

基于多子网交汇点的以太网物理拓扑发现算法

曾光^{1,2} 陈性元¹ 杜学绘^{1,2} 王超^{1,2}

(解放军信息工程大学 郑州 450002)¹ (数学工程与先进计算国家重点实验室 郑州 450002)²

摘要 首先介绍了目前以太网物理拓扑发现的主要算法及存在的问题,然后提出了一种基于多子网交汇点的拓扑发现算法。算法围绕交汇点和利用最小需求地址转发表信息来推理、约简和建立物理拓扑连接关系。结合一种典型的多子网拓扑进行了算法推导。理论与应用分析表明,该算法能够利用不完整的地址转发表构造出整个网络拓扑结构,在发现效率和准确性等方面都有了较大提高,适用于解决包含哑设备的大型、异构的多子网拓扑发现问题。

关键词 物理拓扑,子网连接树,交汇点,最小需求地址转发表

中图分类号 TP393.02 **文献标识码** A

Physical Topology Discovery Algorithm for Ethernet Based on Intersection of Multi-subnet

ZENG Guang^{1,2} CHEN Xing-yuan¹ DU Xue-hui^{1,2} WANG Chao^{1,2}

(PLA Information Engineering University, Zhengzhou 450002, China)¹

(State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450002, China)²

Abstract The main achievements were introduced in physical topology discovery for Ethernet and the defaults of those methods were pointed out, then a new algorithm was proposed based on the intersection of multi-subnet. The algorithm uses the Minimum Requirements of Address Forwarding Table(AFT) around the intersection of multi-subnet to reason and establish the connection between switchers. And combining a typical multi-subnet topology, the algorithm was derived. Theoretical analysis proves that our method correctly infers the network topology with the incomplete AFTs, and has low communication and computational overheads, in which other methods fail. The algorithm is appropriate for discovering the physical topology of large, heterogeneous Ethernet that may include multiple subnets as well as uncooperative network elements, like hubs.

Keywords Physical topology, Connecting tree of subnet, Intersection node, Minimum requirements of address forwarding table

1 引言

网络拓扑是指组成网络的各种设备之间的连接关系,是计算机网络的重要特征之一。从网络管理的角度而言,准确、及时的拓扑信息对于设备部署、资源管理、性能监测和故障定位等一系列网络维护工作具有重要意义。从网络安全的角度而言,拓扑信息同样对入侵检测、网络监控、防火墙部署以及访问控制等安全服务十分重要,会影响到安全策略实施的有效性。

网络拓扑发现是指通过获取一定的信息,分析得到网络中各个节点之间的连接关系。目前,网络拓扑的发现方法分为基于网络层的逻辑拓扑发现和基于链路层的物理拓扑发现。早期对拓扑发现的研究主要集中在网络层,利用 SNMP 或 ICMP 分析路由表,确定路由设备、子网及其之间的连接关系。随着交换技术的广泛应用,以太网的规模迅速增加,交换机已经成为政府、企业等进行网络部署的主要手段。相对于渐趋成熟的网络层拓扑发现,由于二层设备固有的透明性、异

构性和拓扑的复杂性,使得其拓扑发现面临更多难点和挑战。利用生成树协议(Spanning Tree Protocol, STP)进行拓扑发现是一种可行方法^[1,2],但在异构网络上存在兼容性问题;Loran Networks 所提出的基于流量分析的方法^[3]适用于异构网络,但算法要求所有的网桥设备都是可网管的,并且在若干端口流量相似的情况下无法判别。由于网桥、交换机等二层设备均采用地址转发表(Address Forwarding Table, AFT)进行帧转发,基于地址转发表的方法^[4-15]已经成为目前适用范围最广的通用方法。

2 相关研究

2000年,贝尔实验室的 Breitbart 等人^[4]开创性地对链路层拓扑发现问题进行了形式化描述,提出一种基于完整地址转发表的拓扑发现算法,并给出了严格的数学证明。算法的核心在于直接连接定理,该定理的前提是所有交换机的地址转发表都是完整的,这在实际网络中是难以实现的。为此,郑海等人^[5]提出一种改进算法,该方法仅要求下行端口转发

到稿日期:2013-07-02 返修日期:2013-10-19 本文受国家 973 重点基础研究发展计划(2011CB311801),国家 863 高技术研究发展计划(2012AA012704),河南省科技创新人才计划(114200510001)资助。

曾光(1988-),男,硕士,主要研究方向为网络安全, E-mail: myzengguang@sina.com;陈性元(1963-),男,教授,博士生导师,主要研究方向为信息安全;杜学绘(1968-),女,教授,硕士生导师,主要研究方向为网络安全;王超(1975-),男,博士,讲师,主要研究方向为网络安全。

表完整即可构造出交换机之间的连接关系。随后,国内不少研究人员在郑海的理论基础上进行进一步的研究。如文献[6-9]对该算法进行完善,能够发现实际网络中存在的哑设备,并从理论上证明了叶子节点法^[6]能在多项式时间内发现物理拓扑。

