32nd IEEE Conference on Local Computer Networks

Discovering Network Topology of Large Multisubnet Ethernet Networks

Hassan Gobjuka, Yuri Breitbart
Department of Computer Science
Kent State University
Kent, OH 44242
{hgobjuka,yuri}@cs.kent.edu

Abstract—
In this paper we investigate the problem of finding the physical layer network topology of large, heterogeneous multisubnet Ethernet networks that may include uncooperative network elements. Our approach utilizes only generic MIB information [20] and does not require any hardware or software modifications of the underlying network elements. We propose here the $first$ $O(n^3)$ algorithm that guarantees discovering a topology that is compatible with the given set of input MIBs, provided that the input is complete. We prove the correctness of the algorithms and the necessary and sufficient conditions for the uniqueness of the restored topology. Finally, we demonstrate the application of the algorithm on several examples.

Keywords: Physical-Layer Topology Discovery, Ethernet LANs, SNMP MIB, Hubs.

I. Introduction

Many network management tasks (such as performance analysis, root cause analysis, and fault identification) critically depend on knowledge of network connectivity. There are, however, very few network tools that enable network managers to maintain an accurate view of network connections. Without such tools there is a high probability of making wrong decisions either on adjusting network performance or in identifying network faults and network traffic bottlenecks.

Despite its importance, especially at the LAN level (layer-2 of the ISO hierarchy), there are significant difficulties in obtaining topology information. Very few commercial network management platforms available on the market today offer general-purpose tools for automatic network topology discovery. Commercial tools that are currently on the market (such as HP's OpenView (openview.hp.com), IBM's Tivoli (tivoli.com), Cisco's Discovery Protocol (`www.cisco.com`), and Nortel's Discovery Protocol (`www.nortelnetworks.com`)) are based on proprietary information and often fail to capture many layer-2 connections in large multisubnet Ethernet networks. Topology discovery becomes even more complicated for networks that include uncooperative elements that either do not collect any MIB information [20] ($hubs$) or disallow access to their MIBs ($semihubs$) for security or other reasons. Unlike $hubs$, however, MIBs of accessible nodes contain addresses of $semihubs$. As far as we know there are no algorithms (or commercially available products) for topology discovery in networks that contain $hubs$ and/or $semihubs$.

Many algorithms for discovery layer-2 network topology use the Bridge and IP MIB information of local nodes. However, bridges and switches are involved in limited information exchanges. They intensely communicate with their neighbors only during the *spanning tree protocol* operation [18]. Consequently, the only useable MIB information maintained by switches and bridges is in the *Address Forwarding Table* (AFT) - the set of all MAC addresses that can receive packets from a port of a given node [10] using only layer-2 network elements. If AFTs are $complete$, (that is, they contain all nodes that can receive packets from a port without involving routers), then the procedures to derive network connections have been described in [5], [6]. However, these procedures fail to discover network topology for multisubnet networks that include $hubs$ and $semihubs$. In [3] we addressed the network topology for networks that included $hubs$ but not $semihubs$. The algorithm in [3] has $O(n^7)$ time complexity (where n is the number of internal network nodes) which renders it impractical.

In this paper we investigate the problem of finding the physical layer network topology of large, heterogeneous multisubnet Ethernet networks that may include uncooperative network elements (both $hubs$ and $semihubs$). Our algorithms utilizes only generic MIB information and does not require any hardware or software modifications of the underlying network elements. First, we define the notion of a potential connection between nodes. Then, we propose a topology algorithm that is based on the minimum number of potential connections between two nodes. We show that our algorithm has time complexity of $O(n^3)$ which is a significant improvement over the previous algorithm from [3]. The algorithm also guarantees the discovery of all possible topologies that are defined by a given set of input MIBs.

A. Related Work

Layer-2 network topology problems were addressed in the research community by several researchers [2], [4], [5], [6], [15], [21], [22]. For the set of complete AFTs the algorithms from [5], [6] find the layer-2 topology for multisubnet networks without hubs. In [5] the authors observed that for multisubnet networks the network topology may not be unique even for the set of $complete$ AFTs. In such a case finding the exact topology is not possible and algorithm from [6] generates some network fragments that can be uniquely determined. In [13] a criterion was introduced on the set of complete AFTs guaranteeing a unique topology for multisubnet networks. The authors in [3], [4] proposed algorithms to discover the topology of multisubnet networks that may contain hubs. However, both of these methods may not discover any topology if the given input AFTs defines a non-unique topology.

For networks with a single subnet, the set of incomplete AFTs may also define more than one topology [15]. Lowekamp *et.al.* [15] described a technique for inferring a connection between two nodes based on the AFT of these nodes. They also addressed the existence of hubs. However, their solution may fail to restore a network topology in some

0742-1303/07 $25.00 © 2007 IEEE
DOI 10.1109/LCN.2007.25

cases.

To discover the layer-2 topology for multisubnet networks with incomplete AFTs, [21] proposed a two-stage approach. At the first stage they try to complete incomplete AFTs by using AFT's extension rules. If AFTs can be successfully completed, then the topology discovery enters the second stage, where the algorithms from [5], [6] are used to generate the set of network connections. The authors of [21] asserted that their set of rules is complete. That is, an application of these rules always completes the set of incomplete AFTs. However, as the authors of [14] proved that the topology restoration problem with incomplete AFTs is NP-hard. Consequently, provided that $P \neq NP$, either the set of rules in [21] requires an exponential time to derive network connections, or the set of these rules is incomplete.

In [22] another method for deriving layer-2 topology that is based on a knowledge of a root of a spanning tree produced by the spanning tree protocol was proposed. However, unlike the Bridge information, the information on the spanning tree root is not regularly supplied by a majority of network vendors.

Recently Black *et. al.* [2] listed some problems with finding a layer-2 topology using Bridge MIB data. They proposed a new protocol to find a layer-2 topology without querying network MIB information. However, their approach requires placing custom designed network daemons on each host in the network, which some network managers might find objectionable.

