Available online at www.sciencedirect.com

SciVerse ScienceDirect

Physics Procedia 33 (2012) 252 – 257

Physics Procedia

2012 International Conference on Medical Physics and Biomedical Engineering

Research and Implement of an Algorithm for Physical Topology Automatic Discovery in Switched Ethernet

Yuting Xiong[1], Zhaojun Gu[1], Wei Jin[2]

[1] Institute of Computer Science and Technology, Civil Aviation University of China, Tian Jin, China
[2]IP Development Dept, Huawei Technologies Co., Ltd, Beijing, China
ytxiong@cauc.edu.cn, zjgu@cauc.edu.cn,jinwei0809@huawei.com

Abstract

In this paper, a novel practical algorithmic solution for automatic discovering the physical topology of switched Ethernet was proposed. Our algorithm collects standard SNMP MIB information that is widely supported in modern IP networks and then builds the physical topology of the active network. We described the relative definitions, system model and proved the correctness of the algorithm. Practically, the algorithm was implemented in our visualization network monitoring system. We also presented the main steps of the algorithm, core codes and running results on the lab network. The experimental results clearly validate our approach, demonstrating that our algorithm is simple and effective which can discover the accurate up-to-date physical network topology.

Keywords—Physical Network Topology, Address Forwarding Table, SNMP MIB, Switched Ethernet

Introduction

The discovery of network topology plays an important role in modern network management. It mainly focuses on auto topology discovery for network layer (layer-3) and data-link layer (layer-2). Layer-2 topology, also called physical topology, refers to show the physical connectivity relationships that exist among entities in a communication network. The discovery algorithms on network layer are very mature. Unfortunately, network layer topology covers only a small fraction of the interrelationships in an IP network, since it fails to capture the complex interconnections of layer-2 network elements (switches, bridges, and hubs) that comprise each Ethernet LAN.

Developing effective algorithmic solutions for automatically discovering the up-to-date physical topology mainly facing the following challenges:

1) Inherent Transparency of Layer-2 Devices. Unlike layer-3 network elements (routers), layer-2 network elements (switches, bridges, and hubs) are completely transparent to endpoints. Without knowing the existence of layer-2 network devices, endpoints transport data directly.

 doi:10.1016/j.phpro.2012.05.059

2) Heterogeneity of network elements. The discovery algorithm should be able to gather topology information from heterogeneous network elements, making sure that the relevant data collected in the elements of different vendors are accessed and interpreted correctly[1].

3) Complexity of network topology. In complex network there may be kinds of topology(star, bus and ring), running different protocals(IBM SNA, Xero XNS, AppleTalk, TCP/IP and DECnet). In the other hand, deploying "dumb" elements like hubs and "uncooperative"switches which can't supply Address Forwarding Tables (AFTs) aslo add the difficulties to automatic phiyscal topology discovery.

Much research work has been processed to provide solutions for physical topology discovery in IP or Ethernet networks since 2000. Here lists the representative work. Yuri Breitbart et al. [2] [3] proposed an algorithm that relies solely on standard AFT information collected in SNMP MIBs to discover the direct port connection relationship between switch. Kaihua Xu et al. [4] proposed an algorithm used information reachable address and port's traffic statistics information. Li Yupeng et al. [5] proposed algorithm for Ethernet topology discovery based on STP. However, these methods proved to be complex and ineffective or need additional support in physical devices such as STP.

Actually, we need our network management tool can provide up-to-date physical topology of monitored Ethernet in short time for the topology may change frequently with the some devices added and other devices removed. And the manageable physical devices are more and more popular in modern switched network. In this paper, we propose a novel, practical algorithmic solution for automatic discovering the physical topology of switched Ethernet which collects standard SNMP MIB information that is widely supported in modern IP networks. Practically, it was proved to be simple and effective.

Definitions and system model

In this section, we present necessary background information and the system model that we adopt for the physical topology discovery problem.