为了降低地址转发表的完整性要求,Lowekamp 等人^[10]在 SIGCOMM 2001 会议上提出一种基于非完整地址转发表的拓扑发现算法。算法基于间接连接定理(MKR),核心思想是利用反证法排除不可能的连接关系。文献[11]利用以太网树型无环的特点,提出了另一种更简单的方法。算法给出了一组判定定理,用于确定节点之间的祖孙关系,不断逼近每个节点的最近祖先,最终构建整棵网络拓扑树。

上述算法需要首先选择一台主机作为根节点,并确定根节点与各个节点的根端口,因此算法适用于单子网拓扑发现^[12]。当网络中存在多子网交换域时,隶属不同子网的二层设备即使物理上直接相连,相互通信也必须经过路由器转发,使得部分节点对于其邻近节点完全不可见,难以正确地发现网络拓扑。

2003年,Bejerano 等人^[13]提出一种基于完整地址转发表的多子网拓扑发现算法。算法的核心思想是利用路径约束信息(Path Constraints)反复优化不同节点对之间的概要路径(Skeleton Paths),最终确定一条唯一的路径。2009年,Bejerano 等人^[14]针对文献[13]中算法复杂度过高等不足,提出一种更简单的方法。算法通过定义概要树(Skeleton Tree)结构辅助发现单子网连接树(Connection Tree);然后基于图论进行后序遍历,补充完整各个节点的上行端口地址转发表;最后基于固定节点(Anchor)进行概要树合并,直至拓扑发现完毕。算法仅需要每个子网连接树的下行端口转发表是完整的。2010年,Gobjuka 等人^[15]提出一种新的启发式多子网拓扑发现算法,并证明该算法能够发现所有可能的拓扑结构。算法的核心思想是通过定义可达节点集(Reachable Set, RS)和扩展规则集(Extension Rules),将多子网拓扑发现问题转化为单子网拓扑发现问题。在网络不包含哑设备以及每个节点的RSs丢失率不超过50%的情况下,算法是完备的。

文献[16,17]虽然进行了多子网拓扑发现的研究,但算法侧重从网络层与链路层的结果融合的角度对文献[5]的算法进行扩展,忽略了对发现的单子网连接树进行相应合并,未能完全适用于多子网交换域的拓扑发现情况。

本文在上述研究工作的基础上,提出了一种基于多子网交汇点的拓扑发现算法。算法围绕交汇点进行拓扑发现,利用改进的连接信息扩展规则和最小需求地址转发表降低完整性要求,提高拓扑发现的效率,适用于包含哑设备的大型、异构的多子网拓扑发现。

3 物理拓扑发现方法

3.1 相关定义及其性质

在链路层中,网络的拓扑结构可以是任意形状的。为确保节点之间能够正常通信,避免广播风暴的发生,以太网通过运行生成树协议,消除备份链路影响,即交换域(Switched Domain, SD)中任意一对网络节点之间有且仅有一条只包含二层设备的路径。子网(Subnet)是指任意两节点不经过路由器通信的最大IP地址集合。子网和交换域并不是一一对应

的关系,如图1所示,一个交换域可以包含一个子网,也可以包含多个子网。

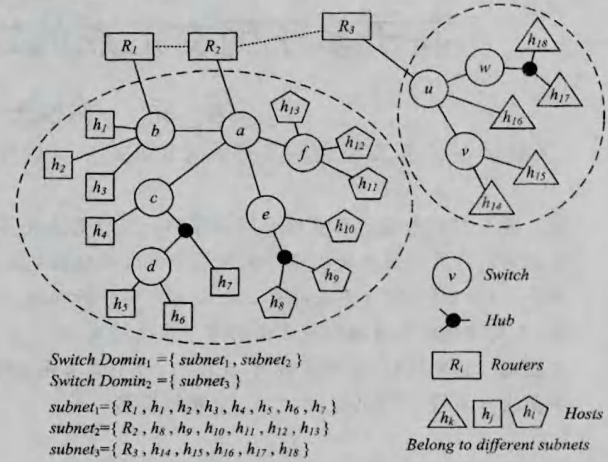


图1 典型管理域的网络拓扑结构

我们将交换域抽象成一棵无向连通树 $T(V, E)$,其中 V 代表路由器、交换机、主机等网络设备, E 代表设备之间的直接连接关系。为了算法描述方便,我们在后面的叙述中用 (v, i) 表示节点 $v(v \in V)$ 的第 i 个端口,简记为 v_i ;用 D_v 表示节点 v 的所有活动端口集, $|D_v|$ 表示该集合的端口数。用 $\text{Path}(v, w)$ 表示任意两节点之间的路径。用 $\text{AFT}(v_i)$ 表示端口 v_i 通过逆向学习机制所得的地址转发表, $\text{RS}(v_i)$ 表示在交换域中通过生成树路径可以到达端口 v_i 的节点集;用 $\text{CRS}(v_i)$ 表示除端口 v_i 以外,节点 v 的其他端口的可达节点集的并集,包括节点 v 本身,称为 $\text{RS}(v_i)$ 的补集;用 $\text{RS}(v)$ 表示节点 v 所有端口的可达节点集,包括节点 v 本身。

