推理算法[6]和生成树协议算法[7]各自的优势,克服这两种算法各自的缺点,本文提出了一种链路层网络拓扑发现方法及装置。

1　基于谓词逻辑的网络拓扑发现

首先介绍网络管理的基本概念和链路层拓扑发现算法中用到的术语和符号。

(1) 管理域(administrative domain)[8]。IP 子网定义为任意两个节点不经过路由器通信的最大的 IP 地址集合,即为一个广播域(broadcast domain)。

(2) 网络设备类型。主要可以分为 3 类:一类是交换机(switch),每个交换机维护一张地址转发表,可以利用该 SNMP 读取这张表;一类是主机(host),在上面没有地址转发表;还有一类是集线器(hub),这类设备没有 MAC 地址,在网络中无法直接发现。

(3) 地址转发表。每个交换机上都维护着一张 AFT 表,该表记录着接收到的数据包应该从哪个端口转发出去。地址转发表的记录格式可以简单地表示为由端口和 MAC 地址组成的二元组(port,mac),其中 port 称为转发端口,mac 称为转发地址。

管理域抽象为一个树,用 T 表示,则 T 中所有的网络设备集用 D 表示,主机集用 H 表示,交换机集用 S 表示,则 $D=H+S$。用 $S_i(S_i\in S)$表示交换机 i,用 $H_i(H_i\in H)$表示主机 i,用 D_i $(D_i\in D)$表示结点 i。用 port(S_i,j)表示交换机 S_i 的端口 j,简称 S_{ij}。一个交换机的地址转发表是二元组(port,mac)的集合,其中 port 称为转发端口,mac 称为转发地址。交换机 S_i 的地址转发表中关于 S_j 的条目可以用三元组(S_i,port,S_j)表示。

交换机 S_i 的地址转发表转发端口为 j 的子集称为端口 S_{ij} 的地址转发表。用 $AFT(S_i,j)$表示端口 S_{ij} 地址转发表中的转发地址的集合,简称 $A_{ij}(A_{ij}\subset D)$;$AFT(S_i)$表示交换机 S_i 所有的下行端口的地址转发表中转发地址的集合,简记为 A_i。

如果一个管理域的所有交换机的地址转发表都是完整的,那么就可以采用贝尔实验室的 Breitbart 等提出的基于完整地址转发表 AFT 的单子网拓扑发现方法[8-9]。

对于一个 $AFT(S_i,\text{port},S_j)$,根据它和根节点的关系和方向,可分为逆根 AFT、逆非根 AFT、根 AFT 和非根 AFT,如表 1 所示。本算法结合谓词逻辑推理方法[5,10]和基于生成树协议的拓扑发现算法。先用连接推理技术进行计算,"修补" 不完整的逆根 AFT 数据(或其他类型的 AFT),然后采用生成树协议算法计算拓扑结构。

表 1　AFT 类型

Table 1　AFT type

AFT 类型	含义	在生成树协议算法中的作用
逆根 AFT	S_j 为根节点,port 为该 S_i 的根端口	是整个算法的基础,必须完备,如果不完备,无法推导出结果
逆非根 AFT	S_j 为非根节点,port 为该 S_i 的根端口	无
根 AFT	S_i 为根节点	在递推过程中使用,如果不完备,可能推导出结果
非根 AFT	S_i 为非根节点,port 为非根端口	

