An Efficient Approach Towards IP Network Topology Discovery for Large Multi-subnet Networks

[1]Fawad Nazir, [1]Mohsan Jameel, [1]Tallat Hussain Tarar, [1]Hamid Abbas Burki, [2]Hafiz Farooq Ahmad, [1]Arshad Ali
[2]Hiroki Suguri,
[1]NUST Institute of Information Technology (NIIT)
Rawalpindi, Pakistan
E-mail: { fawad.nazir,mohsan.jameel,tallat.tarar,hamid.abbas,arshad.ali}@niit.edu.pk

[2]Communication Technologies (Comtec)
Sendai, Japan
E-mail: {farooq,suguri}@comtec.co.jp

***Abstract*— Knowledge of the up-to-date network topology (i.e. Layer 2 & Layer 3) is crucial for efficient network management. Issues like congestion avoidance, resource management, resource discovery, root-cause analysis and event correlation require accurate information of the topology map. Due to dynamic nature of today's IP networks, keeping track of topology information manually is a daunting task. Thus efficient algorithms for automatically discovering physical network topology are necessary. The earlier work has typically concentrated on discovering topology either using completely SNMP-MIB or ICMP echo request/reply, DNS, Trace route, etc. Our proposed algorithm does not rely totally on SNMP-MIB information as it is not usually supported by all devices in the network. Instead we propose an approach where SNMP agent should only be enabled on the routers and managed switches. Rest of the network computers/hosts need not have SNMP agent enabled. We propose an efficient algorithm not only to find available computers/hosts but also to find appropriate timeouts and delays in the network. The experimental results validate our approach, demonstrating that our algorithm discovers accurate physical topology.**

1. INTRODUCTION

Network topology is a representation of the interconnection between directly connected peers in a network. In a physical network topology, peers are ports on devices connected by a physical transmission link. At the IP level, peers are hosts or routers, one IP hop away from each other, and at the workgroup level the peers are workgroups connected by a logical link. Network topology constantly changes as nodes and links join a network, personnel move offices, and network capacity is increased to deal with added traffic. Keeping track of network topology manually, is a frustrating and often impossible job. It is sometimes impossible to identify the existence of multiple paths between hosts, switches, routers and printers. Network topology knowledge including the path between endpoints, can play an important role in analyzing, engineering, and visualizing networks. Network topology information is useful in deciding whether to add new hosts, switches, routers, printers and to figure out whether current hardware is configured correctly. It also allows network managers to find bottlenecks and failures in the network. Discovering the physical layout and interconnections of network elements is a prerequisite to many critical network management tasks, including reactive and proactive resource management, server sitting, event correlation, and root-cause analysis. This fuels the need for the development of effective, general-purpose algorithmic solutions for automatically discovering up-to-date physical topology of an IP network. One of the main challenges in the design of such algorithms is dealing with the lack of established, industry-wide standards on the topology information maintained locally by each network element, and the diversity of elements and protocols present in today's multi-vendor IP networks. Traditional topology discovery algorithms are entirely based on SNMP, which is not universally deployed. Both network management and performance analysis benefit from network topology knowledge. Network managers equipped with software's to automatically detect the topology of their network will be in better position to prevent and solve continuous network problems. They will not only be able to detect the exact location of a problem and trace it back to the source, but also analyze the use of the network under normal operations. This knowledge allows them to anticipate problems and plan for them before the services are impacted. In this paper we have proposed an algorithm for IP network topology discovery for large and multi-subnet networks. Our algorithm does not require configuration of SNMP agent on every device in the network. We argue to accurately discover the physical topology by only enabling the SNMP agent on router, switches and network printers.

The paper is organized in the following manner: Section 2 briefly describes the related work which has been undertaken in this field. Section 3 describes our contribution. This is followed by Section 4 which explains in detail our implementation architecture. Finally, we discuss the algorithms used in our approach in Section 5, followed by conclusions and future work in Section 6.

