Physical Topology Discovery in Large Ethernet Networks

Myunghee Son, Yongjoon Lee, Cheolsig Pyoo, Byungcheol Kim, Jaeyong Lee
Group of RFID/USN Research, Department of Information and Communication Engineering
Electronics and Telecommunications Research Institute, Chungnam National Univ.
Daejeon
Korea
http://www.etri.re.kr

Abstract: - Recent developments in Ethernet technology are pushing Ethernet from the local area network environment to metropolitan and wide area network environments. Automatic discovery of physical topology plays a crucial role in enhancing the manageability of modern Large Ethernet networks. Despite the importance of the problem, earlier research and commercial network management tools have typically concentrated on either discovering logical topology, or proprietary solutions targeting specific product families. Recent works [1], [2], [3] has demonstrated that physical topology can be determined using standard SNMP MIBs, but these algorithms depend on AFT entries and can find only spanning tree paths in the Ethernet network. The previous work [1] requires that AFT entries are complete, however it is very critical assumption in the true Ethernet network. In this paper, we have discarded the critical assumption because we divide Large Ethernet networks into bridged networks and host networks in order to overcome another AFT’s limitation, an enormous size of AFT in core bridges [1], [3]. Most of all, our algorithm is the first solution to be able to discover real physical topology including inactive interfaces eliminated by the spanning tree protocol in Large Ether networks.

Key-Words: - Physical Topology Discovery, Ethernet, SNMP MIB, Bridges, Hubs, Spanning Tree Protocol

1 Introduction

Metro Ethernet services [4] are now offered by a wide range of service providers. Some providers have extended Ethernet services beyond the metropolitan area network (MAN) and the wide area network (WAN). Thousands of subscribers already use Ethernet services and their numbers are growing rapidly. These subscribers have been attracted by the benefits of Ethernet services, including: Ease of use, Cost Effectiveness, and Flexibility. Now there is the characterization of pushing Ethernet from the local area network (LAN) environment to MAN and WAN environments. In this paper, we define these kinds of Ethernet networks as Large Ethernet networks which consist of LAN, MAN, and WAN. For Large Ethernet network providers, which are mainly the existing telephone network providers and service providers, the factors include: The existing technologies for data transport have limitations with respect to scalability for bandwidth and network management complexity. Actually most system operators do not manage Ethernet devices, which is not critical in the LAN environment, however, in the Large Ethernet network environment it is inevitable issue. In spite of the importance of the problem, earlier research and commercial network management tools have typically concentrated on either discovering logical (i.e., Layer-3) topology, which implies that the connectivity of all Layer-2 devices is ignored, or proprietary solutions targeting specific product families. The IETF has acknowledged the importance of this issue by designating a “physical topology” SNMP Management Information Base (MIB) [5], but the proposal merely reserves a portion of the MIB space without defining any protocol or algorithm for obtaining the topology information. More over, the recent works related to physical topology discovery in the Ethernet network had a serial common problem which can not discover true physical topology, instead of it they found only spanning tree paths. In reference to related work we addressed in the section 2. For physical topology discover, including multiple redundant paths in Large Ethernet network, we suggest novel and practical algorithms which can perform automatic discovery while minimizing overhead of processing an enormous number of Layer-2 address forwarding table (AFT) entries in core bridges. Our algorithm used standard Interface MIB [6] and Bridge MIB [7] and we can discover physical topology whenever it is needed.

We begin by describing related work and our contribution. Section 3 reviews necessary background information. Section 4 describes our topology discovery algorithm, which derives locating edge bridges 1, neighbor bridges, and hosts with spanning tree protocol (STP) information and Layer-2 AFT. In Section 5, we

1 To be connected to some hosts or routers
present our experiments including implementation and test results and discuss how our solution can be extended to deal with rapid STP (RSTP) and multiple STP (MSTP). Finally, Section 6 concludes the paper.

2 Related Work

Moving Layer-3 communication to Layer-2 communication, a number of vendors have recently developed proprietary tools and protocols for discovering physical network connectivity in the Large Ethernet network. In recent work related physical topology discovery for Large Ethernet networks, Breitbart et al. [1] proposed an algorithm that relies solely on standard AFT information collected in SNMP MIBs to discover the physical topology of heterogeneous networks comprising bridges organized in multiple subnets. Unfortunately, the algorithm assumes that AFT information is available from every node in the underlying network and, thus, cannot cope with hubs or uncooperative switches 2. In a follow-up paper, Lowekamp et al. [2] suggested techniques for inferring network-element connectivity using incomplete AFT information and also discussed how to handle dumb, uncooperative elements. In the latest paper, Bejerano et al. [3] suggest more advanced algorithm using skeleton path for discovering physical topology in Large Multi-Subnet networks. The common features of the previous works are that they only use AFT entries. Because of that, they can only find Layer-2 spanning tree paths excluding multiple redundant paths.