同时,我们用 N 表示网络中所包含的子网集合, N_i 表示子网集合中第 i 个子网的节点集合,并将其抽象成一棵子网连接树 $T^{N_i}(V^{N_i}, E^{N_i})$,其包含子网中任意一对节点 $(v, w \in N_i)$ 所确定的路径。然后用 $\text{Subnet}(v)$ 表示节点 v 所属连接树所连接的子网集合,即 $\text{Subnet}(v) \subseteq N$ 。如图2所示。

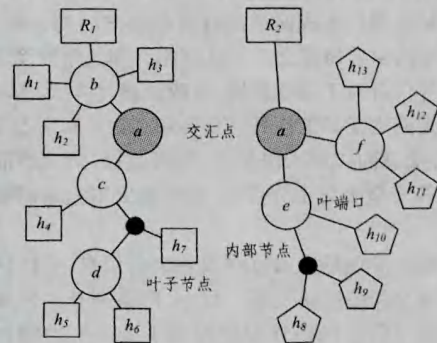


图2 子网连接树与多子网交汇点

定义1(交汇点) 表示在交换域中与所有的子网连接树都存在关联的节点 v ,即 $\text{Subnet}(v) = N$ 。

定义2(上行端口) 表示与根节点存在连接关系的节点端口。

定义3(下行端口) 表示除上行端口以外的所有活动端口。

定义4(叶端口) 如果 $|D_v| > 1$,并且 $\text{RS}(v_i)$ 中未出现交换域内其他交换机的地址,则称 v_i 为交换机的叶端口。

定义 5 最小需求地址转发表 (Minimum Required Addresses Forwarding Table, MRAFT)。对节点 v 的叶端口所包含的主机地址进行简化, 只保留 $Subnet(v)$ 中每个子网的一个节点地址。

性质 1 如果 $|D_v|=1$, 则节点 v 为子网连接树的叶子节点, 且 $v \in N_i$, 代表主机和路由器; 如果 $|D_v|>1$, 节点 v 可能与一个或多个子网关联, 属于子网连接树的内部节点, 代表网桥和交换机, 如图 2 所示。

性质 2 如果 $w \in AFT(v_i)$, $Subnet(v) = Subnet(w)$, 则节点 v, w 之间必然存在一条路径 $Path(v_i, w_j)$ 。

性质 3 如果 $w \in AFT(v_i)$, $Subnet(v) \neq Subnet(w)$, 并且存在节点 u , 使 $Subnet(u) = Subnet(w)$, 则节点 w, u 之间的路径 $Path(w_j, u_k)$ 必然包含节点 v , 即存在一条生成树路径 $w_j \cdots v_i \cdots u_k$ 。但对于 w_j 而言, $v \notin AFT(w_j)$ 。

性质 4 如果 $AFT(v_i)$ 包含所有能向端口 v_i 传递数据包的节点集, 则称 $AFT(v_i)$ 是完整的。如果 $RS(v_i)$ 包含所有通过路径可以到达端口 v_i 的节点集, 则称 $RS(v_i)$ 是完整的。对于单子网交换域, $RS(v_i)$ 等价于 $AFT(v_i)$; 对于多子网交换域, $RS(v_i)$ 是 $AFT(v_i)$ 的超集。当节点 v 所有端口的 RS s 信息都完整时, 称 $RS(v)$ 是完整的。

3.2 理论基础

3.2.1 交汇点定理

定理 1 在交换域中, 如果所有交换机的地址转发表是完整的, 则交汇点的 RS s 信息是完整的, 并且包含的地址数目最多。

证明: 根据定义和性质 1, $Subnet(v) = N$, 即交汇点 v 与所有子网的节点都存在连接关系, 由于交换机的地址转发表是完整的, 根据性质 4, 交汇点的任意端口包含所有通过节点 v 所属的子网连接树的路径可以到达的节点集, 因此交汇点的 $RS(v)$ 是完整的。将交汇点作为交换域拓扑树的根节点 r , 如图 3 所示, 根据性质 2, 其连接节点下行端口的 RS s 信息是完整的; 根据性质 3, 因为不能同其他子网的节点进行直接通信, 则其连接节点上行端口的 RS s 信息不完整, 所以交汇点的节点地址数目最多。证明完毕。

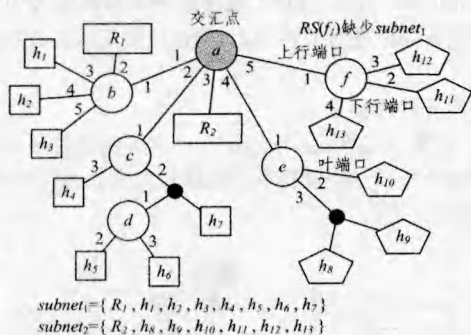


图 3 以交汇点为根的多子网拓扑结构

3.2.2 连接判定定理

定理 2 以交汇点作为拓扑树的根节点, 并且所有交换机的地址转发表都是完整的, 如果节点 v 下行端口 v_i 和节点 w 上行端口 w_j 满足条件 $RS(v_i) = CRS(w_j)$, 则两端口直接连接。如果不满足该条件, 但节点 v 下行端口 v_i 与多个节点 ($w \in V'$) 上行端口满足条件 $RS(v_i) = \bigcup_{w \in V'} CRS(w_k)$, 则上述节点通过哑设备连接。