2. RELATED WORK

SNMP-based algorithms for automatically discovering network layer topology are featured in many common network management tools, such as HP's OpenView (www.openview.hp.com) and IBM's Tivoli (www.tivoli.com). Peregrine's Infratools software (www.peregrine.com), Riversoft's NMOS product (www.riversoft.com), and Micromuse's Netcool/Precision application (www.micromuse.com) claim to support layer-2 topology discovery, but these tools are based on proprietary technology to which is not widely used. Several approaches to finding layer-3 topologies have been proposed (e.g., [1, 2, 3, 4, 5]). One approach [6] uses pattern matching on interface counters available through SNMP. Another approach [7] finds the topology based on tables for the spanning tree algorithm available through SNMP. In [8] the concept of the operational topology of an Internet Protocol (IP) network and a technique for discovering it are discussed. Several heuristics and algorithms discover both intra-domain and Internet backbone topology while making as few assumptions about the network as possible these algorithms quantitatively evaluate their performance and also present a new technique for visualizing Internet backbone topology [9]. Algorithms rely on standard SNMP MIB information that is widely supported in modern IP networks and require no medications to the operating system software running on elements or hosts [10]. This minimal knowledge requirement significantly expands the scope of the networks that can be discovered. The network topology information especially at the LAN level, is important for both the management and use of networks [11]. Algorithms to discover some details of the link layer topology of the Intranet. The network layer topology discovery finds the devices in the subnet. The link layer discovery is to detail the connection among the link layer devices and the active spanning tree. An important characteristic of Untwine is that it does not require the forwarding database to be complete for link layer topology discovery [12]. There are tools available that are used for monitoring the network and for discovering the network topology like InterMapper LAN Surveyor 4.1, Solarwinds, NetworkView. Our algorithm is used to discover both Layer 3 and Layer 2 topology of the network. In our approach the display and discovery process works in parallel all the changed in the toplogy are traced at run time on the visualization module. Another unique feature of our algorithm is this that multiple users can view our topology map at one time and all users will me updated by the changes.

3. OUR CONTRIBUTION

In this paper, we propose a novel, practical algorithm for discovering the physical topology of large multiple subnets networks. Our algorithm discovers routers, managed and unmanaged switches, network hosts, network printers and interconnection between them. Some earlier approaches have focused on the internet topology using ICMP and trace route information as internet topology require discovery of Layer 3 only. Other approaches which are focusing on network topology discovery either require completely SNMP-MIB information on all the nodes of the network or using network ICMP echo request/reply, DNS, Trace route. Using totally SNMP-MIB information can lead toward accurate topology discovery but configuring and enabling SNMP agent on every device in the network is not practical. On the other hand using ICMP-Echo request/reply technique can not discover an accurate topology discovery. In this paper we have proposed an algorithm which does not rely totally on SNMP-MIB information as usually it is not widely supported by all the computers in the network and enabling SNMP agent on all the computers in large network is not practical. Instead we propose an approach where SNMP agent is enabled only on the routers, managed switches and network printers. Rest of the network devices need not have SNMP enabled on them. We have developed an algorithm based on the ARP cache of the router to determine the IP ranges that are expected to be available thus reducing the query to IP ranges which do not have any device running (As querying the IP which does not reply makes the algorithm slow and inefficient due to the timeouts). Our proposed algorithm gets rid of the timeouts in our request. We send multiple probes to the device with out allocating any resource that will wait for the reply. The machines which replies are considered to be up and other are considered being down. Further more our approach can be used to find out the unmanaged devices and their connectivity with the managed devices. This is achieved by getting information from the Address Forwarding Table of the managed switches. Using this information, we find out the switch to switch connections, switch to router connection and switch to host connections. The complete Address Forwarding Table information from all network nodes is often insufficient to uniquely identify the underlying physical network topology as the Address Forwarding Table contains a mapping of the MAC address which is connected to the specific port of the switch. We have successfully solved this problem in two simple steps by creating a data structure of IP to MAC on the monitoring host. In the first step we send ICMP echo request to the IP whose MAC we require and then in the second step we get the ARP cache from the monitoring host to get the MAC Mapping of that particular IP and insert it into the data structure. The above mentioned technique will work well as far as the IP device is in the same subnet (broadcast domain) as the monitoring host. As the routers do not allow the MAC level broadcast to pass through, so the MAC address of the computers which are one or more hops away will not be available to the monitoring host. This problem is solved by sending an SNMP query to get the ARP mapping from the router with which the host is connected. In this way we will be able to discover the topology of other subnets in our network. The information about the IP and subnet mask is obtained from the routing table of the router in our network. In simple words, our algorithm first discovers the topology of its own subnet and then from the gateway router gets the information about other subnets and discovers the topology of other subnets. So in this way our algorithm enters an iterative mode to discover complete arrangement of the network elements. Our algorithm finds the interconnections to infer the underlying network topology including the connections of "invisible" hubs and uncooperative switches. We are able to demonstrate a strong *completeness* property of our algorithm. More specifically, we show that without SNMP agents on all the machines, network topology can be discovered accurately and efficiently. To the best of our knowledge, ours is the first non SNMP-based topology discovery algorithm to provide such a strong completeness guarantee. Our algorithm is currently under implementation as a sub-module of MOAINA (Monitoring Agent for Intelligent Network Analysis) a product by NUST Institute of Technology and Communication Technologies.

