Research White Paper

WHP 188

October 2010

Physical Path Tracing for IP Traffic using SNMP

Yang Xiao

BRITISH BROADCASTING CORPORATION

BBC Research White Paper WHP 188

Physical Path Tracing for IP Traffic using SNMP

Yang Xiao

Abstract

Large-scale enterprise IP networks often involve complex topology consisting of routers, bridges and hosts. Due to multiple redundancies and minimized Single Point of Failure (SPOF), sometimes it is difficult to determine the actual path of a packet even with the most up-to-date physical connectivity diagram. This presents a great challenge for troubleshooting end-to-end traffic flows.

This paper describes a novel solution using standard SNMP and a number of powerful algorithms developed for processing MIB data to discover unknown network devices and their physical interconnections, including those between bridges. In the topology diagram delivered by this solution, physical connections are determined at port-level precision. When this is combined with layer3 routing information, the entire physical path of an IP packet between any two hosts on the network can be traced.

This paper gives detailed elaboration for all algorithms. It also introduces another important feature of the solution, which is the summarization of multicast groups and their routes, which is derived only from the final routing decision of the routers and therefore completely independent of any multicast routing protocols or manual configurations.

This document was originally published in the proceedings of 2010 International Conference on Information and Network Technology (ICINT 2010).

Additional key words:

White Papers are distributed freely on request.

Authorisation of the Head of Operations, R&D is required for publication.

BBC Research White Paper WHP 188

Physical Path Tracing for IP Traffic using SNMP

Yang Xiao

1 Introduction

Network administrators around the world face a similar challenge when traffic between two hosts on the network suffers delay: to locate the connection bottleneck on a specific device port. Knowledge of network connectivity is essential for this task. However it is extremely difficult and costly to keep this information up-to-date, often due to the absence of a centralized database for the network infrastructure [1].

There have been many existing tools for probing unknown devices and extracting connections and routes, such as Ping, Traceroute, DNS zone transfer [2] and ARP, but none is able to produce an overall picture of the network. To discover the full network topology, a large number of different types of data need to be obtained from the devices. Simple Network Management Protocol (SNMP), as an industry standard for accessing device information, has been widely-adapted as the basic mechanism for data extraction.

System variables for network devices are stored in their local memory in Management Information Base (MIB) format. Each MIB contains data about a specific aspect of the device. SNMP-compliant network devices support a number of standard MIBs as defined in various RFCs.

However there is no standard MIB defined for discovering topology. Although CISCO defined the PTOPO-MIB [3] for this purpose, it has not been widely supported by other manufacturers. Various studies on network topology discovery by manipulating data from existing standard MIBs, have achieved limited success.

G. Mansfield et al. attempts to derive the overall topology from the configurations of the routing protocols on individual routers [1]. However this method only discovers layer3 topology of the network and any static route present on a router will upset the algorithm. One advantage of this approach is the capacity to detect WAN topology as it uses BGP-MIB, which contains information on the Border Gateway Protocol (BGP) configuration. However routers configured with BGP normally locate at boundaries between external networks and SNMP traffics are normally not permitted due to security concerns. Therefore the benefit of WAN topology can hardly be realized anyway.

Having identified the disadvantages of methods proposed by R. Siamwalla et al. [2] and B. Yuri et al. [5], S. Pandey, et al. [4] proposed a practical solution based on the work of Lowekamp et al. [6]. Nevertheless although an excellent algorithm for network device discovery was developed, the algorithm for layer2-to-layer2 (L2-L2) connectivity is limited to single connections between two bridges. The algorithm fails in cases such as daisy-chains, rings or fully-meshed groups of three or more bridges. In addition, the algorithm for L2-L3 connectivity involves using spanning tree information, which will cause the algorithm to fail when the spanning tree is disabled on any of the bridges.

Hence to provide a robust solution for finding topology, which is independent of all those factors and limitations mentioned above, this study proposes the following advanced algorithms:

- Revised device discovery algorithm expanding from the work of S. Pandey, et al. [4].
- Layer2 Connectivity Theorem, which is enhanced from the L2-L2 connectivity discovery algorithm.
- L2-L3 and L3-L2 connectivity discovery methods.

With these algorithms, which will be explained in Section II, it is possible to discover the physical topology for any unknown network of any level of complexity.

Section III explains a new algorithm for discovering the full physical path of any IP packet, using the physical topology in conjunction with layer3 routes.

This study also developed methods for discovering multicast groups and summarizing their routing information. By combining the multicast information with the knowledge of physical paths, all multicast streams can be monitored. The algorithm involved is elaborated in Section IV.

A software engine written in Java implementing the algorithms and methods is also completed as part of this study. The details of the software and testing results are given in Section V.

Overall this study has established an integrated platform for monitoring both end-to-end and multicast streams. There is much potential for further developments towards a fully comprehensive and universal monitoring solution. These future works are discussed together with the conclusion in Section VI.