2.1 Our contributions

The practicality of our algorithms stems from the fact that they rely solely on standard information routinely collected in the SNMP MIBs [8] of bridges and they require no modifications to the operating system software running on bridges or hosts. Three previous works [1], [2], [3] also have the same advantage as above. However there were a lot of problems about their researches as follows: First, AFT entries typically employ an aging mechanism to evict infrequent source MAC addresses from the AFT. In case of aging-out, topology discovery is not feasible. Second, In the Large Ethernet network the size of a AFT is from tens of Kbytes to scores of Mbytes. Moreover the core bridge must be have hundreds of Mbytes. It will take up very long time to obtain all AFT entries using standard SNMP MIB and be difficult to obtain complete information. Third, All data traffic must be bi-directional but real data traffic doesn’t meet the requirements. However Lemma III.1 presented by Breitbart et al. [1] always needs bi-directional complete AFT entries. Fourth, After a topology change happens the bridge must flush all AFT entries. In this case it is impossible to discover topology in Ethernet network. Fifth, they didn’t find edges between interfaces that are not active (i.e., are eliminated by the STP). The topology found like this, is not physical graph but Layer-2 spanning tree paths excluding multiple redundant paths. In order to overcome these problems we developed novel, practical algorithmic solutions which can discover accurate physical topology in a Large Ethernet network. The key idea of our algorithm is as follows: First, To find edge bridges to discover some host connectivity by eliminating aging-out influence and avoiding an enormous size of AFT entries in core bridges. Second, to discover edges between interfaces that are eliminated by the STP. The reason that we were able to find multiple redundant paths is STP BPDUUs that are received on inactive port states, such as blocking and listening. The best contribution of this paper is that we find true physical topology, not a tree, of the Ethernet networks, where loop structures present.

3 Our Algorithm

The goal of our proposed algorithm was to find a lot of redundant physical paths in the Large Ethernet network. In this section, we describe our topology discovery algorithm, which derives locating edge bridges, designated bridges3, and hosts with minimal amount of data required by the Bridge MIB [7]. A Large Ethernet network containing the nodes N can be divided into a set of bridges, B, a set of edge bridges, Q, and a set of hosts, H. Figure 2 shows a simple and representative Large Ethernet network composed LAN, MAN, and WAN. Figure 2-a shows a sample Large Ethernet network and the result of STP is shown in Fig. 2-b, which multiple redundant paths are eliminated owing to loops.

Fig. 2-a

Fig. 2-b

Fig. 1: A Large Ethernet network: B1, B2, B3, B4, and B5 are bridges. S, T, and U are hosts. V is a router. H is a hub. A

2 The terms “switch” and “bridge” can be used interchangeably; we will primarily use “bridge” in the remainder of this paper.

3 These are neighbor bridges connected to a bridge.
network composed of B1, B2, and B3 with some 1Gbps links are considered as MAN or WAN and the others are considered as LAN connections.

Realistically, there are many redundant paths in Large Ethernet network for cost effective. And spanning tree builds an ordered network from a root bridge to designated bridges. The number of each bridge shown in Fig. 2 represents bridge priority and the small number has better priority. That is, a bridge B1 has better priority than the other bridge B2. After spanning tree calculation, the logical network is presented as in Fig. 2-b (i.e., spanning tree eliminating multiple redundant paths for loops). In Fig. 2-a, a bridge B5 is an edge bridge and it was connected to hosts, T and U, through hub and it belongs to a LAN segment. B4 and S are connected to a point-to-point LAN segment. A root bridge, B1, is connected to a router V, which is identical to hosts for the purpose of the Ethernet bridging algorithm. The bridged networks represent MAN or WAN. In other words they are expressed as network with the range from one edge bridge to the other edge bridge. Therefore some edge bridges, i.e., B1, B4, and B5, are boundaries in bridged networks and host networks4.

**DEFINITION 3.1. (EDGE BRIDGE)**

An edge bridge is a bridge that contains at least an interface connected to an interface of any hosts or any routers. In other words, it is ingress or egress bridge to or from bridged networks.