4. ARCHITECTURE

The architecture (Fig-1) contains two main modules, the discovery module and the display module. The layer responsible for communication between both modules is the topology hierarchy generation and topology object persistence layer. The focus of this paper is on the discovery module. First of all, discovery of one subnet will be discussed and then it will be extended to multiple subnets. The discovery module takes community string and IP of the NMS. The Initialization layer gets the NMS IP and community string, initiates the IP Generation layer and creates a thread pool that can be used by layers working under the initialization layer like IP generation layer, active probing layer and device status check layer. An intelligent (discussed in section 5.3) IP Generation layer is created. Its reads the router ARP cache and depending on the IP's in the ARP cache, it intelligently takes decision on the IP Ranges to query. This intelligent algorithm is implemented to query those IP addresses whose probability of existence is higher. Now we have generated the high priority IP ranges in our network to query, but we can not ignore the other IPs so IP Generation Layer initiates another process which is Active Probing. This process will be responsible for finding more devices in the network and check for their availability. IP Generation layer initiates three more processes: device status check, device type check and device store fill. All these processes along with the active probing will run in parallel. Device status check, this process will send ICMP echo request to each IP address which is recommended by the IP Generation layer and Active probing process. The purpose of sending ICMP echo requests is to check the status of the device as well as obtaining the IP to MAC mapping of that particular device in ARP cache of NMS. This layer will also be responsible for the invocation of another layer i.e. the server check layer. This layer's responsibility is to further improve the discovery by checking each IP and ports so that we can find any DNS, Web, file or any other server inside our LAN and query those servers to get the ARP cache. From the ARP cache we can get more mappings (As many machines inside the network access resources from the LAN like DNS, file and web servers). Now once we have got the IP and MAC of the devices, we need to know what type of network device it is. The device type check process will send a request to the IP from the device status check layer mapping generated and check what type of device it is router, switch, computer, network printer etc. Lastly every thing is stored in its related store. For this purpose device store fill process will do the related work. It will insert the devices according to their type into their specific stores. Extending the approach to multiple subnets is simple. We will query the router to get IP of its interfaces and their related subnet masks. This input will be given to the IP Generator layer then the same cycle will run again. When we send ICMP request to some host across the router, MAC address of machines across the router will not come into the ARP cache of the NMS (As routers block the Mac layer broadcast) so for that we will have to query the router of that particular subnet for its ARP cache. In the next section we will go into technical detail of algorithms.

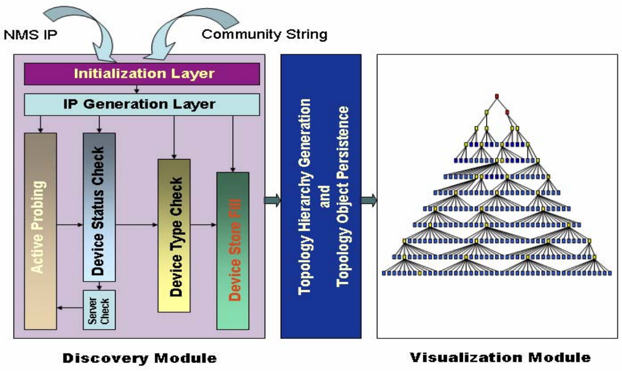

Figure 1: Proposed Architecture

5. ALGORITHMS USED

5.1 Discovery Algorithm Single Subnet

Our algorithm can be executed at any host in the network. Before the discovery starts some parameters are checked (IP address is assigned, ARP command is working, Gateway is set, Ping command is available, Gateway is UP and SNMP agent is available on the Gateway). After checking that, the topology discovery will start. First of all we will get the NMS IP Address, MAC Address, SNMP community String, Default Gateway of NMS. Secondly we will query the Default gateway to get the IP to MAC mapping from the ARP Cache of the router. This mapping will help to intelligently get the ranges of IP's to check in priority (The Algorithm is discussed in the following sections) and in the end it gets the subnet mask of the router. Now we will ping the complete range of IP's which were selected to be checked in priority. Pinging the IP address will give two important pieces of information. First the IP is alive or not and secondly MAC address of that device, from the ARP cache of the NMS. (Remember here we are talking about single subnet). Then we will query all the devices which are alive to check their type. (Types can be switch, computer, router, network printers etc) and in the end we have to accordingly insert the nodes into their respective Stores (i.e. Switch store, Router store, Computer store and printer store). Up till now we have found the computers, routers and Switches in one subnet. Now we have to find the interconnects to discover the network topology. To discover the interconnects we have discussed four algorithms below i.e. switch to switch connection, switch to router connection, switch to host connection and router to router connection.