2 Physical topology discovery

As pre-requisites, a list of SNMP read-only community strings of all the devices on the network is needed, and all bridges and routers must support SNMP.

The entire process of physical topology discovery is divided into two parts: device discovery and topology analysis, both involve accessing a number of MIB objects, which are system variables stored on the local memory of the network devices. This study only uses the ones from the most widely-adopted standard MIBs.

2.1 Standard MIBs for Physical Topology Discovery

2.1.1 MIB-II

MIB-II (RFC1213) [7] defines a set of basic system variables that every SNMP-compliant device must support. It contains data required to identify and obtain settings for device types, network interface settings, ARP records and layer3 routing configuration.

Although it also provides an option for indicating the physical location of the device, the validity of this data cannot be guaranteed because it depends completely on manual input and therefore is prone to error and obsolescence.

The MIB object “sysDescr” cannot be used as unique device identifier, because manufacturers use the same text string for all devices of the same model.

2.1.2 IP-FORWARD-MIB

The use of IP-FORWARD-MIB (RFC 2096) [8] was not mentioned in the work of S. Pandey, et al. [4]. Nevertheless it is discovered by this study that Cisco [9] and Alcatel [10] have stopped supporting the “ipRouteTable” defined by MIB-II in there lasts range of products. Instead, the “ipCidrRouteTable” defined in IP-FORWARD-MIB is used.

The main advantage of “ipCidrRouteTable” is its compatibility to Classless Inter-Domain Routing (CIDR), which recognizes the fact that IP routes can have the same network number with different network masks.

Although it was proposed in an earlier version of IP-FORWARD-MIB (RFC 1354) [11] that the “ipRouteTable” defined by MIB-II be deprecated and replaced by one defined in this MIB, it is only supported by a few manufacturers. The current version of the MIB, RFC 2096, states that this MIB is only an “update” [8] to MIB-II. In fact “ipRouteTable” defined by MIB-II is still largely being used by many manufacturers.

2.1.3 BRIDGE-MIB

BRIDGE-MIB (RFC 4188) [12] defines a table called the “dot1dBasePortTable”,

The older version of this MIB, RFC 1493 [13] referenced in the work of S. Pandey, et al. [4] has now been deprecated and replaced with RFC 4188.

There is a supplement to RFC 1493 called Q-BRIDGE-MIB (RFC 2674), which was also referenced in the work of S. Pandey, et al. [4]. Q-BRIDGE-MIB defines two MIB modules for handling the new features from the IEEE 802.1D-1998 MAC Bridges and the IEEE 802.1Q-1998 Virtual LAN (VLAN) standards [14]. However it is out of the scope of this study and therefore not used.

TABLE I. MIB Objects for Physical Topology Discovery

MIB / Table	Object	Device Discovery	Topology Analysis
MIB-II / (Leaf Objects)	sysDescr	√	-
	sysServices	√	-
	ipForwarding	√	-
MIB-II / ifTable	ifIndex	√	√
	ifDescr	√	-
	ifPhysAddress	√	√
MIB-II / ipAddrTable	ipAdEntAddr	√	√
	ipAdEntIfIndex	√	-
	ipAdEntNetMask	√	√
MIB-II / ipRouteTable	ipRouteDest	-	√
	ipRouteIfIndex	√	-
	ipRouteNextHop	√	√
	ipRouteType	√	√
MIB-II / ipNetToMediaTable	ipNetToMediaIfIndex	√	-
	ipNetToMediaPhysAddress	-	√
	ipNetToMediaNetAddress	√	√
IP-FORWARD-MIB / ipCidrRouteTable	ipCidrRouteDest	-	√
	ipCidrRouteMask	√	√
	ipCidrRouteNextHop	√	√
	ipCidrRouteIfIndex	√	-
	ipCidrRouteType	√	√
BRIDGE-MIB/dot1dTpFdbTable	dot1dTpFdbAddress	-	√
	dot1dTpFdbPort	-	√
	dot1dTpFdbStatus	-	√

√: Required for the process; -: Not required for the process

2.2 Device Discovery

2.2.1 Starting Point

SNMP messages are transmitted as UDP packets in layer3. Only devices with valid IP addresses can communicate using SNMP. In a typical monitoring scenario, one device known as the Network Management Station (NMS), running the Network Management Application (NMA), sends SNMP messages to other devices running the Management Agent (MA) and receives replies from them. Therefore in order to start the discovery process, a valid IP address on the network must be given. When this is unknown a loopback address is used to start the process from the NMS itself, provided the NMS is also running a MA.

2.2.2 Router and Logical Neighbor Detection

Two groups of MIB objects are used for discovering new devices: the “ipRouteTable” and “ipNetToMediaTable”. The “ipRouteTable” contains layer3 routing information, also known as Next Hop, which is used for discovering the IP addresses of routers connected to a device. Once a new router is found, the Next Hop of this router leads to the discovery of another router. The

“ipNetToMediaTable” contains ARP records, which are used for discovering IP addresses of other logical neighbours of the device. Using the ARP record of all devices the recursive algorithm could potentially span over the entire network.