**PROPERTY 3.1.**

It is impossible for the number of active ports on a bridge to be less than the number of STP enabled ports.

Remark: If we subtract the number of STP enabled ports from the number of active ports, the number of interfaces connected to hosts remains. We don’t need to run STP on those interfaces, because the interfaces are not connected to bridged networks with loops.

The first step in discovering a network’s topology is determining some edge bridges’ locations, and decide whether AFTs are necessary in such locations or not. The second step is discovering designated bridges’ locations and the last step is finding locations of hosts connected to some edge bridges. More specifically, we can model the Large Ethernet network as an undirected graph $G = (V, E)$, where each node in V represents a network element and each edge in E represents a physical connection between two bridge interfaces. Table 1 summarized the key notation used throughout the paper with a brief description of its semantics. Additional notation will be introduced when necessary.

**Table 1. Notation.**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G = (V, E)$</td>
<td>The Large Ethernet network graph</td>
</tr>
<tr>
<td>$b_i$</td>
<td>Bridge $b_i \in B = {b_1, b_2, ..., b_n}$</td>
</tr>
<tr>
<td>$D_i$</td>
<td>Set of designated bridges connected to $b_i$, $D_i \subseteq D$</td>
</tr>
<tr>
<td>$P_i$</td>
<td>Set of active ports on a bridge $b_i$, $P_i \subseteq P$</td>
</tr>
<tr>
<td>$T_i$</td>
<td>Set of STP enabled ports of a bridge $b_i$, $T_i \subseteq T$</td>
</tr>
<tr>
<td>$H_i$</td>
<td>Set of AFT entries in an edge bridge $b_i$, $H_i \subseteq H$</td>
</tr>
<tr>
<td>$Q$</td>
<td>Set of edge bridges</td>
</tr>
<tr>
<td>$H_{ij}$</td>
<td>Set of hosts connected to $j$th interface of an edge bridge $b_i$</td>
</tr>
<tr>
<td>$D_i^j$</td>
<td>$j$th element of $D_i$</td>
</tr>
<tr>
<td>$e = (v, v')$</td>
<td>An edge is an element in the set of $E$</td>
</tr>
<tr>
<td>$n(S)$</td>
<td>The number of elements in a set of $S$</td>
</tr>
</tbody>
</table>

**LEMMA 3.1 (EDGE BRIDGE SELECTION).**

Let $b_i$ be a bridge in Large Ethernet Networks and $Q$ be a set of edge bridges. If $n(P_i) > n(T_i)$, then $b_i \in Q$.

**Proof:** Suppose to the contrary that $b_i$ is not a element of $Q$. Then, the number of active ports on a bridge $b_i$ must be equal to the number of STP enabled ports on a bridge $b_i$. The case that the number of active ports on a bridge is less than the number of STP enabled ports violates the property 3.1.

Our physical topology discovery algorithm is depicted in Fig. 3, which gives the pseudo-code for finding edge bridges based on a Definition 3.1, Property 3.1, and a Lemma 3.1.

```
procedure TopologyDiscovery(bridgeSet)
/* bridgeSet B = \{b_1, b_2, ..., b_n\} */
begin
    edgebridgeSet Q ← φ
    designatedbridgeSet D_i ← φ
    enableportSet P_i ← φ
    stpportSet T_i ← φ
    for each bridge $b_i \in B$ do{
        $P_i$ ← getFMI B(b_i)
        $T_i$ ← getSTPMIB(b_i)
        $D_i$ ← getSTPMIB(b_i)
    }
    for each bridge $b_i \in B$ do{
        if($n(P_i) > n(T_i)$)
            $Q$ ← $Q \cup \{b_i\}$
    }
FindBridgeGraph( B , D )
```

4 They represent the network from edge bridges to hosts.
Fig. 3. Our Physical Topology Discovery Algorithm.

The main goal in our algorithm, as shown in Fig. 3, is to collect a set of edge bridges to minimize for an SNMP manager to get a dot1dTpFdbTable and a dot1dStaticTable in the BridgeMIB [7] and a set of designated bridges to draw a bridge graph. Both getIFMIB and getSTMIB are metaphysical terminologies and are the subsets of BridgeMIB. They must include active ports set, STP enabled ports set, and designated bridges set for each bridge.