B. Background and Definitions

In our work, we designed an automatic physical topology discovery algorithm to help locating the illegal access in inner switched Ethernet of cooperation and campus. We refer to the domain over which topology discovery is to be performed as a switched domain, which essentially comprises a maximal set S of switches such that there is a path between every pair of switches involving only switches in S. The following assumptions and basic requirements drove the design of the algorithm:

- The network consists of Layer-2 Ethernet switches in the core and Layer-3 Ethernet switches (also known as switch/routers) at the edge;
- Each switched domain (i.e., collection of switched IP subnets) consists of a single switched subnet and connects to the "outside world" through one or more layer-3 routers(also known as gateway)(see Fig. 1);
- The nodes must have the following protocols and MIB implemented: SNMP, MIB-II [6] and Bridge MIB [7].

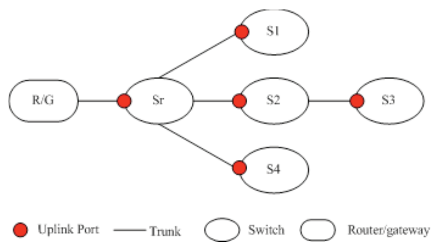

Figure 1. Switch domain model

Fig. 1 shows a switch domain model, the circle with a letter inside is switch. The round angle rectangle is router/ gateway. Since loop path(s) in active switched Ethernet would cause broadcast storm [8] and bring the network down, we model the target switched domain as an undirected tree, which captures the (tree) topology of unique active forwarding paths (e.g., Sr→S2→S3) for elements within a switched domain. On each forwarding path, we define the pair of switches (e.g., {s2, s3}) connecting directly has father-son relation, the one (s2) closer to root switch (Sr) is called father, the other one (s3) is called son. In this algorithm, several concepts are mentioned which were defined as following:

Definition1: We say that AFT information on one switch port is complete if it contains the MAC addresses of all network nodes from which frames can be received when the AFT is large enough and wouldn't age.

Definition2: We say that AFT information on one switch port is uncompleted if it may not contain the MAC addresses of all network nodes from which frames can be received in all time, for the capacity of AFT is limited and the entry corresponding to that source address in AFT would be deleted if no frame(s) received from a source address at intervals smaller than the aging interval.

Definition3: We define that all switches are offspring (i.e., {s2, s3} are offspring of Sr.) of the switch with $\tilde{r}$ (uplink complementary set) in which there are these switches' MAC address.

Definition4: We define that leaf switch (also called access switch, i.e., {s1, s3, s4}) is the switch with $\tilde{r}$ in which there is no any other switch's MAC address.

Table I summarizes the key notation used throughout the paper with a brief description of its semantics. Additional notation will be introduced when necessary.

Table 1 Notation

Notation	Meaning
S	Set of all switches in a switch domain
Sr	Switch directly connects to router/gateway, called root switch.
S_i	Any switch in S
rp	Switch port on son (switch) directly connects to father (switch) or router/gateway, called uplink port. It is represented by red point in Fig.1.
fp	Switch port on father (switch) directly connects to rp.
r	Set of AFT entries on rp, called uplink port set
$\tilde{r}$	Set of AFT entries on all switch ports except rp, called uplink complementary set.
f	Set of AFT entries on fp, called father port set
M_g	MAC address of router/gateway
U	Set of all switched MAC address in a switch domain

C. System Model

Dissimilar other algorithms, we try to reason the physical topology relationship between switches by new point of view. Therefore, we introduced the following theorems.

Theorem 1: The uplink port (rp) in a switch must be the port with the router/gateway MAC address (M_g) in its AFT(r).

Proof: The uplink port of Sr connects directly with R/G (gateway/router) and the frame from R/G must be received on it. In reverse path learning process, a switch associates the source MAC address of an incoming frame to the port on which that frame arrived and the AFTs of the port are automatically filled up with MAC addresses. So r of Sr should record M_g. By the same token, r of Sr's offspring would record M_g when the frame was forwarded along the trunk path. So rp must be the only port in Si with M_g. This completes the proof.

Theorem 2: In the network with complete AFT information, U = Sr. $\tilde{r}$ and f $\supset$ son. $\tilde{r}$

Proof: According to definition 1, fp of Si (if Si isn't leaf switch) can record MAC addresses of nodes sending frames which have been recorded by son. $\tilde{r}$ after sufficient long time for all the nodes to communicate with each other, so f $\supset$ son. $\tilde{r}$. As the root switch, all the other switches are its offspring, Sr. $\tilde{r}$ contains the switched MAC address of all switches, which is denoted by U, then U= Sr. $\tilde{r}$. This completes the proof.