证明: 根据性质 3 和定理 1, 易知所有节点下行端口的 RS s 是完整的, 定理 2 根据文献[5, 6]中的定理改进而来, 因此证明从略。

推论 1 在以交汇点为根的拓扑树中, 如果节点 v 是节点 w 的父亲节点, 则 $CRS(w_i) \subset CRS(v_i)$ 。

证明: 根据定理 2, 节点 v 的下行端口 v_i 和节点 w 的上行端口 w_j 连接, 则 $CRS(w_i) \subseteq RS(v_i)$ 。由于 $RS(v_i) \subset CRS(v_i)$, 则 $CRS(w_i) \subset CRS(v_i)$ 。

3.2.3 最小需求地址转发表定理

定理 3 根据 MRAFT 要求对节点 v 的叶端口所包含的叶子节点地址进行简化, 只保留 $Subnet(v)$ 中每个子网的一个节点地址, 不影响网络拓扑的发现。

证明: 根据性质 1, 叶子节点属于一棵子网连接树, 内部节点可能与多个子网关联, 如图 3 所示, 节点 a 与 $Subnet_1$ 和 $Subnet_2$ 关联, 属于多棵子网连接树。根据定义保留 $Subnet(v)$ 中每个子网的一个节点地址, 将图 3 所示的网络拓扑进行简化, 所得拓扑如图 4 所示。即将叶子节点地址与邻接的交换机绑定, 作为交换机的识别地址, 表明该交换机相对于其他交换机的方向, 仍满足定理 2 的连接判定条件; 因此删除冗余的叶子节点地址, 不影响网络拓扑的发现。

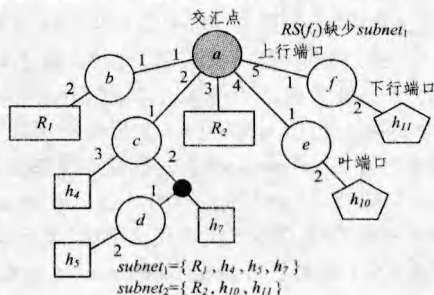


图 4 满足 MRAFT 需求的多子网拓扑结构

定理 4 以交汇点作为拓扑树的根节点, 如果所有交换机都满足 MRAFT 要求, 则可以发现唯一的链路层网络拓扑结构。

证明: 如果所有交换机端口都满足 MRAFT 要求, 假设删除所有绑定的叶子节点以及与叶子节点相连的边, 则邻接的交换机节点的 $|D_v|$ 变为 1, 演化为新的叶子节点。同时, 由于叶子节点不与其他节点相连, 因此得到的拓扑图连通不存在回路, 仍是一棵树。依次重复上述操作, 可以不重复地遍历所有的节点和边, 直到拓扑发现完毕为止[7], 即满足定理 2 的连接判定条件, 可以发现唯一的链路层拓扑。证明完毕。

推论 2 拓扑发现所需要的地址转发表信息并不随着网络主机规模的扩大而增加, 但会随着子网划分的数量而线性增加。

3.2.4 连接信息扩展规则

规则 1 在交换域中, 如果端口 v_i 连接节点 w , 则 $w \in RS(v_i)$; 如果存在连接 (v_i, w_j) , 则 $CRS(w_j) \subseteq RS(v_i)$, $CRS(v_i) \subseteq RS(w_j)$ [15]。

规则 2 如果 $RS(v_i) \cap RS(w_p) \neq \emptyset$, $RS(v_i) \cap RS(w_q) \neq \emptyset$, 并且 $q \neq p$, 则 $w \in RS(v_i)$ 。

证明: 如图 5 所示, $x \in RS(w_p)$, $y \in RS(w_q)$, $q \neq p$, 且 $RS(v_i)$ 同时包含节点 x 和 y 。假设 $RS(v_i)$ 不包含节点 w , 由于节点 v, w 之间必然存在一条路径, 即存在端口 w_j 与端口 v_i

以外节点的任一端口 v_k 相连, $k \neq i$ 。显然节点 v, x 或者 v, y 之间存在两条路径, 即生成树中存在回环, 矛盾。

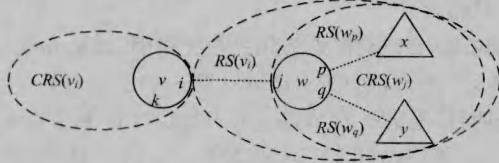


图5 连接信息扩展规则

规则3 如果 $v \in RS(w_j)$, 并且 $CRS(w_j) \cap RS(v_i) \neq \phi$, 则 $CRS(w_j) \subseteq CRS(v_i)$ 。