### 3.1 Finding Edge Bridges

The first task faced by our algorithm is to find edge bridges in Large Ethernet Network. It is easy to find an edge bridge because it has interfaces which are not running STP. Therefore, if it follows Lemma 3.1, it becomes an edge bridge and is inserted to the set of edge bridges, Q.

This procedure offers for every core bridges, which are not edge bridges in bridged networks, not to obtain AFT entries for discovering Ethernet topology. Most of all, the reason why we find edge bridges is for core bridges not to get an enormous size of AFT entries and for only edge bridges to get AFT entries.

### 3.2 Finding Connections in bridged network

The main goal of this procedure is to discover physical graph in bridged network and we had sufficient information to perform it through TopologyDiscovery procedure shown in Fig. 3. We used dot1dStpPortDesignatedBridge and dot1dStpPortDesignatedPort for discovering edges between bridges instead of core bridge’s AFT entries for discovering neighbor bridges of any bridge. Then, we resolved the problem of following Property 3.2.

**PROPERTY 3.2 (Calculation of core bridge’s AFT size).**

The size of an AFT in a core bridge increases in proportion to the number of active interfaces of it and the value is as follow:

\[
\text{n(AFT}_{\text{CoreBridge}}) = \sum_{\text{active port}} \text{n(AFT}_{\text{active port’s DesignatedBridge}})
\]

Remark: The number of active ports from i to j is equal to the number of total active interfaces in a core bridge.

**PROPERTY 3.3 (Connection between Bridges).**

Let i and j be bridges in Large Ethernet Networks and G=(V, E) be an undirected graph. If i has a designated bridge, j then i be connected to j.

Remark: In an undirected graph, (i, j) is same as (j, i).

We first briefly describe how to find the set of bridges in the Large Ethernet network which form vertices. And we discover edges between bridges with designated bridge information. Figure 4 gives the pseudo-code based on Property 3.3, for finding the set of G=(V, E) regardless of calculating a spanning tree.

```plaintext
procedure FindBridgeGraph(B, D)
	/* G = (V, E) */
	begin
	vertexSet V ← ϕ
	edgeSet E ← ϕ

	V ← V ∪ B

	for each designatedbridgeSet D_i C D do{
		for each designatedbridge D_j in D_i do{
			if(b = D_j) then{
				D_i ← D_i - {D_j}
			}
		} for each designatedbridgeSet D_i C D do{
			if(D_i = ϕ) continue
			for each designatedbridge D_j in D_i do{
				E ← E ∪ {<b, D_j>}
			}
		}

	enend

Fig. 4. The FindBridgeGraph procedure
```

A bridged network is an undirected graph for offering multiple redundant paths that is, it only needs one direction information by Property 3.3. The number of sending neighbors have been changed according as a spanning tree was calculated or not. In the case of calculating a spanning tree, a root bridge can start sending a BPDU and some designated bridges receiving it also send other designated bridges with less priority than them. However, before calculating a spanning tree, for example a topology change happens in network, all bridges can start sending a BPDU. If a bridge has a lot of interfaces of running STP it has designated bridges same as the number of STP enabled interfaces. We denoted the set of designated bridge of a bridge i by D_i in the TopologyDiscovery procedure. The FindBridgeGraph procedure’s input arguments are the set of bridges and the set of designated bridge sets and the main goal of this algorithm is to find the set of edges, E and the set of vertices, V in bridged network. This procedure can
obtain real edge set after filtering duplication and dummy information.

### 3.2 Finding hosts connected to edge bridges

The last step of physical topology discovery is to find hosts connected to edge bridges. We used AFT entries of a bridge for discovering some connectivity with hosts. And we found following Property 3.4 in shared segments.

**PROPERTY 3.4 (Connection in the shared segments).**

Let $A$, $B$, and $C$ be hosts and $bj$ be $j$th interface of an edge bridge $b_i$. If $b_j$ has $A$, $B$, and $C$ for same interface $j$ as AFT entries, then $b_j$ is connected to interfaces of $A$, $B$, and $C$ through hub.

Remark: The $j$th interface of an edge bridge $b_i$ is active interface without running STP.