具体算法描述如下。

(1) 采用连接推理技术"修补"AFT 数据。

(2) 选取根节点,要求其逆根 AFT 数据完整。

(3) 如果所有的节点都不能符合作为根节点的要求,则主动去 ping 网络中的节点,更新 AFT,返回(1)。

(4) 如果选取的节点其逆根 AFT 数据完整,采用生成树协议算法计算拓扑结构。

(5) 如果在(4)中计算拓扑结构是 AFT 数据不完整,则返回到(3),重新选取其他符合条件的根节点。算法的流程图如图 1 所示。

算法的系统结构如图 2 所示。

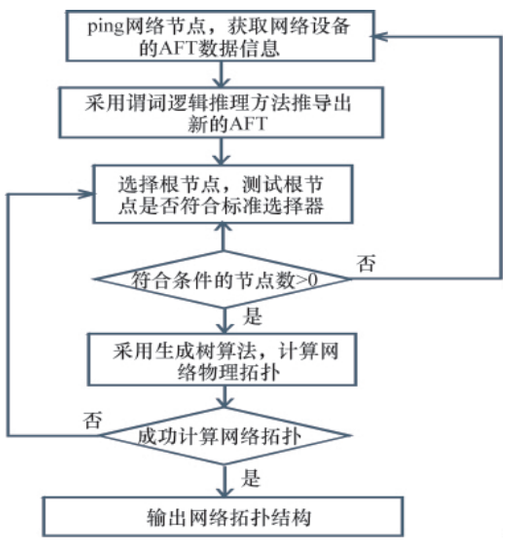


图 1　算法流程

Fig. 1　Flowchart of the algorithm

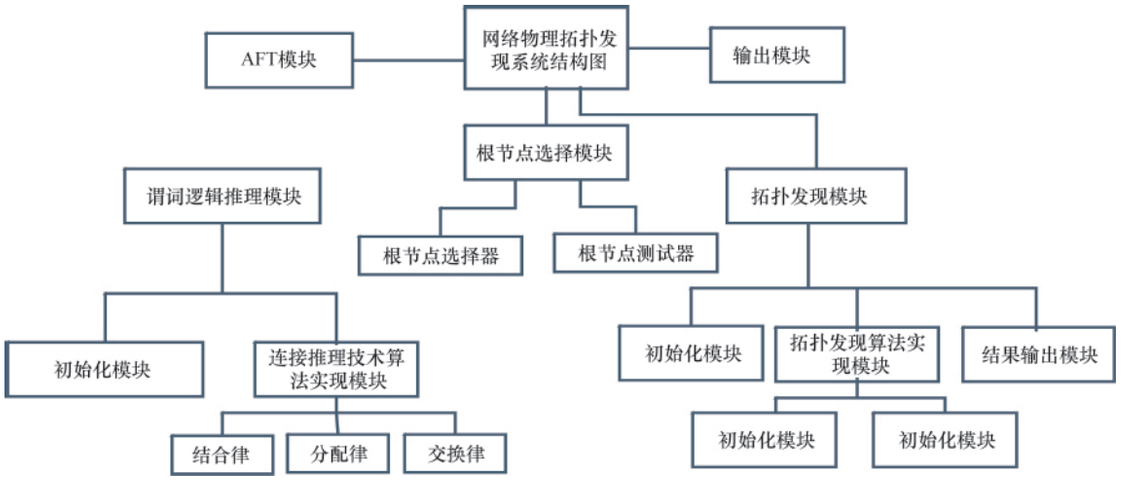


图 2　算法的系统结构

Fig. 2　System structure of the algorithm

2　算法实施与比较

设变量 P_{AFT} 为当前获得的 AFT 数量(N_{AFT})占总 AFT 数量(n_{AFT})的比重,即获得的 AFT 的完整性,$P_{AFT}=(N_{AFT}/n_{AFT})\times100\%$;设变量 $P_{TOPO}(n)$ 为采用当前算法,在网络交换机数目为 n 的情况下,获得正确网络拓扑结构的可能性,其具体计算方法为,当网络交换机数目为 n 时,共有 N 种 AFT 的获取情况,对每一种 AFT 的获取情况,都采用该算法进行运算,然后统计所有的运算结果,正确计算拓扑结构的次数占总数的比值就是 $P_{TOPO}(n)$;设变量 $P_{TOPO}(n,P_{AFT})$ 为采用当前算法,在网络交换机数目为 n,且获取 AFT 的完整性为 P_{AFT} 的情况下,获得正确网络拓扑结构的可能性;设变量 p_{TOPO} 为该算法获得正确网络拓扑结构的可能性,$p_{TOPO}=\lim\limits_{n\to\infty}\dfrac{\sum_{n=1}^{\infty}P_{TOPO}(n)}{n}$,该数值大体上表示了该算法的优劣。

采用连接推理技术的谓词逻辑推理方法,虽然可以推算出一部分的节点连接信息,但从实验分析来看,$P_{TOPO}(n)$ 数值偏低,即 $P_{TOPO}(n,P_{AFT})$ 要达到比较高的值,相应地,要求 P_{AFT} 较高。言外之意就是该算法只有在 AFT 获取“较为”完整的情况下,才能发挥作用,得到所有节点的连接关系,从而获得整个网络的拓扑结构。