5.2 Multi-Subnet Discovery

To discover multiple subnets, we get the Next HOP information from the gateway assigned at Network Management Station (NMS) and subsequently from next routers. Then on each subnet discovered, run the single subnet algorithm to get the complete topology. One problem is that getting the Mac addresses of the devices beyond the router. This problem is solved by getting the ARP Cache of the subnet router. Whenever we will ping a remote device which is beyond the router its MAC to IP entry will come into the stub router. We can get the ARP cache of the router to get the MAC address of the desired device.

5.3 Algorithm to find IP's of devices that are alive

Sending ICMP request to all the IP in a network is not feasible in determining the devices that are up in the network because of the large number of possible IP addresses e.g. there will be more than 16 million possible addresses for a Class A network. So in our approach we have an intelligent algorithm for generating a list of IP addresses having a high probability of being assigned to devices in the network. The algorithm utilizes the entries from ARP cache of the router and NMS. Suppose we have two IP address for one subnet in the ARP cache of the router 10.10.5.2 and 10.10.100.1. Then we will get the subnet mask of that subnet address suppose 255.255.255.0 (The network subnet mask is 255.255.0.0). Now from the above given address will remove the bits of last octet from the two IP found in ARP cache (i.e. 10.10.5, 10.10.5). Then in the last octet we will start sending ICMP to 00000001 to 11111110 (i.e. 10.10.5.1 to 10.10.5.254 and 10.10.100.1 to 10.10.100.254). The number of High Priority IP addresses = N * 254 (where N is the distinct first 3 octets in the routers ARP cache). Now this range will become the high priority. Now for second priority we will also change the third octet entry. As we have 5 and 100 initially so now we will get +1 and -1(i.e. 10.10.6, 10.10.4, 10.10.99, 10.10.101) and so on. For the third priority we will query the rest of the address left.

5.4 Switch to Switch Connection

We are using the Direct Connection Theorem with few modifications to determine switch to switch connections. Two nodes are referred to as directly connected if there are no other nodes between them. If a switch A sends packets on port *x* and the packets are received on port *y* of switch B without going through any other device , then switch A and B are directly connected via ports *x* and *y* ,respectively. The Direct Connection Theorem states that two switches *i* and *j* are directly connected via link connected to port x on I and port y on j if and only if where F represents the forwarding set of the switch. (Fig-2).

$$F_i^x \cap F_j^y = \emptyset \text{ and } F_i^x \cup F_j^y = N.$$

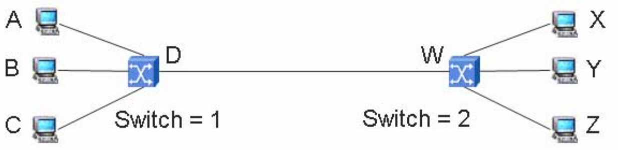

Figure 2: Switch to Switch Connection

S(S1) = { A,B,C,D } Set of Ports of switch 1
S(S2) = { W,X,Y,Z } Set of Ports of switch 2
MAC visible at Port W V(W)= { A,B,C,D }
MAC visible at Port D V(D)= { W,X,Y,Z }
S (N) = {A,B,C,D,W,X,Y,Z}
If (V(D) Ú V(W) = S(N) And V(D) Π V(W) = ø)
Then the Two Switches are directly connected.

The direct connection theorem works fine in normal condition of network but we have identified that it fails if there are cycles (i.e. two switches having more than one connection between them) in the network. (Fig-3)

5.5 Switch to Router Connection

To find switch to router we have to go through the following steps. From the algorithm No 5.1 & 5.2 we got both the IP's and Mac address of the routers in the network. Now when ever we detect a switch we will get its Address Forwarding Table through SNMP and check for the routers MAC address in the AFT. If we find the MAC of router this means that the router and switch are connected otherwise not.