Theorem 3: In network with uncompleted AFT information, the AFT of a switch can contain all sons' MAC addresses of it by communication with them constantly

Proof: For looked as a common network node, the MAC address of a switch should be record by its father and not be aging through constant communication. And in our switched Ethernet the number of switches is far less than the capacity of AFT in any switch. This completes the proof.

Inference 1: The port must directly connect with the leaf switch when its AFT only contains one switch MAC address of the leaf switch and no other switch MAC address.

We can prove it easily by reduction to absurdity from theorem 3.

Theorem 4: The offspring of a switch must connect with its corresponding non-uplink port directly or indirectly.

Proof: According to definition3, the MAC address of each offspring switch is recorded in $\tilde{r}$. And it must point to a non-uplink port in $\tilde{r}$ which are the set of MAC address and port mapping entries. Just as all switches would record M_g in its r given by theorem1, there must be a trunk path from each switch to forward frame to the non-uplink port; in other words, they must connect directly or indirectly on the port. Thus, the theorem 4 is proven.

From theorem 4, we can easily induce the following inference 2.

Inference 2: The offspring of S_i must connect indirectly with father of S_i.

Proof: Assume the offspring of the S_i connected directly with father of S_i, the offspring and S_i must connect directly to different ports of S_i's father. In other words they both are the son of same father which doesn't match with the proposition. Thus, the inference 1 is proven.

Overview of our topology discovery algorithm

We can obtain a set of constraints and conclude the physical connection relationship among the switches of S according to above definitions and theorems. As following are the main steps of our physical topology discovery algorithm:

Step1 S= Φ ;

Step2 Get all the nodes' MIB information of the switched domain using SNMP, judge the device type as switch according to the value of *sysServices* in SNMP System group[7], and then add all found switches (S_i) into S;

Step3 Get the AFT of Si, use hash table to record AFT entries for each port of S_i in S;

Step4 Get M_g, search M_g in the AFT of Si and fix on the uplink port of Si according to theorem 1;

Step5 Get $\tilde{r}$ of Si, fix on Sr by judging whether the MAC address of other switches in S are contained in it according to theorem 2 and 3;

Step6 Build the offspring list of Sr and other switches in S by searching their $\tilde{r}$ according to theorem 4 ;

Step7 According to inference 1and 2, travel over the offspring list of S_i (Sr first) built in step 6 on the rule of from up to down and depth-first, let the father pointer of S_i to point fp and fp's son pointer point to son if the direct connection relation has been gained, build the complete physical topology (tree) of S until recursion finished.

Implementation

The algorithm has been implemented in Java as an independent module of our visualization network monitoring system and can run on Windows/Linux Platform. Publicly available, open source SNMP4J API is used to retrieve management information of network nodes.

D. Main Function Function of the module

The main function of the module is to build the physical topology of designed monitored network, trace and locate the given nodes at access port on the topology.

E. Implementation of the module

As following are the design and implementation process of the module:

1) Scanning the monitered network: Send UDP datagram to all nodes of the target network on random port (port number>4000). If Port Unreachable messages received, we can know the node is active. Send ICMP ping packets to the node again if it has no response. By scanning, we can get the latest AFT information prepared for next discovery task.

2) Build physical topology tree: Build physical topology automatically according to the algorithm proposed in section Ⅲ. Fig. 2 show*s* the core codes of the procedure. Here *Switch* is the object class defined to represent real switch device discovered in the program. The function of core codes given by Fig.2 is to travel over the switch list beginning with root switch, find the son, and record the physical connection relationship between father and son by changing the value of corresponding member variables (i.e. child, parent, parentPort) of the Switch Object.

```
    /* Module of switch topology tree discovery*/
public void getTop()
{// travel from Sr
    Switch root=switchList.get(rootSwitch);
    int hasFond=0;
    //travel the port list of Sr
       for(inti=0;i<root.getPortList().size()&&
           hasFond<(switchList.size()-1);i++)
    {   //don't process uplink port
          if(i==root.getRootPort()-1)continue;
       SwitchPort port=root.getPortList().get(i);
       for(Switch sw:switchList)
       { //judge whether sw is the offspring of port
           if(port.macAddress.containsKey(sw.getMac()))
          {//add sw to offspring set
          port.child.add(sw);
          hasFond++;           }}
       for(Switch sw:port.child)
       {//build the topology tree of son by recursion
             sw.findChild(port);
          for(Switch sw1:port.child)
          {// connect leaf switch to the switch port
             if(sw1.isLeaf&&!sw1.isFinish) {
                   port.nextSwitch.add(sw);
                     //pointer point to father
                sw.parent=root;
                     //record the number of fp
                sw.parentPort=i+1;                }}}}
```