采用生成树协议[11],虽然不需要完整的 AFT 数据,但建立生成树根节点相关的 AFT,即逆根 AFT 数据必须完整,这在实际网络中也是很难达到的,并且算法对逆非根 AFT 的信息没有利用。从实验分析来看,在满足逆根 AFT 完整的情况下,$P_{TOPO}(n)$ 数值相对连接推理技术较高。

总结这两个算法,可以看到,连接推理技术在 AFT 数据较为完整的情况下,才能很好发挥作用,依赖性较高。如果 AFT 不完整,连接推理技术虽然不能获取全部节点的连接关系,但可以获得部分节点的连接关系,即可以“修补”AFT 数据。而生成树协议算法,优势较为明显,在 AFT 不太完整的情况下,依然可以获得网络拓扑结果,但它要求在 AFT 不完备的情况下,AFT 符合一定的分布(逆根 AFT 数据要完整),才能发挥作用。

下面对 3 种算法进行有效性分析。其中,算法一为采用连接推理技术的谓词逻辑推理方法[6],算法二为采用生成树协议算法[7],新算法为基于谓词逻辑推理与生成树的优化链路层网络拓扑发现算法。

模拟 6 个交换机拓扑环境,分别为 3 个交换机(1 种拓扑),4 个交换机(2 种拓扑)和 5 个交换机(3 种拓扑)。

对于交换机个数为 n 的拓扑网络,其总 AFT 的数量为 $n_{AFT}=n(n-1)$。每次实验都会从 n_{AFT} 个 AFT 数据中随机抽取(0,n_{AFT})个数据进行算法的比较测试,则实验次数为 $2^{n_{AFT}}$,即 $2^{n(n-1)}$ 次。考虑到当 n=6 的时候,实验的数据为 $2^{6\times5}$,约 10.74 亿,数据量太大,构造数据和算法测试都要花费大量时间,因此本算法有效性分析仅考虑 n=3,4,5 的情况进行分析。具体拓扑结构图和拓扑简称见表 2。

这里以拓扑结构 TOPO(5_1)为例,分析算法的有效性,总共进行了 1048576 次计算。对每一运算,都分别采用算法一、算法二和新算法进行运算,最后统计实验数据见表 3。

举例说明如下。当前含有 AFT 数据 14 条,而总 AFT 数据为 n_{AFT}=20,则当前行含有的 AFT 数目占总 AFT 的比例 AFT%=0.7,此时,一共有 38760 种拓扑情况,其中算法一成功计算出 28 次拓扑结构,算法二为 20915 次,新算法为 26046 次,新算法较算法二提高了 0.132379。

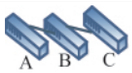
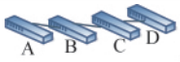
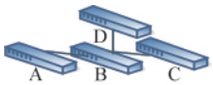
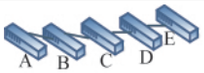
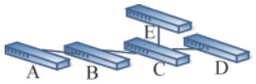
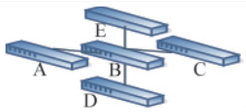
在本文的实施例中,算法一、算法二和新算法之间有效性的对比如图 3—图 5 所示。

(1) 从 $P_{TOPO}(n)$ 来看,新算法>算法二>算法一。特别地,当 AFT%>0.9,即 AFT 含量基本完善的情况下,算法一的成功率仍不到 50%,而算法二和新算法可达 100%。

(2) 随着 AFT%增大,3 种算法的 $P_{TOPO}(n)$ 均增加。其中算

表 2　实例拓扑结构

Table 2　Topology for examples

拓扑简称	AFT 总数	实验次数	拓扑图
TOPO(3)	6	64	
TOPO(4_1)	12	4096	
TOPO(4_2)	12	4096	
TOPO(5_1)	12	1048576	
TOPO(5_2)	20	1048576	
TOPO(5_3)	20	1048576	

法二和新算法增长趋势类似,AFT%从 0.5 到 0.8,$P_{TOPO}(n,P_{AFT})$显著增加;而对于算法一,AFT%从 0.8 开始,$P_{TOPO}(n,P_{AFT})$显著增长,这也符合之前所说的,算法一只有在 AFT%较大的情况下,才能发挥作用。