The FindHostGraph procedure, as shown in Fig. 5, has the set of edge bridges as an input parameter. The getAFTMIB is a metophysical terminology and is the subset of BridgeMIB. Each edge bridges must include its own active ports’ set, STP enabled ports’ set, and designated bridges’ set. To minimize influence of the size of AFT, a SNMP manager gets AFT from only edge bridges and the information is available for finding host connectivity. Hosts are connected to edge bridges through hub or direct link. We can obtain the edge set $E$ after running following procedure. We denote $j$th interface of an edge bridge $b_i$ by $b_{ij}$.

```plaintext

procedure FindHostGraph( $Q$ )
/* */

begin
  vertexSet = $\phi$
  edgeSet = $\phi$
  hostSet = $\phi$
  $V$ = $V \cup Q$

  for each edgebridge $b_i \in Q$ do(
    getAFTMIB($b_i$)
    $H_{i,j} = \text{findAFTMIB}(b_j)$
    $V = V \cup H_{i,j}$
  }

  for each edgebridge $b_i \in Q$ do(
    for each host $H_{i,j} \in H_{i,j}$ do(
      $E = E \cup \{(b_i, H_{i,j})\}$
    )
  )

end

Fig. 5. The FindHostGraph procedure
```

We can decide an edge bridge to be connected with hosts through a hub by Property 3.4. Finally we can find a point-to-point connection between an edge bridge and a host, if the number of the set $H_{i,j}$ is one.

### 4 Experiments

We have implemented the physical topology discovery algorithm presented in this paper and we have conducted several experiments using parts of ETRI’s own test network. The main purpose of these experiments is to find real physical topology including multiple redundant paths eliminated by STP.

#### 4.1 Implementation

We begin with a brief description of our implementation. We have developed SNMP manager using C plus and GUI with JAVA. We use IF-MIB and Bridge MIB for discovering physical topology in Large Ethernet networks. Figure 6 depicts a high-level view of our implementation architecture. When we click a Physical Topology Discovery button in GUI-Terminal, it communicates RMI Server by JAVA and RMI Server communicates a SNMP Manager. A SNMP Manager also communicates all SNMP Agents to get IF-MIB and Bridge MIB. After obtaining such information, SNMP Manager executes our topology discovery algorithm and then GUI-Terminal We used various vendors’ products such as Cisco3550, Riverstone3000, and Paxcomm NDX2124 on purpose for interoperability guarantee.

![Fig. 6. Implementation Architecture](image)

#### 4.2 Results

A primary goal of the experimental study with our topology discovery tool was to discover physical topology including a lot of edges eliminated by STP. And we found a graph with multiple inactive paths for a Large Ethernet network as well as spanning tree which related work had found. A second goal of our experimental study was to verify the practicality of our topology discovery algorithm, by measuring its running time requirements for various network sizes. Earlier future we will test the relationship between the number
of bridges and physical topology discovery time. Our topology discovery algorithm was not seriously affected by the number of hosts because we divided Large Ethernet networks into bridged networks and host networks, and in the bridged networks our algorithm does not need AFT entries. However, previous researches require AFT entries in core bridges and they have been seriously affected by the number of hosts as we addressed in the section 2.

Therefore our algorithm is very good solution to discover physical topology for Large Ethernet Networks. To discover physical topology in Large Ethernet networks, our solution uses STP MIB in the Bridge MIB for bridged network, and AFT MIB in the Bridge MIB for host network. For rapid convergence RSTP became the IEEE 802.1w [8] standard in 2001 and for supporting multiple trees in VLAN environments MSTP became IEEE P802.1s/D15 [9] standard in the end of 2002. But the standard MIB for RSTP is processing in IEEE draft [10] and the MSTP MIB is ready to propose to IEEE draft. If the RSTP MIB would be implemented we don’t need to get IF-MIB to decide edge bridges with the edge port in RSTP MIB instead of the number of active ports. Moreover, if we could get MSTP MIB in bridged network we will be able to offer multiple logical VLAN paths.

5 Conclusions

Automatic discovery of physical topology plays a crucial role in enhancing the manageability of modern Large Ethernet networks. Despite the importance of the problem, earlier research and commercial network management tools have typically concentrated on either (i) discovering logical (i.e., Layer-3) topology, which implies that the connectivity of all Layer-2 elements is ignored, or (ii) proprietary solutions targeting specific product families. In this paper, we have developed novel, practical algorithms for discovering true physical topology in Large Ethernet Network. The common problem of recent works is that they can only find spanning tree paths, not the physical topology in the Large Ethernet network. However our algorithm is only solution to be able to find physical topology in the Large Ethernet network. We are currently in the process of optimizing our implementation and conducting more extensive experimental tests, and hope to be able to report more detailed performance results in the near future. Also we will continue to study about physical topology discovery using RSTP and MSTP to pass the limit of STP.

References: