Network Management and Control System for a Homogeneous Network using Light Weight SNMP

(Light weight Simple network management protocol)

Hamsaveni B S
Department of Computer Science and Engineering,
H K B K C E Bangalore.
Bangalore, India

Prof. Smitha Kurian
Department of Computer Science and Engineering,
H K B K C E Bangalore.
Bangalore, India

***Abstract*— Discovering the network topology helps in analyzing the faults in the network and their locations. Rectifying such faults is a key role of an organization network management system. Thus, the automatic discovery of network topology has been the subject of rigorous study for many years. This paper proposes a simple method of discovering network topology using Simple Network Management Protocol (SNMP) that handles various types of network devices, including all layer switches, routers, printers, and hosts. Here we are using the database to store and generate the results. And also discovers connectivity among these devices. Main aim of this paper is to discover end host connectivity with the switches and routers in the subnet of an organization which helps to improve network performance and reduce infrastructure costs. It is simple because it is configured in such a way that we can execute it wherever we need.**

Keywords*— *Simple network management protocol (SNMP), Management information base (MIB), Object identifier (OID).

I. Introduction

Network topology is the study of the arrangement of links and nodes in a network and the interconnections among the nodes. We need to discover the issues like discovering various types of devices, Network topology visualization, discovering complete topology. Our main strength is using a simple SNMP algorithm to connect the end host with the network.

Simple Network Management Protocol (SNMP) is an application–layer protocol used for exchanging management information between network devices. SNMP is one of the widely accepted protocols to manage and monitor network elements. Most of the professional–grade network elements come with bundled SNMP agent. These agents have to be enabled and configured to communicate with the network management system (NMS). Enabling the agent allows it to collect the management information database from the device locally and makes it available to the SNMP manager, when it is queried for. We first analyse and then utilize management information base (MIB) objects required to build a discovery algorithm.

In this paper we distributed work into three types a) Device discovery, b) Device type discovery, c) Connectivity Discovery. We will be retrieving the details of the devices in the network using the management information base (MIBs) through specific object identifier (OID) value. Specific object identifier value is used to retrieve specific details required by user. The three types will be discussed in later sections.

II. Related Work

Many studies [1–4] conducted in the area of discovering network topology using ping, trace route, Simple Network Management Protocol (SNMP) [15], and other methods have shown remarkable results. However, they fail to address the following issues: *Discovering various devices*, Yuri et al. [2] presents novel algorithms for discovering physical topology in heterogeneous IP networks. Our algorithms rely on standard SNMP MIB information that is widely supported by modern IP network elements and require no modifications to the operating system software running on elements or hosts. R.Siamwalla. [1] described several heuristics and algorithms to discover both intra-domain and Internet backbone topology while making as few assumptions about the network as possible. We quantitatively evaluate their performance.

B. Lowekamp. [3] described an approach to Ethernet topology discovery that can determine the connection between a pair of the bridges that share forwarding entries for hosts. This minimal knowledge requirement significantly expands the size of the network that can be discovered. Cloghrie[12] found MIB-II, RFC 1213 which helps in Recursive Device Discovery. This information is sufficient for discovering almost all the devices in the network. Fedor [12] gave brilliant idea SNMP which uses Structure of Management Information (SMI) which defines the Management Information Base (MIB). Internet community became network manageable in a timely fashion. The requirements of the SNMP and the OSI network management frameworks were more different than anticipated. We found that Simple network management protocol (SNMP) is the best among all the techniques like ping, trace route, ARP, etc.,

The paper describes as follows. The topology discovery algorithms are explained in Section 2, implementation and experiments are discussed in Section 3. Section 4 concludes our work and discusses future research directions.

III. Topology Discovery Algorithm

In this section, we present our technique to discover network nodes and connectivity among them. We utilize MIBs to build a discovery algorithm, which is basically divided into three different modules, namely *device discovery*, *device type discovery*, and *connectivity discovery.* Since our technique is mainly based on SNMP, we first analyse the management information base (MIB) objects required to build our algorithm.

TABLE I . MIB INFORMATION FOR TOPOLOGY DISCOVERY

MIB-II (RFC 1213)
sysDescr, sysUptime, sysServices, ifIndex, ifDescr, ifPhyaddress, ipRouteNextHop, ipRouteType, ipAdEntAddr, ipAdEntNetMask, ipNetToMediaNetAddress, ipNetToMediaPhysAddress.
PRINTER MIB (RFC 1759)
HrDeviceStatus, HrPrinterStatus, PrtInputVersion, PrtInputMediaType, PrtOutputType, PrtOutputStatus, PrtOutputDescription, PrtMediaPathIndex, PrtMediaPathDescription.
BRIDGE-MIB (RFC 1493)
dot1dBase, dot1dBasePort, dot1dBasePortIfIndex, dot1dTpFdbAddress, dot1dTpFdbPort, dot1dTpFdbStatus.