B. Organization of the Paper

The rest of the paper is organized as follows. The next section describes the network model used in this paper. Section III contains our main result. Namely, it describes our topology discovery algorithm and proves its correctness. Section IV illustrates the application of our algorithm to two sets of AFTs. The first set defines a unique topology and the second set defines more than one topology. In both cases the algorithm finds all these topologies. Section V proves a necessary and sufficient condition to decide the uniqueness of the topology defined by the input set of AFTs. Finally, Section VI concludes the paper.

II. Background and Network Model

In this section we describe the network model. We also describe the source of information we use for topology discovery methods.

A. Formal Model

In this subsection, we describe the system model that we adopt for the topology discovery of multisubnet networks that include uncooperative network elements. The model is similar to the one defined in [6]. We refer to the network domain over which the network topology is defined as a *switching domain*. *Switching domain* S is defined as the maximal set of nodes such that there is a path between any two nodes in S that includes only nodes in S. Nodes in a switching domain that collect MIB information employ the *spanning tree protocol* [18] to determine a unique forwarding path between them. In many cases network manager may restrict access to the node MIBs for a variety of reasons. Hence we subdivide the set of all nodes in the network into three groups: (1) nodes whose MIBs are accessible by topology discovery process; (2) nodes whose MIBs are not accessible by the topology discovery process but these nodes may appear in the MIBs of the nodes of the first type; (3) nodes that neither have an accessible MIB nor appear in the MIBs of any other nodes. Nodes of the second and third type are called $semihubs$ and $hubs$, respectively. Clearly, when semihubs or hubs are directly connected with each other, then any topology discovery process can determine neither the order in which these nodes are connected nor the connection itself. Consequently, we impose an additional restriction on our model prohibiting any direct connections between hubs and/or semihubs. That is, we assume that between any two hubs or semihubs there is at least one node with an accessible MIB.

Thus, we model the network as an undirected tree $N =< V,\ E >$, where V is a set of all network nodes and each element of E represents a physical connection between two nodes. With each node a of network N, which is not a hub we associate the number of ports denoted by $p(a)$ and refer to port i of node a as a_i. In addition, with each node a we associate a subnet denoted by $subnet(a)$ to which a belongs. The node a is called terminal if and only if $p(a) = 1$.

The internal (i.e. non-leaf) nodes of the network represent layer-2 network elements (switches and bridges). Packets in a switching domain are forwarded from one node to another without involving a layer-3 device - router. However, a router is the point for the packet to enter a switching domain. Thus, a router uses a single interface to forward the packet within the switching domain. Hence, for the purposes of this paper we treat routers as hosts. Hosts and routers are represented as terminal (i.e. leaf) nodes of the network.

We say that two nodes a and b are connected by ports a_i and b_j if and only if there is a path in N between nodes a and b that starts at port a_i and ends at port b_j. The $length$ of the path is the number of edges in the path. If the length of the path is one, we say that ports a_i and b_j are $directly$ connected.

For each port a_i, a set of node addresses that have been learned on that port by the backward learning algorithm [23] is called the *address forwarding table (AFT)* [23] and denoted by $AFT(a_i)$. In our model we define $AFT(a_i)$ as the set of addresses of all ports such that:

1. $b_j \in AFT(a_i)$, where $subnet(b_j) = subnet(a_i)$ and there is a path from a_i to b_j; and
2. $b_j \in AFT(a_i)$, where $subnet(b_j) \neq subnet(a_i)$ and there is a port c_k such that $subnet(c_k) = subnet(a_i)$ and there is a path $c_k \dots b_j \dots a_i$.

Intuitively, it means that $AFT(a_i)$ includes all addresses of all ports that may appear as destination addresses of packets received on port a_i. When a packet is received by node a, the node forwards the packet along port a_i, if $AFT(a_i)$ contains the destination address in the packet. We say $AFT(a_i)$ is *complete* if $AFT(a_i)$ contains *all* the addresses from which packets can be received at a_i and does not include any node that cannot be reached from a_i. If $b \in AFT(a_i)$ then we say that port a_i *sees* node b and that node b is seen by port a_i.

Since a terminal node a contains a single port, it can see all nodes of the network from the a's subnet except itself. The set of all nodes that are seen from all ports of node a except port a_i is called *complementary to a_i address forwarding table* and is denoted by $CAFT(a_i)$. Since a network is a tree, no node can see itself on any of its ports. Thus, for any port a_i, $a \in CAFT(a_i)$. If node a is a terminal, then $CAFT(a_i)$ contains only a, since it is the only node that a cannot see.

port	AFT	$CAFT$
a_1	1, 3, x	2, 4, a
a_2	2, 4	1, 3, x, a
b_1	1, x	2, b
b_2	2	1, b, x

TABLE I

AFTs AND $CAFT$s FOR THE NETWORK IN FIG. 1(A)

port	RS	CRS
a_1	1, 3, x	2, 4, a, b
a_2	2, 4, b	1, 3, a, x
b_1	1, 3, 4, a, x	2, b
b_2	2	1, 3, 4, a, b,x

TABLE II

RSs AND CRSs FOR THE NETWORK IN FIG. 1(A)

Suppose that there is a path between ports a_i and b_j in N. Then $CAFT(a_i) \bigcap CAFT(b_j)$ is empty, since otherwise we would have at least one node c that can be reached by two different paths: one from some port a_k, $k \neq i$ and another from some port of b_l, $l \neq j$. Thus, if $CAFT(a_i) \bigcap CAFT(b_j)$ is not empty, then there is no path between ports a_i and b_j. If ports a_i and b_j are directly connected, then the intersection of $AFT(a_i)$ and $AFT(b_j)$ is empty.