证明: 如图5所示, $RS(w_j)$ 包含节点 v , 存在端口 $p \neq j$, $RS(v_i)$ 和 $RS(w_p)$ 包含相同的节点 x , 则 $RS(v_i)$ 包含节点 w 和 $RS(w_p)$ 的所有节点。假设还存在端口 v_k 使得 $RS(v_k)$ 和 $RS(w_q)$ 包含相同的节点 y , 并且 $q \neq j, q \neq p$ 。如果 $k \neq i$, 显然节点 v, w 之间存在两条路径, 即生成树中存在回环, 矛盾。

3.3 算法描述与推导

3.3.1 数据预处理

在指定的拓扑发现范围内, 以子网为单位进行节点发现、确定节点类型以及子网所属交换域; 然后以交换域为单位采集交换机地址转发表信息。

在上述过程中, 主要用到 RFC-1213 定义的 MIB-II 库和 RFC-1443 定义的 Bridge MIB 库信息。首先, 通过 Ping 管理域内所有的 IP 地址来确定活动节点; 其次, 利用标准 SNMP MIB-II 库中 system 组的 sysServices 变量确定设备工作的协议层次, 结合 ip 组的 ipForwarding 变量确定设备类型; 第三, 利用 Bridge MIB 库中 dot1dStp 组的 dot1dStpDesignateRoot 变量, 根据生成树规则, 确定子网所属交换域; 最后, 以交换域为单位, 采集 MIB-II 库中 ip 组的 ipAddrTable 信息, 获取交换机的地址转发表。

3.3.2 基于多子网交汇点的拓扑发现算法

以交换域为单位, 首先根据定义和性质确定交汇点; 然后以交汇点为核心推理、约简和确定连接信息, 算法具体描述如下:

/* 输入: 交换域, 节点集, 交换机地址转发表 */
/* 输出: 以交汇点为根的多子网拓扑结构 */

Procedure BuildTree(SD, V, AFTs)

Initialization;

1. for every node $v \in V$ do
2. for every port $v_i \in D_v$ do
3. Initialize $RS(v_i)$, Calculate $CRS(v_i)$;
4. end for
5. Calculate $Subnet(v)$, $|Subnet(v)|$, $RS(v)$ and $|RS(v)|$;
6. end for

Connection Information Reasoning;

7. $L = \text{Sorted list of } V \text{ in non-increasing order according to their } |Subnet(v)| \text{ and } |RS(v)| \text{ values. // 定理 1}$
8. $v = \text{Get the first node in } L$
9. while $Subnet(v) \neq N$ do // 定义 1
10. Around node v apply RS Extension Rule;
11. Extend RSs and CRSs;
12. end while
13. $r = v$;
14. while $RS(r)$ is not satisfied MRAFT do // 定义 5
15. for every node $v \in V - \{r\}$ do

16. Around node r apply RS Extension Rule;

17. Extend RSs and CRSs ;

18. end for

19. end while

20. for every node $v \in V - \{r\}$ do

21. Around node r apply simplify of MRAFT; // 定理 3

22. Around node r confirm the up-port v_i // 定义 2

23. for every port $r_i \in D_r$ do

24. Around $RS(r_i)$ apply RS Extension Rule;

25. Update RSs and CRSs;

26. end for

27. end for

Direction Connection Discovery;

28. $L' = \text{Sorted list of } V \text{ in non-increasing order according to their } |CRS(w_1)| \text{ values. // 推论 1}$

29. while $L' \neq \phi$ do

30. $v = \text{Get and remove the first node in } L'$.

31. for every port $v_i \in D_v - \{v_1\}$ do

32. Find the first node $w \in L'$ s. t. $CRS(w_1) \subseteq CRS(v_i)$;

33. if $RS(v_i) = CRS(w_1)$ then // 定理 2

34. Connect port v_i to w_1 ;

35. else

36. while $RS(v_i) \neq \bigcup_{w \in W} CRS(w_1)$ do

37. Find the next node $w \in L'$ s. t. $CRS(w_1) \subseteq CRS(v_i)$;

38. $W = W \cup \{w\}$;

39. Connect port v_i to w_1 ;

40. end while

41. end if

42. end for

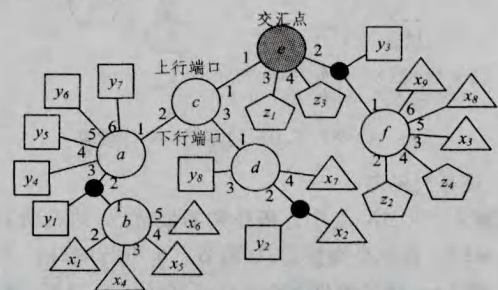
43. end while

44. return $T(V, E)$

如上所述, 算法首先根据定义 1 和定理 1 确定多子网的交汇点。然后根据性质 1 和定义 5 判断是否满足 MRAFT 要求, 若不满足, 则利用扩展规则推理补充 RSs 信息, 重复判断; 若满足, 则以交汇点作为拓扑树的根节点, 根据定理 3 进行 MRAFT 约简, 并利用扩展规则推理补充完整所有交换机下行端口的 RSs 信息。最后, 根据推论 1 对所有节点进行排序, 并通过定理 2 进行连接判定, 构建以交汇点为根的交换域拓扑树。