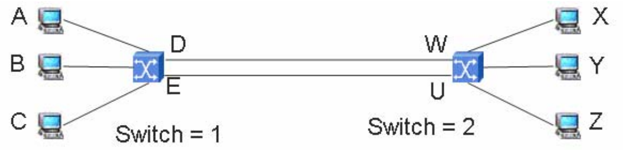

Figure 3: Switch to Switch Connection

5.6 Switch to Host Connection

Get the switches from the store and operate on them to check the connected IP's on them. Get Address Forwarding Table (AFT), to get port to MAC mapping of the switch. For example.

00-d0-b7-89-91-80 = port 2
00-d0-b7-89-92-67 = port 3

Check the IP to MAC mapping (generated in algorithm 5.1) to get the IP's corresponding to the MAC address in the address forwarding table. Now we will know that which devices are connected to the switch.

5.7 Router to Router Connection

The routing table in the router contains the information about the next HOP. In our algorithm we use Next Hop, default gateway and traceroute information to check the connected routers. Traceroute information is used, as routing table may contain entry of a router which at multi-hop distance.

5. CONCLUSION

Physical topology information plays a crucial role in enhancing the manageability of modern IP networks. Despite the importance of the problem, earlier research works that claim to discover the physical topology accurately are totally SNMP based. In this paper, we explain a detailed algorithm for discovering the topology map of IP Network using ICMP echo request/reply and SNMP-MIB and also the supporting algorithms which could help to improve the efficiency of our algorithm. Our algorithm does not rely totally on SNMP-MIB information as usually it is not widely supported by all the computers in the network and enabling SNMP agent on all the computers in large networks is not practical. Instead we propose an approach where SNMP agent is enabled only on the routers, managed switches and network printers. Rest of the network computers need not have SNMP enabled. Our discovery includes routers, computers, switches both managed and unmanaged, network printers and other IP enabled devices, along with physical connectivity relationships that exist among entities in a communication network. We have implemented our algorithm which has been tested over our research network, Kyung Hee University South Korea, CERN and COMTEC Japan. The results clearly validate our methodology, demonstrating the accuracy and practicality of the proposed algorithms. In future we plan to extend this algorithm to discover topology of adhoc networks using software agents.

REFERENCES

[1] D. T. Stott, “Snmp-based layer-3 path discovery,” Tech. Rep. ALR-2002-005, Avaya Labs Research, Avaya Inc., Basking Ridge, NJ, 2002.

[2] R. Siamwalla, “Discovering Internet topology.”, 1999.

[3] H. Burch, “Mapping the Internet,” *IEEE Computer*, vol. 32, pp. 97–98, Apr. 1999.

[4] R. Govindan, “Heuristics for Internet map discovery,” in *Proc. of the 2000IEEE Computer and Communications Societies Conf. on Computer Communications (INFOCOM-00)*, Mar. 26-30, 2000.

[5] B. Huffaker, “Macroscopic analyses of the infrastructure: Measurement and visualization of Internet connectivity and performance,” in *Proc. of PAM2001–A workshop on Passive and Active Measurements*, (Amsterdam, Netherlands), Apr. 23-24, 2001.

[6] W. Zhao “A method for heterogeneous network discovery.” Internal Technical Report, Avaya Labs Research, Avaya Inc., Dec. 2001.

[7] David T. Stott “Layer-2 Path Discovery Using Spanning Tree MIBs” Avaya Labs Research, Avaya Inc. 233 Mount Airy Road Basking Ridge, NJ 07920 March 7, 2002.

[8] Akshay Adhikari “Operational Layer 3 Topology” Avaya Labs Research, Basking Ridge, NJ _ akshay, ld, jmeloche, August 12, 2003

[9] R. Siamwalla “Discovering Internet Topology” Cornell Network Research Group Department of Computer Science Cornell University, Ithaca, NY 14853

[10] Yigal Bejeano, Yuri Breitbart_ , Minos Garofalakis, Rajeev Rastogi “Physical Topology Discovery for Large Multi-Subnet Networks” Bell Labs, Lucent Technologies 600 Mountain Ave., Murray Hill, NJ 07974.

[11] Bruce Lowekamp David R. O'Hallaron Thomas R. Gross “Topology Discovery for Large Ethernet Networks”

[12] Kapil Bajaj D. Manjunath “Intranet Topology Discovery Using Untwine” Indian Institute of Technology, Bombay Powai Mumbai 400 076 INDIA

[13] Yuri Breitbart, Minos Garofalakis, Ben Jai, Cliff Martin, Rajeev Rastogi, and Avi Silberschatz “Topology Discovery in Heterogeneous IP Networks: The *NetInventory* System”