As we stated earlier, an AFT of any port can be discovered by using SNMP to access the node's MIB information, provided that the node is not a semihub or hub. With each port a_i we associate nodes that are reachable in N by a path that starts at port a_i and denote this set by $RS(a_i)$. By definition, a does not belong to $RS(a_i)$. If a is a terminal (that is, a has a single port), then $RS(a_1)$contains all network nodes except itself. This follows from the assumption that the network is a tree.

For a single subnet network with complete AFTs, $RS(a_i) = AFT(a_i)$. For multisubnet networks, however, the set of nodes reachable from a_i is a superset of $AFT(a_i)$. It is well known that the MIB information of node a does not contain information about $RS(a_i)$. It does, however, contain $AFT(a_i)$. Thus, we initially approximate $RS(a_i)$ by $AFT(a_i)$. We define $V - RS(a_i)$ as a complement of set $RS(a_i)$ and denote it by $CRS(a_i)$. To illustrate the concepts of AFT, $CAFT$, RS, and CRS, consider the network depicted in 1(a). Node x is semihub and the blacked circle is a hub. Subnets are as follows: s_1=$\{1, 2, x\}$, s_2=$\{3, 4\}$. Each of nodes a and b comprises a subnet by itself. The AFTs and $CAFT$s are shown in Table I, and RSs and CRSs are shown in Table II.

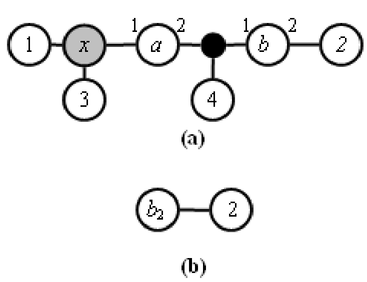

Fig. 1. Simple network with semihub and hub.

To model a direct connection between two ports a_i and b_j we introduce a notion of a *Potential Direct Connection (PDC)* as follows [13]:

1. Intersection of $AFT(a_i)$ and $AFT(b_j)$ is empty;
2. Union of $AFT(a_i)$ and $AFT(b_j)$ is a set of subnets;
3. Intersection of $CAFT(a_i)$ and $CAFT(b_j)$ is empty;
4. If a_i and b_j belong to the same subnet, then there is no c_k such that its subnet is different from a_i, and the union of $AFT(a_i)$ and $AFT(b_j)$ is equal to the union of $AFT(a_i)$ and $AFT(c_k)$; and, in addition, $AFT(a_i)$ and $AFT(c_k)$ satisfy the above three conditions.

In [13] we proved that the set of direct connections in N is a proper subset of the set of potential direct connections. We say that there is a *Potential Connection (PC)* between ports a_i and b_j if and only if the intersection of $CRS(a_i)$ and $CRS(b_j)$ is empty. Recall that initially $AFT(a_i)$ = $RS(a_i)$. Thus, if there is a PDC between ports a_i and b_j, then there is a PC between these ports as well. However, the existence of a PC between two ports does not imply that there is a PDC between these ports.

Finally, we define a notion of a *Potential Direct Connections Graph (PDCG)* for the given set of input AFTs as follows. Each port of N is a node in the $PDCG$. There is an edge between ports a_i and b_j in $PDCG$ if and only if there is a Potential Direct Connection between a_i and b_j. For example, in the network depicted in Figure 1(a), there is a PDC between ports b_2 and 2 and the following PCs: $(a_2,\ b_1)$ and $(a_1,\ b_2)$. The $PDCG$ of the network depicted in Figure 1(a) is shown in Figure 1(b).

III. Topology Discovery Algorithm

We start by introducing the extension process used by our topology discovery algorithm. Then we describe the algorithm and proof its correctness.

A. RS Extension Process

In this subsection we introduce two extension rules to extend RSs based on the content of initially given AFTs and based on already established connections between some nodes in network N. Systematically applying the extension rules defined here to the input set of AFTs results in the completion of every RS in a polynomial in the number of network nodes time.

We introduce the following two rules for the AFTs extension with other nodes based on the content of other AFTs.

Rule 1: If there is a connection between ports a_i and $b_j, a \neq b$, then $RS(a_i)$ = $RS(a_i) \bigcup CRS(b_j)$ and $RS(b_j)$ = $RS(b_j) \bigcup CRS(a_i)$.

The proof of this rule is trivial.

Rule 2: If there is a connection between ports a_i and $b_j, a \neq b$, and there is node c that belongs to subnet s_u, such that $c \in RS(a_k)$ for some port $k \neq i$ of node a and for any port b_l, $RS(b_l)$ does not contain any node from subnet s_u, then $RS(b_j)$ = $RS(b_j) \bigcup C$, where C contains all nodes of subnet s_u as well as all nodes that see any node from s_u.

Proof: The proof is by contradiction. Recall that the network topology is a tree and consequently, there is exactly one path between any two nodes in the network.

Suppose that there are two nodes c and d ($c \neq d$) from subnet s_u such that $c \in RS(a_k)$ and $d \in RS(a_q)$ and no node from subnet s_u belongs to $RS(b_l)$ for any port l of node b. Suppose that nodes c and d are added to $RS(b_j)$ and $RS(b_k)$ ($j \neq k$), respectively. However, since a_i and b_j are connected, node d must also appear in $RS(b_j)$. Contradiction. □

To illustrate, consider the network depicted in Figure 1(a). Since the $PDCG$ contains a single edge $< b_2, 2 >$, it results into a direct connection between b_2 and terminal node 2. Consequently, we conclude that there is a unique potential connection between nodes a and b and thus, this potential connection becomes a connection in the network. Applying rule 2 we conclude that $RS(b_1)$ contains nodes 1, 3, 4, x, and a, whereas $RS(a_2)$ contains nodes 2, 4, and b as it is shown in Table II.

Applying the extension rules systematically to a given set of AFTs eventually results in a complete set of RSs for each network port.