3.3.3 算法示例推导

为了便于表述算法, 本文给出了一种典型的多子网拓扑结构, 如图 6 所示。假设获取的地址转发表如表 1 所列, 显然拓扑信息不完整。



$subnet_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$

$subnet_2 = \{y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8\}$

$subnet_3 = \{z_1, z_2, z_3, z_4\}$

图6 典型的多子网拓扑结构

表1 图6的地址转发表信息

port	Initial RS	port	Initial RS	port	Initial RS
a ₁	x ₃ , y ₂	b ₅	x ₆	e ₃	z ₁
a ₂	x ₄ , x ₆ , y ₁	c ₁	x ₃ , x ₆ , x ₉	e ₄	z ₃
a ₃	y ₄	c ₂	x ₁ , y ₁ , y ₆	f ₁	x ₁ , z ₃
a ₄	y ₅	c ₃	y ₂ , y ₈	f ₂	z ₂
a ₅	y ₆	d ₁	x ₁ , x ₃ , y ₃	f ₃	x ₃
a ₆	y ₇	d ₂	x ₂ , y ₂	f ₄	z ₄
b ₁	x ₃ , x ₇	d ₃	y ₈	f ₅	x ₈
b ₂	x ₁	d ₄	x ₇	f ₆	x ₉
b ₃	x ₄	e ₁	x ₁ , y ₂ , y ₇		
b ₄	x ₅	e ₂	x ₃ , y ₃ , x ₆		

首先,由于 $Subnet(e) = \{N_1, N_2, N_3\} = N$, 根据定义1可以确定节点 e 为多子网的交汇点。其次,由于地址转发表不完整,无法确定节点 e 与节点 b 之间的连接关系,即 $RS(e)$ 不满足 MRAFT 要求。利用扩展规则围绕交汇点 e 进行连接信息推理,例如通过规则2可以确定 $\{a, c, d\} \in RS(e_1)$, $\{f\} \in RS(e_2)$, 通过规则3可以确定 $\{e\} \in RS(a_1)$, $RS(c_1)$, $RS(d_1)$ 和 $RS(f_1)$; 然后再运行规则2和规则3可得 $\{b\} \in RS(e_1)$, $\{e\} \in RS(b_1)$, 并且扩展得 $RS(e)$ 的信息完整,满足 MRAFT 要求。以交汇点 e 为根节点,对拓扑信息进行约简,如表2所列。

表2 进行 MRAFT 简化后的 RSs 信息

port	RS	port	RS
a ₁	x ₂ , x ₃ , y ₂ , y ₃ , e	d ₂	x ₂ , y ₂
a ₂	x ₁ , y ₁	e ₁	x ₁ , x ₂ , y ₁ , y ₂ , a, b, c, d
b ₁	x ₂ , x ₃ , y ₃ , e	e ₂	x ₃ , y ₃ , z ₂ , f
b ₂	x ₁	e ₃	z ₁
c ₁	x ₃ , y ₃ , e	f ₁	x ₁ , x ₂ , z ₁ , e
c ₂	x ₁ , y ₁	f ₂	z ₂
c ₃	x ₂ , y ₂	f ₃	x ₃
d ₁	x ₁ , x ₃ , y ₁ , y ₃ , e		

然后,利用扩展规则推理补充完整所有交换机下行端口的 RSs 信息,例如 $\{a, b\} \in RS(c_2)$, $\{d\} \in RS(c_3)$ 等。最后,根据连接判断定理构建网络拓扑,并结合初始叶端口信息进行拓扑补充。

3.4 算法对比分析

从前面的推导可以得出,算法发现的连接关系结果同实际网络一致。假设网络中存在 n 个节点、每个节点端口数为 m 。通过分析,围绕交汇点进行连接判断和信息扩展需要对每个交换机的端口进行判断,即要进行 $m \cdot (n-1)$ 次对比,每次的计算复杂度为 n ,而在最坏的情况下,需要对所有节点进行两两对比和连接信息扩展,因此算法的计算复杂度为 $O(n^3)$ 。

表3从4个方面对本文方法和其他基于地址转发表的方法进行比较,包括是否支持多子网拓扑发现、是否要求地址转发表的完整性、能否发现哑设备以及算法的时间复杂度。

表3 物理拓扑发现算法对比

Algorithm	Multi-subnets	Completeness	Hub	Complexity
DC ^[4]	No support	Require	No support	$O(n^2)$
郑海 ^[5]	No support	Partly require	No support	$O(n)$
MKR ^[10]	Partly support	No require	Support	$O(n^2)$
Skeleton Path ^[13]	Support	Require	Support	$O(n^7)$
Skeleton Tree ^[14]	Support	Partly require	Support	$O(n^3)$
RS-ER ^[15]	Support	No require	No support	$O(n^3)$
本文算法	Support	No require	Support	$<O(n^3)$