Figure 2. Core codes of building physical topology tree

3) Trace and locate the appointed node on physical topology tree above: The detail content about node locating wouldn't be mentioned here, for it isn't the emphases of our paper.

4) Show the discovery result of physical topology tree in graphics: Lay out the automatic physical topology discovery result dynamically by drawing the discoverd nodes (i.e. router, switch, host) and the link among them in graphics(e.g. line represent the direct connection between two nodes). Fig. 3 shows an running result of the module under lab network which contains one Catalyst3550, one Catalyst2950 switch and a gateway. The number depict the active port number on network node connecting to other device and the green line represent the active link which is working normally.

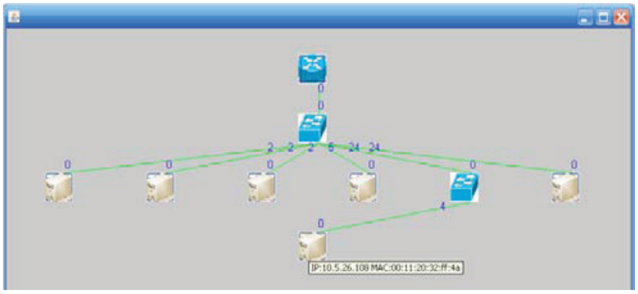

Figure 3. Result of pyisical topology discovery module

Conclusions

Automatic discovery of physical topology information plays a crucial role in the modern network management. In this paper, we have proposed a novel algorithm solution for discovering the physical topology of switched Ethernet which relies on standard SNMP MIB information that is widely supported in modern IP networks. And the algorithm has been applied to the design of visualization network monitoring system for the airport IP network management. It brings the up-to-date physical topology of active switched Ethernet to us in short time and has been proved to be simple and effective.

Acknowledgment

The authors gratefully acknowledge Science Technology Fund of CAAC for their support.

References

[1] YANG Xia, BI Yan bing and SUN Yan tao, "Progress and research on physical topology discovery of switched Ethernet," in Application Research of Computers, 12rd ed.,vol.24. Bei Jing: pp. 24–27, July 2007.

[2] Y. Breitbart, M. Garofalakis, C. Martin, R. Rastogi, S. Seshadri, and A. Silberschatz, "Topology Discovery in Heterogeneous IP Networks," INFOCOM, March 2000, also in IEEE/ACM Transactions on Networking, Vol. 12, No 3, June 2004, pp. 401-414.

[3] Y. Bejerano, Y. Breitbart, M. Garofalakis and R. Rastogi, "Physical Topology Discovery for Large Multi-Subnet Networks," IEEE INFOCOM 2003, San Francisco, USA, Apr. 2003, pp. 342-352.

[4] Kaihua Xu, Jiwei Cao , Yuhua Liu2 , and Shaohua Tao, "An Algorithm of Topology Discovery in Large Multi-Subnet Physical Network," Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06).

[5] Li Yupeng, Wang Huanzhao and Zhao Qingping, "An Algorithem Based on STP for Ethernet Physical Topology Discovery," in Journal of Beijing Electronic Science and Technology Institute, 2rd ed.,vol.12. Bei Jing: pp. 8–13, Jun 2004.

[6] K. McCloghrie and M. Rose, "Management Information Base for Network Management of TCP/IP based internets: MIB-II," RFC 1213, March 1991.

[7] E. Decker, P. Langille, A. Rijsinghani and K. McCloghrie "Definitions of Managed Objects fro Bridges," RFC 1493, July 1993.

[8] Cisco Systems Inc., CCNA 1 and 2 Companion Guide, Revised (Cisco Networking Academy Program) (3rd Edition). Indianna, USA: Cisco Press, July 2005.