B. Algorithm

Now we are ready to describe our algorithm termed Multisubnet that discovers a topology of multisubnet network that may contain semihubs and hubs.

The algorithm Multisubnet consists of two phases. In Phase 1, the algorithm discovers all direct connections between any two MIB-enabled nodes. This is done using the algorithm from [13]. At the end of this phase, either the topology is completely discovered, or there is at least one semihub or hub between any two MIB-enabled nodes in the topology discovered so far. At the second phase, the algorithm selects two nodes that have the minimum number of *potential connections* between them. We prove that such a number cannot be more than 2. After selecting a potential connection, the RSs are extended using the extension rules described in III-A. The process is repeated until all RS's are complete.

As we prove in subsection III-C, the order of selecting potential connections does not affect the topology discovery process. Once the complete set of RSs is restored for each MIB-enabled node, the rest of the topology can be discovered using the method described in [6] for single subnet networks. The formal description of algorithm Multisubnet is given in Figure 2.

To derive the time complexity of the algorithm we observe that each phase of the algorithm requires a verification of the emptiness of the CRS's intersection. Since the complexity of the intersection is linear in the number of set elements, and the number of intersections to verify is on the order of n^2 (where n is the number of network nodes) we derive that the time complexity of the algorithm is $O(n^3)$.

C. Correctness Proof

We now prove that our algorithm always generates a topology that is compatible with the set of input AFTs. We first prove that there are at least two nodes a and b such that the number of potential connections between a and b is at most two. That allows us to start the topology restoration with these nodes. Next we prove that for nodes a and b that have exactly two potential connections and in the absence of nodes with a unique potential connection, selecting any potential connection between a and b leads to a correct topology.

Lemma III.1: If $AFT(a_i) \bigcap AFT(b_j) \neq \emptyset$, for any two ports i and j of nodes a and b, respectively, then there are at most two potential connections between nodes a and b.

Input: A set of complete AFTs
Output: A set $M \subseteq PDC$ of matchings

uniqueness = *unique*;
Phase#1
Generate Potential Direct Connection Graph $PDCG(N)$;
$M = \emptyset$; CC **=(**$\{a\}, \ldots, \{c\}$**) where** CC **is the set of connected components each of which at this stage is a single node;**
Do while ($PDCG(N)$ **is not empty)**
1. **If there is a terminal node** a_i **in** $PDCG(N)$ **and** $U(a_i,\ b_j)$ **is an edge in** $PDCG(N)$**, select** $U(a_i,\ b_j)$**;**
   (a) $M = M \cup \{(AFT(a_i),\ AFT(b_j))\}$;
   (b) **If** $a \in CC_l$ **and** $b \in CC_k$ **and** $k \neq l$**, then** $CC_l = CC_l \cup CC_k$**; delete** CC_K **from** CC**;**
   (c) **remove from** $PDC(N)$ **Graph all edges, whose endpoints are in** CC_l**;**
2. **If there are no terminal nodes in** $PDCG(N)$**, uniqueness =** *not unique*;
   (a) **select an arbitrary** PDC $(a_i,\ b_j)$**;**
   (b) **Goto 1a;**

Phase#2
Generate PC**s as follows:**
for any two ports a_i **and** b_j **do if** $CRS(a_i) \bigcap CRS(b_j) = \emptyset$**, then add** $< a_i,\ b_j >$ **to** PC**s;**
Do while (set of PC**s is not empty)**
1. **find two nodes** a **and** b **such that the number of potential connections between** a **and** b **in the set of** PC **is minimal;**
2. **if** $< a_i,\ b_j >$ **is not unique potential connection, uniqueness =** *not unique*;
3. $M = M \cup \{(RS(a_i),\ RS(b_j))\}$**; (break ties arbitrarily)**
4. **Delete** $< a_i,\ b_j >$ **from further considerations;**
5. $oldRS(a_i) = RS(a_i)$;
6. $oldRS(b_j) = RS(b_j)$;
7. **Extend** $RS(a_i)$ **and** $RS(b_j)$**;**
8. $oldRS(a_i) = RS(a_i)$;
9. $oldRS(b_j) = RS(b_j)$;
10. **Update the set of** PC**s;**

Return M **and uniqueness;**

Fig. 2. A formal description of the Multisubnet algorithm.

Proof: Suppose there is node u such that $u \in AFT(a_i) \bigcap AFT(b_j)$ and node u belongs to subnet s. Since no AFT contains all nodes that belong to one subnet, there is node v in the network such that v belongs to subnet s and $v \in AFT(a_k)$ and $v \in AFT(b_l)$, $i \neq k$, $j \neq l$. We consider the following cases:

1. *All nodes that belong to subnet s appear in at least three ports of either node a or b.* Suppose W.L.O.G that there is path between ports a_i and b_j in the network as depicted in Figure 3(a). Suppose there are nodes u, v and w from subnet s such that $u \in AFT(a_i)$, $v \in AFT(a_k)$ and $w \in AFT(a_m)$. Since the topology is tree, both nodes v and m must belong to $AFT(b_j)$. Also, since u belongs to subnet s, it must appear in the AFTs of nodes a and b. Suppose $u \in AFT(b_l)$ and $u \in AFT(a_i)$, then, $\{v, w\} \in CAFT(a_i) \bigcap CAFT(b_l)$, $u \in CAFT(b_j) \bigcap CAFT(a_p)$, $p \neq i$, and $\{u, v, w\} \in CAFT(a_p) \bigcap CAFT(b_q)$, $p \neq i \neq k \neq m$, $j \neq l \neq q$. Consequently, $(a_k,\ b_j)$ is a unique potential connection between nodes a and b.