如表3所列, MKR 算法^[10]只能支持部分多子网拓扑发现,例如表1的地址转发表便无法判断连接 (e_1, b_1) 。Skeleton Path^[13]和 Skeleton Tree 算法^[14]都支持多子网拓扑发现,但是 Skeleton Path 算法的计算复杂度过高,不易实现; Skeleton Tree 算法简单易实现,但在进行概要树合并时,可能由于缺乏共同的固定节点导致拓扑发现失败,如图6所示,3棵子网连接树都无法进行合并,不保证一定能够发现完整的拓扑结构。RS-ER 算法^[15]能够在地址转发表不完整的情况下发现多子网拓扑,但要求网络不包含哑设备,以及每个节点的 RSs 丢失率不超过 50%,并且部分扩展规则(Subnet Rule)要求地址转发表信息完整,制约了算法的适用性。

本文算法围绕多子网交汇点进行拓扑发现,利用最小需求地址转发表降低完整性要求;同时,相比于 RS-ER 算法^[15]需要反复对任意一对节点的端口进行连接信息推理和判断,本文算法依托交汇点端口的 RSs 信息划分节点集,仅需要计算和推理每个节点与其父亲节点之间的连接关系,因此时间复杂度要比文献[15]中的算法小,提高了拓扑的发现效率,并且能够发现实际网络中包含的哑设备。

结束语 物理拓扑发现对于网络管理、协议设计和网络安全等领域具有重要的作用。本文提出的基于多子网交汇点的拓扑发现方法适用于包含哑设备的异构网络环境,围绕多子网交汇点和利用最小需求地址转发表信息推理和确定连接信息,降低了地址转发表的完整性要求,弱化了拓扑发现的条件,并且提高了拓扑发现的效率,使得算法更具有实际应用价值。同时,我们还证明了拓扑发现所需要的地址转发表信息并不随着网络主机规模的扩大而增加,但会随着子网划分的数量而线性增加。

参考文献

- [1] 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
- [2] 李元龙, 邱玉祥. 基于生成树协议的交换域拓扑发现算法[J]. 计算机科学, 2012, 39(11): 288-290
- [3] Schenkel D, Slavitch M, Dawes N. Method of determining the topology of a network of objects[P]. U. S, 6928475. 2005-8-9
- [4] 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
- [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): 264-268
- [6] 张国强, 张国清, 李仰耀. 物理网络拓扑发现算法的研究和系统实现[J]. 小型微型计算机系统, 2006, 27(1): 13-16
- [7] 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
- [8] 马晓波, 杨国林, 马志强, 等. 异构 IP 网络物理拓扑发现的改进算法[J]. 微处理机, 2011, 1(2): 18-20
- [9] Xiong Y, Gu Z, Jin W. Research and Implement of an Algorithm for Physical Topology Automatic Discovery in Switched Ethernet[J]. Physics Procedia, 2012, 33: 252-257
- [10] Lowekamp B, O'Hallaron D, Gross T. Topology discovery for large Ethernet networks[J]. ACM SIGCOMM Computer Communication Review, ACM, 2001, 31(4): 237-248

3.2 组装的构件近似满足需求

假设有如表 5 的构件满足需求的情况。在这里,给出 3 种组装方式的匹配性度量规则如下。

表 5 近似满足需求的构件

构件	需求	对需求的满足情况
A ₁	R1{R1 _{pre} , R1 _{post} }	{R1 _{pre} } _[a₁, a₂] (A ₁) ₁ {R1 _{post} }
A ₂	R2{R2 _{pre} , R2 _{post} }	{R2 _{pre} } _[a₃, a₄] (A ₂) ₁ {R2 _{post} }
A ₃	R3{R3 _{pre} , R3 _{post} }	{R3 _{pre} ∧ b} _[a₅, a₆] (A ₃) ₁ {R3 _{post} }
A ₄	R4{R4 _{pre} , R4 _{post} }	{R4 _{pre} ∧ b'} _[a₇, a₈] (A ₄) ₁ {R4 _{post} }

(1) 顺序组装匹配性度量规则

$$\mathcal{S}[A_1; A_2] = \tau, \tau \in [0, a_2 \cdot a_4]$$

(2) 顺序组装匹配性度量规则

$$\mathcal{S}[A_3 \bowtie A_4] = \tau, \tau \in [0, \max\{a_6, a_8\}]$$

(3) 重复组装匹配性度量规则

$$\mathcal{S}[\mu A_3, F(A_3)] = \tau \times \chi_{[0, L]}(l), \tau \in [0, 1]$$

其中, l 表示循环圈数, L 表示循环圈数阈值。

结束语 本文主要从量化角度研究了构件以及基于构件的软件对需求匹配的问题。首先,通过比较分析了需求前置、后置与构件前置、后置的差别,提出一种构件实现需求的匹配性度量方法。然后,考虑基于构件的软件,通过复用构件之间的顺序组装、选择组装和重复组装 3 种约束关系,给出基于构件的软件的匹配性度量规则。在后续的工作中,我们会进一步考虑如何将本文的研究方法更好地应用到基于构件的软件工程的实际工作中。