A. *Device discovery*

Device discovery uses the MIB-II (RFC-1213) to retrieve the details from the agent from the manager system. In this will take switch or router as the input in the manager. We can discover the devices connected to the switch in two ways; one is from the IP Addresses and other is from Mac Addresses. In IP Address discovery the first step is to get all the IP Address by getting ipNetToMediaNetAddress response from agent. If the ipNetToMediaNetAddress is not available then exit from loop else call IP Address discovery recursively to get IP Address of all the devices connected to the switch. Similarly follow the same instructions for Mac Address discovery but instead of ipNetToMediaNetAddress get the response as ipNetToMediaPhyAddress from the agent.

ALGORITHM 1: DEVICE DISCOVERY ALGORITHM

```
Input: Switch/Router.
Output: IPAddress/MacAddress of all the devices connected.
    1.  Take input as Switch/Router.
    2.  IPAddress Discovery
            i.   Get all ipNetToMediaNetAddress
            ii.  If there is no ipNetToMediaNetAddress,
                 then return.
            iii. Call IPAddress Discovery Recursively.
    3.  MacAddress Discovery
            i.   Get all ipNetToMediaPhyAddress
            ii.  If there is no ipNetToMediaPhyAddress,
                 then return.
            iii. Call MacAddress Discovery.
```

B. *Device type discovery*

In the device type discovery we use the Bridge MIB information and Printer MIB information to distinguish the device type in the network. Take the response from the agent. If that response is supported by the Bridge MIB information then we can find the connectivity so that we can state that device as a switch. And if the response is supported by the Printer MIB then we can get the printer details with its specifications thus we state that device as printer; otherwise we can say that device is a host. Here we can discover the type of the switch that is the layer (L2/L3/L4/L7) to which the switch belongs to. This can be decided by taking the sysService value from the agent response and converting it to seven-bit string. Each bit corresponds to the 7 layer of the OSI network model. After converting it to seven-bit string the position of the enabled bit will say that the switch will provide services to which layer. If a device has *sysServices* 6(0000110)—its second and third bits are set—then the device is an L3 switch that provides services for these two layers.

ALGORITHM 2: DEVICE TYPE DISCOVERY ALGORITHM

```
Input: Response of the given request.
Output: Device type and the switch type of network.
    1.  Get Response or resource obtained.
    2.  Device type Discovery
            i.   If response is supported by Bridge MIB,
                 then return Switch.
            ii.  Else if response is supported by Printer
                 MIB, then return Printer.
            iii. Else return device as Host.
            iv.  Call Device type Discovery Recursively.
    3.  Switch type Discovery
            i.   Get SysServices value.
            ii.  Convert value into seven-bit string.
            iii. Identify switch type by the position of bit.
```

C. *Connectivity discovery*

In the connectivity discovery we mainly use the Bridge MIB information for port connectivity discovery. Here we can find which device is connected to which port of the switch in the network.

ALGORITHM 3: CONNECTIVITY DISCOVERY ALGORITHM

```
Input: MacAddress, IPAddress and Port numbers.
Output: Connectivity between ports.
    1.  IPAddress set IPn= {I1, I2 ,….., In} Get all
        ipNetToMediaNetAddress.
    2.  MacAddress set Mα= {M1, M2 ,…., Mn} from
        ipNetToMediaPhyAddress.
    3.  Port Connectivity Discovery
            i.   Get Port numbers from 1 to n.
            ii.  Get MacAddress set Mβ= {M1, M2….., 
                 Mr} from Bridge MIB.
            iii. If Mα={M1,M2,….,Mn} matches
                 Mβ={M1, M2….., Mr}, then return IPn
                 and Port numbers.
            iv.  If Port number already exists, then return
                 Uplink port.
```

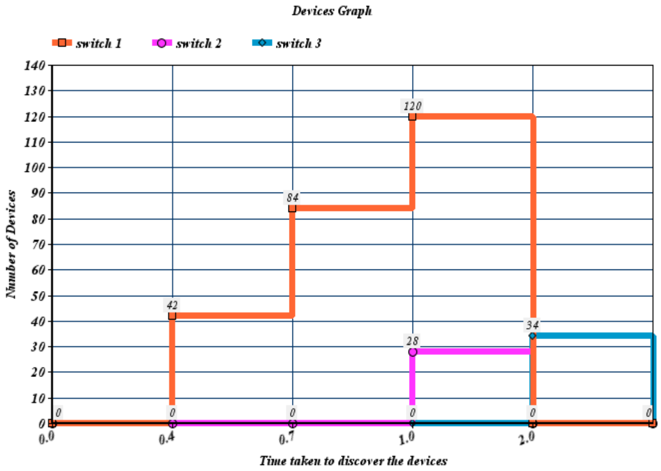
Take the input as set of IP Addresses from ipNetToMediaNetAddress and Mac Addresses from ipNetToMediaPhyAddress in one side. And other side, take ‘n’ port numbers and set of Mac Addresses from Bridge MIB. If Mac Address of one set matches the Mac Address of the other set then the IP Address and the port numbers of that device are printed. Thus we can get the port number of the device connected to switch. And if the port number obtained is already exists in the list then we treat that port as Uplink port.