2. *All nodes that belong to subnet s appear in at most two ports of both nodes u and v.* Suppose all nodes that belong to subnet s appear in two AFTs of nodes a and b, say $AFT(a_i)$ and $AFT(a_k)$ of node a and $AFT(b_j)$ and $AFT(b_l)$ of node b such that $u \in AFT(a_i) \bigcap AFT(b_j)$ and $v \in AFT(a_k) \bigcap AFT(b_l)$ as depicted in Figure 3(b). Then, there are only two potential connections between nodes a and b which are $(a_k,\ b_j)$ and $(a_i,\ b_l)$ be-

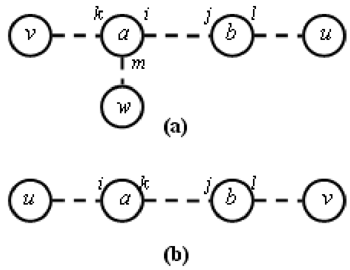

Fig. 3. Figure for the proof of lemma III.1.

cause $u \in CAFT(a_k) \bigcap CAFT(b_p)$, $p \neq l$, and $v \in CAFT(b_j) \bigcap CAFT(a_q)$, $q \neq k$.
Thus, the lemma is proven. □.

Lemma III.2: If $RS(a_i) \bigcap RS(b_j) \neq \emptyset$, for any two ports i and j of nodes a and b, respectively, but the initially given AFTs of nodes a and b didn't intersect, for any two ports p and q of nodes a and b, then (a_i, b_j) is a unique potential connection between nodes a and b.

To illustrate how such situation can occur, consider the network depicted in Figure 4. Nodes u and v belong to subnet s_1 and nodes p and q belong to subnet s_2. Each of nodes a, b, and c comprises a subnet by itself. Observe that AFTs of nodes a and b don't intersect for any two ports i and j of nodes a and b, respectively. However, AFTs of node a intersect with AFTs of node c. Also, AFTs of node b intersect with AFTs of node c. Once the connection between nodes a and c and between nodes b and c are discovered and AFTs extended, the intersection of RSs of nodes a and b will no longer be empty.

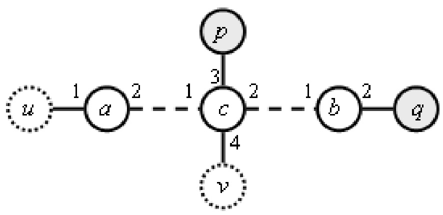

Fig. 4. Example for lemma III.2.

Now we proof the correctness of the lemma III.2.
Proof: Suppose that there is subnet s_1 in the network such that for each node u from s_1, $u \in RS(a_i) \bigcap RS(b_j)$. Note that since subnet s_1 belongs to one port of nodes a and b, s_1 must be added to $RS(a_i)$ and $RS(b_j)$ during the extension procedure. Thus, there is node c in the network such that there is path (c_p,a_i) and (c_q,b_j). Node c either belongs to subnet s_1 or it sees nodes from s_1 on at least two ports. Furthermore, there are nodes u and v from subnet s_2 and nodes w and z from subnet s_3 and nodes from s_2 are seen by AFTs of nodes a and c. Also, nodes from s_3 are seen by AFTs of nodes b and c.

If node a, W.L.O.G., is on the path between nodes c and b as depicted in Figure 5(a), then nodes u and v were seen by two AFTs of a. Consequently, it contradicts our assumption that AFTs of nodes a and b were not intersecting.

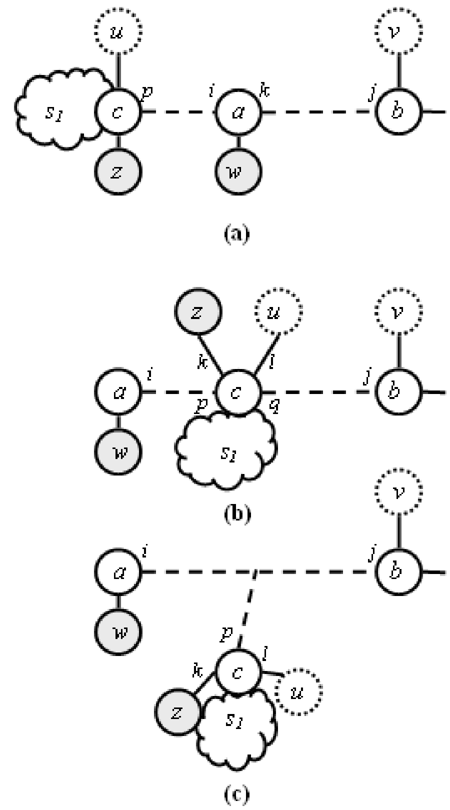

Fig. 5. Figure for the proof of lemma III.1.

If node c is on the path between nodes a and b in the network, then either:

- node c is connected to nodes a and b through ports c_p and c_q, $p \neq q$, respectively, as depicted in Figure 5(b), or
- node c is connected to nodes a and b through port c_p, as depicted in Figure 5(c).

We first consider the first case.

When the potential connection (a_i, c_p) is selected and extension process was applied, by applying rule 2, all nodes from subnet s_1, s_2 as well as all nodes that see any node from s_1 and s_2 were added to $RS(a_i)$. Thus, all nodes from s_1 and nodes u, v, b were added to $RS(a_i)$.

Similarly, connection (c_q,b_j) was selected and all nodes from subnet s_1 and nodes w, z and a were added to $RS(b_j)$.

Since $a \in RS(b_j)$ and $b \in RS(a_i)$, there is a unique potential connection between nodes a and b.

Correctness of the second case can be proven by using same argument as for the first case.

Thus, the lemma is proven. □

Lemma III.3: If there is no unique potential connection between nodes a and b, $a \neq b$, in the network, then there are no two nodes u and v such that $\{u, v\} \in AFT(a_i)$, and $u \in AFT(b_j)$ and $v \in AFT(b_k)$, $j \neq k$.