参考文献

[1] Jeng J J, Cheng B H C. Specification Matching for Software Reuse: A Foundation [C] // Proceedings of the ACM SIGSOFT Symposium on Software Reusability. Seattle, Washington, USA, 1995: 97-105

[2] 王志坚, 费玉奎, 姜渊清. 软件构件技术及其应用 [M]. 北京: 科学出版社, 2005

[3] 张正, 左春, 王裕国, 等. 基于语义的领域构件接口名称匹配方法 [J]. 通信学报, 2007, 28(5): 73-79

[4] Zaremski A M, Wing J M. Specification Matching of Software Components [C] // ACM Transactions on Software Engineering and Methodology. 1997, 6: 333-369

[5] Zaremski A M. Signature and Specification Matching [D]. Carnegie Mellon University, 1996

[6] Arbab F. Coordination for Component Composition [J]. Electronic Notes in Theoretical Computer Science, 2006, 160: 15-40

[7] He J F, Li X S, Liu Z M. Component-Based Software Engineer-

ing [C] // Proceedings of ICTAC' 2005. 2005: 70-95

[8] He J F, Li X S, Liu Z M. A Theory of Reactive Components [J]. Electronic Notes in Theoretical Computer Science, 2006, 160: 173-195

[9] Ke W, Li X S, Liu Z M, et al. rCOS: A Formal Model-driven Engineering Method for Component-based Software [J]. Frontiers of Computer Science, 2012, 6(1): 17-39

[10] Abdellatif T, Bensalem S, Combaz J, et al. Rigorous design of robot software: A formal component-based approach [J]. Robotics and Autonomous Systems, 2012, 60(12): 1563-1578

[11] Farhad A, Peter C O. Formal Aspects of Component Software: 8th International Symposium, FACS 2011, Oslo, Norway, September 14-16, 2011, Revised Selected Papers [M]. Berlin, Heidelberg: Springer, 2012

[12] Chen Z B, Liu Z M, Ravn A P, et al. Refinement and verification in component-based model-driven design [J]. Science of Computer Programming, 2009, 74(4): 168-196

[13] Zhan N J, Kang E Y, Liu Z M. Component Publications and Compositions [J]. Lecture Notes in Computer Science, 2010, 5713: 238-257

[14] Lei B, Liu Z M, Morisset C, et al. State Based Robustness Testing for Components [J]. Electronic Notes in Theoretical Computer Science, 2010, 260(1): 173-188

[15] 雷斌, 王林章, 卜磊, 等. 基于状态机模型的构件健壮性测试 [J]. 软件学报, 2010, 21(5): 930-941

[16] 岳慧敏. 基于语义描述的构件检索技术研究 [D]. 青岛: 中国海洋大学, 2012

[17] 周龙. 基于相似度计算的构件匹配 [D]. 昆明: 云南大学, 2012

[18] Cheng B H C, Jeng J J. Reusing Analogous Components [J]. IEEE Transactions on Knowledge and Data Engineering, 1997, 9(2): 341-349

[19] 马亮, 孙家. 基于规约匹配的构件检索 [J]. 小型微型计算机系统, 2002, 23(10): 1153-1157

[20] 李文敬, 元昌安, 廖伟志. 基于本体相似度的构件查询算法研究 [J]. 计算机工程与科学, 2010, 32(8): 154-160

[21] 吴新星. 基于语言的软件可信性度量理论及其应用 [D]. 上海: 华东师范大学, 2011

[22] Hoare C A R. Let's Make Models (Abstract) [C] // Proc. CONCUR'90, Lecture Notes in Computer Science. 1990, 458: 32

[23] Hailperin T. Probability Logic [J]. Notre Dame Journal of Formal Logic, 1984, 25(3): 198-212

[24] 王国俊, 王伟. 逻辑度量空间 [J]. 数学学报, 2001, 44(1): 159-168

(上接第 177 页)

[11] 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 [J]. 计算机研究与发展, 2007, 44(2): 208-215

[12] Pandey S, Choi M J, Won Y J, et al. SNMP - based enterprise IP network topology discovery [J]. International Journal of Network Management, 2011, 21(3): 169-184

[13] Bejerano Y, Breitbart Y, Garofalakis M, et al. Physical topology discovery for large multisubnet networks [C] // INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies. IEEE, 2003, 1: 342-352

[14] Bejerano Y. Taking the skeletons out of the closets: A simple and efficient topology discovery scheme for large Ethernet LANs [J]. IEEE/ACM Transactions on Networking, 2009, 17(5): 1385-1398

[15] Gobjuka H, Breitbart Y J. Ethernet topology discovery for networks with incomplete information [J]. IEEE/ACM Transactions on Networking, 2010, 18(4): 1220-1233

[16] 马晓波, 杨国林. 通用的异构多子网物理拓扑发现简单算法 [J]. 计算机应用与软件, 2013, 38(5): 206-208

[17] 杨国林, 孙巧凯, 马晓波. 异构多子网物理拓扑发现算法的研究和改进 [J]. 内蒙古工业大学学报: 自然科学版, 2013, 32(2): 125-128