(3) 观察“算法二有效性的提高”曲线可以发现,当 AFT%在 0.4—0.75 区间,新算法较算法二拓扑计算成功率优势明显,而 AFT%在此区间外,算法二和新算法成功率差距不大。

(4) 在 AFT%开始的一段,即从 0 至 0.1,$P_{TOPO}(n,P_{AFT})$有一个小幅增长,原因是当前收集到的 AFT 过少,很有可能无法包含全部的交换机,此时可能会出现算法误认为已经计算出当前的拓扑情况,该拓扑结构是正确的网络拓扑结构的一个子集,而这种情况在实际中不会出现,应该第一步就是要 ping 全网节点,从而获得的 AFT 一定会覆盖全网的交换机。

(5) 对于每一次实验数据,由于算法二和新算法都依赖于拓扑生成树协议, 树的根节点的选择会影响算法的成功率。同时,如果将每次根节点的选择作为一次实验来统计算法成功率,则成功率要低于之算法二和新算法。

(6) 图 4 中表征的算法是新算法较算法一和算法二的成功率,可以看出,新算法对网络拓扑树深度大的网络结构有更好的成功率。

(7) 图 5 中表征的是新算法较算法二提高的成功率。可以看出,当网络拓扑数深度大时,新算法较算法二有更好的成功率。

表 3　算法数据表

Table 3　Data table for the algorithm

当前行 N_{AFT}	AFT%	实验次数	算法二		算法一		新算法		新算法较算法二有效性的提高/%
			成功次数	有效性/%	成功次数	有效性/%	成功次数	有效性/%	
1	0.05	20	0	0	0	0	0	0	0
2	0.1	190	10	5.2632	10	5.2632	10	5.2632	0
3	0.15	1140	22	1.9298	0	0	22	1.9298	0
4	0.2	4845	54	1.1146	8	0.1651	54	1.1146	0
5	0.25	15504	123	0.7933	16	0.1032	127	0.8191	0.0258
6	0.3	38760	227	0.5857	10	0.0258	335	0.8643	0.2786
7	0.35	77520	708	0.9133	4	0.0052	1014	1.308	0.3947
8	0.4	125970	1549	1.2297	18	0.0143	2237	1.7758	0.5462
9	0.45	167960	2772	1.6504	35	0.0208	5443	3.2407	1.5903
10	0.5	184756	6380	3.4532	37	0.02	14134	7.6501	4.1969
11	0.55	167960	14120	8.4068	21	0.0125	27936	16.6325	8.2258
12	0.6	125970	22911	18.1877	6	0.0048	38588	30.6327	12.445
13	0.65	77520	26116	33.6894	8	0.0103	37537	48.4223	14.733
14	0.7	38760	20915	53.9603	28	0.0722	26046	67.1981	13.2379
15	0.75	15504	11687	75.3805	56	0.3612	12937	83.443	8.0624
16	0.8	4845	4460	92.0537	70	1.4448	4571	94.3447	2.291
17	0.85	1140	1131	99.2105	56	4.9123	1131	99.2105	0
18	0.9	190	190	100	28	14.7368	190	100	0
19	0.95	20	20	100	8	40	20	100	0