Proof: Suppose the network contains two distinct nodes a and b and two other nodes u and v and $\{u, v\} \in AFT(a_i)$, $u \in AFT(b_j)$, $v \in AFT(b_k)$, $j \neq k$ as depicted in Figure 6(a). Since each node either sees all nodes belonging to one subnet or none of them, there is node w such that nodes u, v and w belong to the same subnet and $w \in AFT(b_k)$.

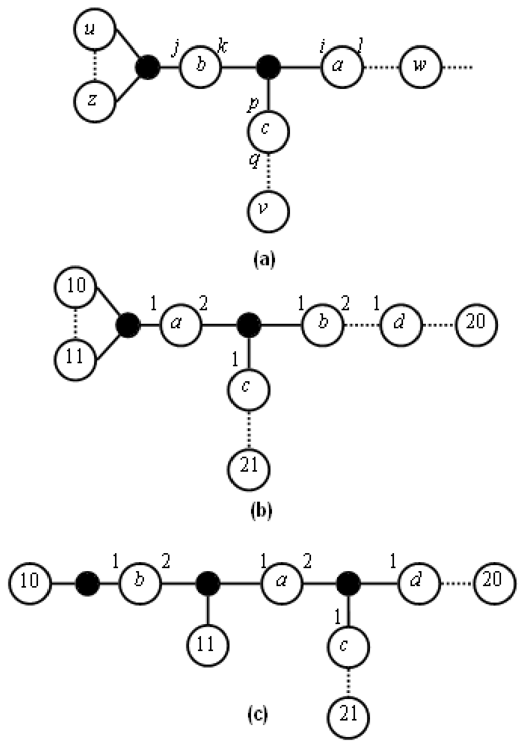

Fig. 6. Figure for the proof of lemmas III.3 and III.4.

Thus, the following holds: $w \in CAFT(a_i) \bigcap CAFT(b_y)$, $y \neq k$, $u \in CAFT(a_x) \bigcap CAFT(b_k)$, $x \neq i$, and $v \in CAFT(a_x) \bigcap CAFT(b_y)$, $x \neq i$, $y \neq k$. Consequently, the connection $(a_i,\ b_k)$ is unique.Thus, the lemmas is proven. $\square$

Lemma III.4: If there is no unique potential connection between any two nodes in the network and there are at most two potential connections between nodes a and b, then selecting one of these potential connections results in a valid network topology that is compatible with the given set of AFTs.

Proof: Suppose port b_1 is not connected directly to the nodes in $AFT(b_1)$ as depicted in Figure 6(b). We proof that connecting port b_1 to the nodes in $AFT(b_1)$ will result in a valid topology.

From lemma III.3, we know that either $AFT(b_i) \bigcap AFT(c_j)$ = $\emptyset$, for any ports i and j of nodes b and c in the network, or $AFT(a_k) \bigcap AFT(c_j) = \emptyset$ for any ports k and j of nodes a and c in the network. Thus, we have the following cases:

1. $AFT(b_i) \bigcap AFT(c_j) = \emptyset$ *for any ports i and j of nodes b and c, respectively.* In this case, $AFT(b_1) \subset AFT(a_1)$. Let F be the set of nodes that belong to both $AFT(a_1)$ and $AFT(b_1)$ (i.e. $AFT(a_1) \bigcap AFT(b_1) = F$). Since there are exactly two potential connections between nodes a and b (by lemma III.1), connecting port b_1 to the nodes in F will result in a unique connection between ports b_2 and a_1. By extending the AFTs of nodes a and b, nodes $c, 11, 21$ and all other nodes that see any of these nodes will be added to $AFT(b_2)$. Finally, if port b_2 was connected to port d_1 in the network, then $(a_2,\ d_1)$ is a valid potential connection since there is path between a_2 and d_1 in the network. Thus, the topology depicted below will be restored.

2. $AFT(a_i) \bigcap AFT(c_j) = \emptyset$ *for any ports i and j of nodes a and c, respectively.* In this case, $AFT(a_1) \subset AFT(b_1)$. Let set F be the set of nodes that belong to both $AFT(a_1)$ and $AFT(b_1)$ (i.e. $AFT(a_1) \bigcap AFT(b_1) = F$). Since there are exactly two potential connections between nodes a and b, connecting port b_1 to the nodes in F will result in a unique connection between ports b_2 and a_1. By extending the AFTs of nodes a and b, the topology depicted in Figure 6(c) will be restored.

Thus, the lemma is proven. $\square$

The correctness of the algorithm follows from lemmas III.1, III.2, III.3, and III.4.

port	AFT	port	AFT
a_1	r, s	d_1	s, u
a_2	q, t	d_2	t, v
b_1	r, s, u, x	e_1	v, y
b_2	q, t, v, y	e_2	x
c_1	s, u, x	e_3	u
c_2	t, v, y		

TABLE III

AFTS FOR THE MULTISUBNET NETWORK IN FIG. 7

IV. COMPREHENSIVE EXAMPLES

In this section we consider two examples. Our first example demonstrates the algorithm for the network whose set of AFTs uniquely define the topology. Our second example demonstrates the set of AFTs that do not define a unique topology and we show that in such a case, our algorithm finds all network topologies.

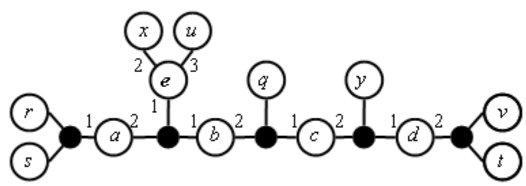

Fig. 7. Example of network with hubs from [3].

Example 1: To illustrate the application of the algorithm, and to show the difference between our approach and the one proposed in [3], we consider the network depicted in Figure 7, which is obtained from [3]. Subnets are as follows: $s_1 = \{s, t\}$, $s_2 = \{x, y\}$, $s_3 = \{u, v\}$, and $s_4 = \{r, q\}$. Each non-terminal node comprises a subnet by itself. The AFTs of nodes a, b, c, d and e are shown in Table III and blacked circles represent hubs.