In this algorithm we are discovering the interface-to-interface connectivity between the L2 switch. The main aim of this algorithm is to find the port connectivity of the devices in the network switch.

IV. Implementation And Experiment

We used JDK 1.6 with Netbeans 6.9.1 tool, SNMP API(SNMP4J), and Jfreechart for graphical representation. We developed and tested our system on Red Hat Linux 5.4 operating system with 2.80-GHz, Intel Pentium 4 CPU with 512 MB RAM.

We tested our approach in the subnet of a network or we can say that department of an organisation. In this subnet we found number of devices connected to the switches, their details and also the connection between them. These are tested multiple times and physically verified. The problem we faced is that some devices are having multiple Mac Addresses so it is difficult for the system to find the connectivity of the device to port. Other than this everything is working properly. The major thing we did is when the device is not supporting for SNMP then we tried to get the details of that system by ICMP echo requests.



The time taken to discover the devices connected to switch are compared with the previous research [4]. And it is very less compared to other research. In very short time nearly 8-9 seconds we can discover all the devices connected to the switch in the network.

V. Conclusion And Future Work

In this paper, we not only focused on discovery of devices and their connectivity. We extended the work of others by introducing an algorithm to collect specific information that helps in device type discovery in the network. We discovered different types of devices, including routers, switches, printers, and end hosts and enhanced the already existing technique of device type discovery and Connectivity discovery. We utilized the SNMP mechanism, which is the most efficient and generates the least amount of traffic, in comparison to mechanisms like ping, trace route, ARP, etc., Our work can be a guideline in implementing an SNMP-based topology discovery system. Irrelevant with SW/HW platform, easy for correcting, maintenance, update and upgrade. This system does well in stability, reliability and safety. Our extensive tests are significant in terms of efficiency and the number of devices discovered. We discovered several devices in 8-9 seconds of time. This work can be extended by integrating it with Traps and notifications.

Our future goal is to add the traps, notification and alarms in case of any network error or damage. And also graphical visualization of network to identify errors easily. For greater accuracy, our end host connectivity algorithm needs more refinement. Various analyses of changes to topology will also be done in future which can help us discover the growth patterns of networks.

Acknowledgment

The Acknowledgement word is very suitable in this work for the organization (ISRO Satellite Centre, Bangalore) where they gave me an opportunity to implement my technique and to utilize my knowledge. I whole heartedly thank the organization and my guide who helped to complete this work.

References

[1]. R.Siamwalla, R. Sharma, and S. Keshav, “Discovering internet topology, ” Cornell Univ., Ithaca, NY, Techical Report, May 1999.

[2]. Y. Breitbart, M. Garofalakis, B. Jai, C. Martin, R. Rastogi, A. Silberschatz, “Topology Discovery in Heterogeneous IP Networks: The NetInventory System,” IEEE/ACM Transactions on Networking, vol. 12, no. 3, June 2004, pp. 401~414.

[3]. B. Lowekamp, D. R. O’Hallaron, T. R. Gross, “Topology discovery for large Ethernet networks,” ACM SIGCOMM, August 2001, San Diego, CA, USA, pp. 237~248.

[4]. Suman Pandey, Mi-Jung Choi, Sung-Joo Lee, James W. Hong “IP Network Topology Discovery Using SNMP”, POSTECH, Korea 2013.

[5]. K. McCloghrie, M. Rose, “Management Information Base for Network Management of TCP/IP-based Internets, MIB-II,” RFC 1213, IETF, March 1991.

[6]. http://www.cse.wustl.edu/~jain/cis788-97/ftp/net_monitoring/ Network monitoring needs and methods.

[7]. http://www.monitortools.com/tech/snmp/mib/ information about SNMP MIBs.

[8]. http://blog.monitorscout.com/2013/04/02/the-benefits-of-snmp-monitoring/ benefits of the SNMP remote monitoring.

[9]. http://www.ietf.org/rfc/rfc1759.txt/ Printer MIB information.

[10]. http://www.faqs.org/rfcs/rfc1493.html Bridge MIB information.

[11]. http://www.ietf.org/rfc/rfc1514.txt Host-Resources-Mib, From Rfc 1514.