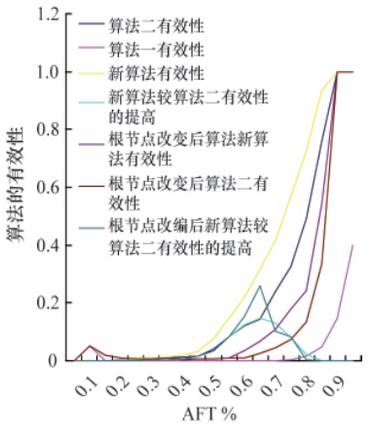


图 3　算法成功率对比

Fig. 3　Algorithm´s successful rate

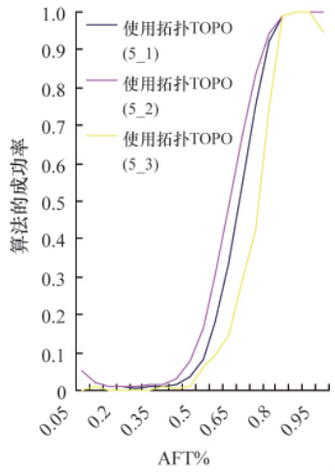


图 4　算法成功率横向对比

Fig. 4　A Comparison of algorithm´s success rates

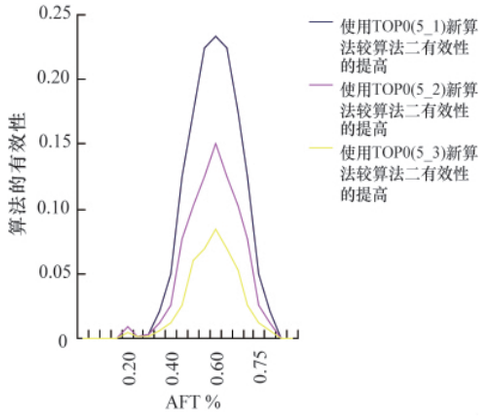


图 5　算法有效性横向对比

Fig. 5　A Comparison of algorithm´s effective ness

3　结论

本文提出的基于谓词逻辑推理与生成树的优化链路层网络拓扑发现方法，通过 Ping 网络节点，获取网络 AFT 信息,之后采用谓词逻辑推理推导出新的 AFT 信息,使用本文算法对根节点新型“修补”,之后利用生成树算法计算出拓扑结构。结合谓词推理技术的谓词逻辑推理方法和基于生成树协议的拓扑发现算法,在 AFT 数据不完全的情况下,提高了获得网络物理拓扑的可能性。算法的下一步工作是要解决当 n 的值比较大的情况下的物理拓扑发现。

参考文献(References)

[1] Huffaker B, Plummer D, Moore D, *et al*. Topology discovery by active probing [C]// Symposium on Applications and the Internet Workshops. Washington DC: IEEE Computer Society, 2002: 90–96.

[2] Donnet B, Raoult P, Friedman T, *et al*. Efficient algorithms for large–scale topology discovery[C]//Eager D L, Williamson C L, Borst S C, *et al*, eds. Proc of the ACM SIGMETRICS 2005. New York: ACM Press, 2005: 327–338.

[3] Govindan R, Tangmunarunkit H. Heuristics for internet map discovery[C] //Sidi M, Sengupta B, eds. Proc of the IEEE Infocom 2000. New York: IEEE Press, 2000: 1371–1380.

[4] Bejerano Y, Breitbart Y, Garofalakis M N, *et al*. Physical topology discovery for large multi–subent networks [C]. IEEE Infocom 2003, San Francisco, USA, March 30–April 3, 2003.

[5] 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): 7–10.
Zheng Hai, Zhang Guoqing. *Journal of Computer Research and Development*, 2002, 39(3): 7–10.

[6] 陈松, 王珊, 周明天. 一种新的物理网络拓扑发现算法[J]. 电子与信息学报, 2010, 32(1): 172–177.
Chen Song, Wang Shan, Zhou Mingtian. *Journal of Electronics & Information Technology*, 2010, 32(1): 172–177.

[7] 邓泽林, 傅明, 刘翌南. 基于生成树的网络链路层拓扑发现算法[J]. 计算机工程与应用, 2010, 46(16): 121–123.
Deng Zelin, Fu Ming, Liu Yinan. *Computer Engineering and Applications*, 2010, 46(16): 121–123.

[8] Breitbart Y, Garofalakis M, Martin C, *et al*. Topology discovery in heterogeneous IP networks [C]//Sidi M, Sengupta B, eds. Proc of the Infocom 2000. New York: IEEE Press, 2000: 265–274.

[9] 孙延涛, 吴志美, 石志强. 基于地址转发表的交换式以太网拓扑发现方法[J]. 软件学报, 2006, 17(12): 2565–2576.
Sun Yantao, Wu Zhimei, Shi Zhiqiang. *Journal of Software*, 2006, 17(12): 2565–2576.

[10] Aoyama M. Evolutionary patterns of design and design patterns[C]. IEEE Intemational Symposium on Principles of Software Evolution, Kanazawa, Japan, Nov 1–2, 2000.

[11] Son M H, Joo B S, Kim B C, *et al*. Physical topology discovery for metro Ethernet networks[J]. *ETRI Journal*, 2005, 27(4): 355–366.

(责任编辑　代丽)