There is a PDC between ports e_2 and x and between ports e_3 and u. Consequently, these direct connections will be discovered. Since no more potential connections can be discovered by applying phase 1, phase 2 will create the following set of PCs: (a_1, b_2), (a_2, b_1), (a_1, c_2), (a_2, c_1), (a_1, d_2), (a_2, d_1), (a_1, e_1), (a_2, e_1), (b_1, c_2), (b_2, c_1), (b_1, d_2), (b_2, d_1), (b_1, e_1), (c_1, d_2), (c_2, d_1), (c_1, e_1), (d_1, e_1).

port	RS
a_1	r, s
a_2	q, t
b_1	e, s, r, u, x
b_2	t, q, v, y
c_1	e, s, u, x
c_2	t, v, y
d_1	b, c, e, s, u, x, y
d_2	v, t
e_1	a, b, c, d, q, r, s, t, v, y
e_2	x
e_3	u

TABLE IV
RESULTING RSS AFTER DISCOVERING PC (e_1, b_1) (e_1, c_1), (e_1, d_1), AND APPLYING THE EXTENSION RULES.

port	RS
a_1	r, s
a_2	q, t
b_1	e, s, r, u, x
b_2	d, t, q, v, y
c_1	e, s, u, x
c_2	d, t, v, y
d_1	a, b, c, e, q, r, s, u, x, y
d_2	v, t
e_1	a, b, c, d, q, r, s, t, v, y
e_2	x
e_3	u

TABLE V
RESULTING RSS AFTER DISCOVERING PCS (b_2, d_1) AND (c_2, d_1).

port	RS
a_1	r, s
a_2	b, c, e, q, t, u, v, x, y
b_1	e, s, r, u, x
b_2	d, t, q, v, y
c_1	e, s, u, x
c_2	d, t, v, y
d_1	a, b, c, e, q, r, s, u, x, y
d_2	v, t
e_1	a, b, c, d, q, r, s, t, v, y
e_2	x
e_3	u

TABLE VI
RESULTING RSS AFTER DISCOVERING PC (a_2, d_1).

port	AFT	port	AFT
a_1	10, 20	c_1	31, 40
a_2	11, 21	c_2	30, 41, x
b_1	11, 30, x	d_1	21, 41
b_2	10, 31	d_2	20, 40

TABLE VII
THE INITIAL AFTS FOR THE MULTISUBNET NETWORK IN FIG. 8

We note that there are the following unique potential connections: (b_1, e_1), (c_1, e_1) and (d_1, e_1). Consequently, we extend $RS(b_1)$, $RS(c_1)$ $RS(d_1)$ and $RS(e_1)$ as follows. Utilizing extension rule 1, we add e to $RS(b_1)$ and nodes b, q, t, v, y to $RS(e_1)$. Also, since none of the nodes from subnet s_4 (i.e. nodes q and r) is seen by any port of e and both are seen by AFTs of node b, by utilizing extension rule 2, all nodes from subnet s_4 as well as all nodes that see any node from that subnet will be added to $RS(e_1)$. Thus, we add nodes q, r and a to $RS(e_1)$. We also extend $RS(e_1)$ and $RS(c_1)$ as follows. We add node e to $RS(c_1)$ and nodes c, s and t to $RS(e_1)$. Finally, we extend RSs $RS(d_1)$ and $RS(e_1)$ and obtain the new set showed in Table IV.

After that, we note that the potential connections (b_2, d_1) and (c_2, d_1) are unique. RSs will be extended and the new set is shown in Table V. At the next step, we observe that the PC (a_2, d_1) is unique. We extend RSs to obtain the ones shown in Table VI.

At this stage, it can be observed that there is a unique PC between every two nodes. Consequently, complete RSs will be obtained and the unique topology is discovered.

Example 2: Consider the network depicted in Figure 8(a). In this example, node x is semihubs and subnets are as follows: $s_1 = \{10, 11\}$, $s_2 = \{20, 21\}$, $s_3 = \{30, 31, x\}$, $s_4 = \{40, 41\}$. Each non-terminal nodes comprises a subnet by itself. The AFTs of nodes a, b, c and d are shown in Table VII. Node x is a semihub and blacked circles represent hubs.

We note that there is no PDC and the PCs are as follows: (a_1, b_1), (a_1, c_1), (a_1, c_2), (a_1, d_1), (a_2, b_2), (a_2, c_1), (a_2, c_2), (a_2, d_2), (b_1, c_1), (b_1, d_1), (b_1, d_2), (b_2, c_2), (b_2, d_1), (b_2, d_2), (c_1, d_1), (c_2, d_2). Observe that, for any two nodes in the network, there is no unique PC between them. However, there are two PCs between nodes a and b, a and d, b and c, and between nodes c and d.

Suppose the algorithm selects (a_1, b_1) as a PC and RSs are extended. The updated set of PCs is as follows: (a_1, c_1), (a_1, d_1), (a_2, d_2), (b_1, c_1), (b_1, d_1), (b_1, d_2), (c_1, d_1), (c_2, d_2). We observe that, there is unique PC between nodes a and c, b and c, and between nodes b and d. Extending RSs at this stage will result in complete RSs for all nodes except semihubs. Thus, the topology depicted in Figure 8(b) will be discovered. It is worth to mention here that the techniques from [3], [4], [15] cannot discover any topology of this set of AFTs.

V. UNIQUENESS CRITERION

In this section we discuss the uniqueness criteria for multisubnet networks with $hubs$ and $semihubs$. Let θ be an input set of complete AFTs. If at every step of the process the selected PC is unique, then the RSs are extended uniquely, and thus, the topology is uniquely restored. However, if, at some step of the process the number of PC is equal to two, then the RS's cannot be uniquely extended and, consequently, the topology is not unique. The following theorem holds:

Theorem V.1: Let θ be a set of complete AFTs for a multiple subnet network N. The topology restored from θ is unique if and only if:

1. The phase 1 of the algorithm Multisubnet doesn't make any random choices, and

2. At each iteration of phase 2, there are two nodes a and b in N such that there is unique potential connection between them.

Proof The proof of the theorem follows from [13] and the proof of III.4. □

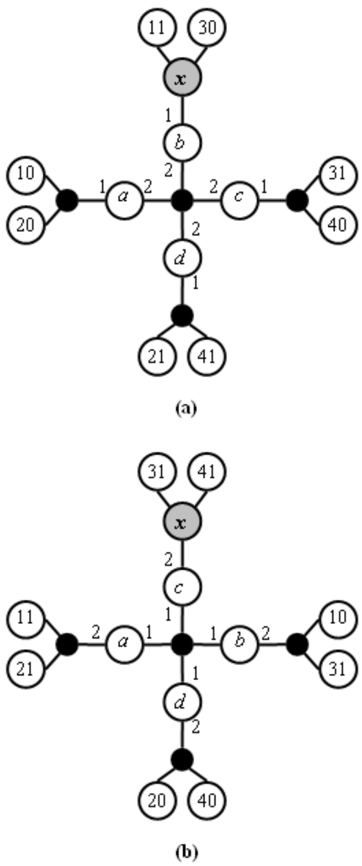

Fig. 8. Two networks that can be defined with the same of AFTs.

VI. Conclusions

Automatic discovery of physical topology information plays a crucial role in enhancing the manageability of modern IP networks. In this paper, we have proposed a simple and practical algorithm for discovering the physical topology of a large, heterogeneous multisubnet Ethernet network that contains $hubs$ and $semihubs$. Our algorithm relies on standard SNMP MIB information that is widely supported in modern IP networks. Our algorithm also determines whether the given set of input AFTs defines a unique topology.

References

[1] A. Bierman and K. Jones, *Physical Topology MIB*. http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2922.txt, September 2000.

[2] R. Black, A. Donnelly, C. Fournet *Ethernet Topology Discovery without Network Assistance*. Proceedings of the 12th IEEEInternational Conference on Network Protocols (ICNP'04), 2004

[3] Y. Bejerano, Y. Breitbart, M. Garofalakis, R. Rastogi, Physical Topology Discovery for Large Multi-Subnet Networks Proceedings of INFOCOM 2003, 2003.

[4] Y. Bejerano A Simple and Efficient Topology Discovery Scheme for Large Multisubnet Networks Proceedings of INFOCOM 2006, 2006.

[5] Y. Breitbart, M. Garofalakis, C. Martin, R. Rastogi, S. Seshadri, and A. Silberschatz Topology Discovery in Heterogeneous IP Networks, Proceedings of INFOCOM 2000, 2000.

[6] Y. Breitbart, M.Garofalakis, B. Jai, C. Martin, R. Rastogi, and A. Silberschatz *Topology Discovery in Heterogeneous IP Networks: The Net-Inventory System*, IEEE/ACM Transactions on Networking, v. 12, 3, June 2004

[7] B. Boardman Layer 2 Layout: Layer 2 Discovery Digs Deep Network and System Management Workshop, Nov. 2003

[8] T. Bu, N. Duffield, F. Lo Presti and D. Towsley, "Network Tomography on General Topology", in *Proc. of ACM SIGMETRICS'2002*, June 2002.

[9] R. Caceres, N.G. Duffield, J. Horowitz, D. Towsley and T. Bu, Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation, in *Proc. of IEEE INFOCOM'99*, Mar. 1999.

[10] E. Decker, P. Langille, A. Rijsinghani, and K. McCloghrie, *Definitions of Managed Objects for Bridges*, Internet RFC-1493 (available from `http://www.ietf.org/rfc/`), July 1993.

[11] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completness, W.H. Freeman and Company, New York, 1979.

[12] J. Case, M. Fedor, M. Schoffstall, and J. Davin, A Simple Network Management Protocol (SNMP), May 1990, Internet RFC-1157 (available from http://www.ietf.org/rfc/).

[13] H. Gobjuka, Y. Breitbart, Characterization of Layer-2 Unique Topologies in Multisubnet Local Networks, Proceedings of The 31st IEEE Conference on Local Computer Networks (LCN), Tampa, Florida, 2006.

[14] H. Gobjuka, Y. Breitbart, Finding Ethernet-Type Network Topology is Not Easy, Technical report, Department of Computer Science, Kent State University, August 2006.

[15] B. Lowekamp, David R. O'Hallaron, and T. Gross, *Topology Discovery For Large Ethernet Networks* Proceedings of ACM SIGCOMM 2001, San Diego, CA, 2001

[16] K. McCloghrie and M. Rose, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*, Internet RFC-1213 (available from `http://www.ietf.org/rfc/`), Mar. 1991.

[17] A. Medina, et. al. *http://www.cs.bu.edu/brite*. Boston University, 2002

[18] R. Perlman *Interconnections, Second Edition*, Addison-Wesley, 2000 (Third Edition).

[19] M. Rabbat, R. Nowak and M. Coates, Multiple Source, Multiple Destination Network Tomography, in *Proc. of IEEE INFOCOM'2004*, Mar., 2004.

[20] W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, Addison-Wesley Longman, Inc., 1999, (Third Edition).

[21] Y. Sun, Z. Wu, Z. Shi *The Physical Topology Discovery for Switched Ethernet Base on Connection Reasoning* , Proceedings of ISCIT, 2005pp. 42-45

[22] Y. Sun, Z. Shi, Z. Wu, *A Discovery Algorithm for Physical Topology in Switched Networks*, Proceedings of the IEEE Conference on Local Computer Networks, pp. 311-317, 2005

[23] A. Tanenbaum, *Computer Networks, Fourth Edition*, Prentice Hall Ptr., 2002.