Discovery through Next Hop is preferred over ARP, as it leads to the discovery of new routers and therefore has higher chance of discovering more devices. The other more important reason is explained in the next chapter.

Therefore whenever a new device is found, either through Next Hop or ARP from the previous device, the first data to look at is always the Next Hop of the current device. The discovery through ARP only starts when no new router has been found through Next Hop.

When the discovered IP address does not respond to SNMP, it is possible that either the device does not support SNMP, or it has been disconnected. If this IP address is a Next Hop, indicating the device is a router, it is more likely that the router is disconnected because it is highly uncommon for a router not to support SNMP. For the same reason given in the next chapter if the device is discovered through ARP, it is more likely to be a connected device without SNMP support.

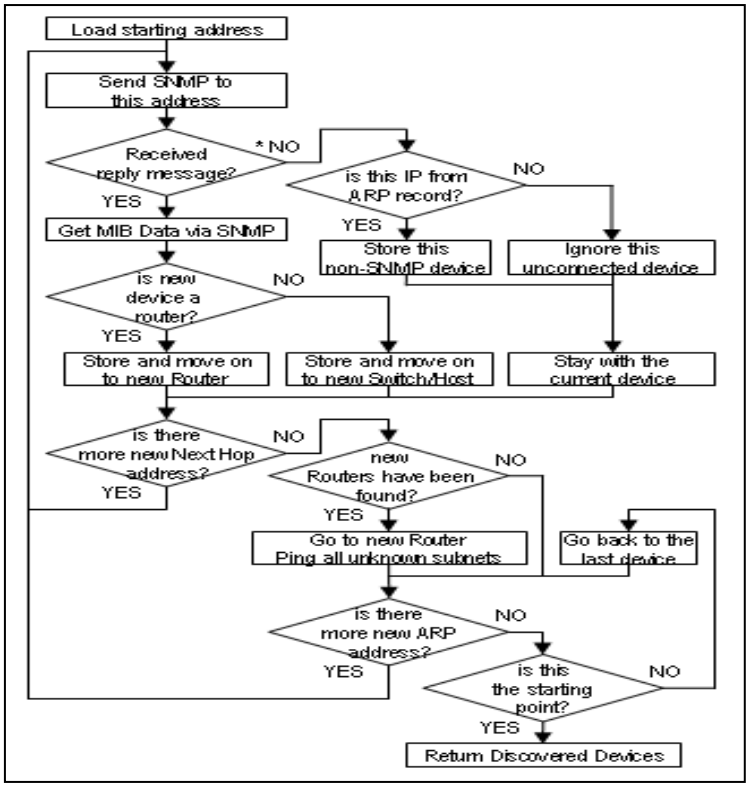

Figure 1. Device Discovery Algorithm. *: Never happens at the start

2.2.3 Refreshing ARP Record Using ICMP

While the layer3 routing information remains constant on a device, the ARP data only resides on the memory for a limited period of time. This causes a problem for device discovery as non-communicating neighbours within the same subnet will disappear from each other’s ARP records.

This problem has a direct impact on discoverability of the bridges, which under normal operations are transparent to the IP layer. The only way SNMP can communicate with a bridge is via its management port, which has a valid IP address. But unless the bridge is constantly being monitored, the management port usually remains idle.

To solve this issue, some mechanism for re-establishing the ARP neighbourhood needs to be deployed together with the recursive algorithm. In this study, a single ICMP (ping) packet is sequentially sent to all the unknown addresses on a known subnet. This will refresh the ARP record for a brief period of time without straining the network.

2.2.4 Device Types

2.2.4.1 Router

Routers are devices that forward IP packets. Whether a device is a router is reflected in the value of “ipForwarding” object in MIB-II. Therefore Gateway PCs are also considered as routers. Routers with switch ports are known as Multilayer Switches.

2.2.4.2 Bridges

Also referred to as Layer2 Switches, bridges are devices that do not forward IP packets and do not have any data entry in the “ipRouteTable”.

2.2.4.3 Hosts

Hosts are network-attached devices that do not forward IP packets but have data entries in the “ipRouteTable”.

2.3 Topology Analysis

All hosts on an IP networks are either connected to a multilayer switch or a bridge, which are referred to as Branch Points in this study.

2.3.1 Overall Algorithm for Branch Point Topology

The process for finding branch point topology starts by determining a hypothetical “core” of the network, which is the router that has been the most popular Hext Hop destination of all other routers. The more a device is routed to by other routers, the more likely this device is located at the centre of the network. This may not be the designed core router of the network but serves as a very good starting point.

```
int max_count = 0;
Router core = new Router();

for (every CURRENT_Router) {
   int current_count = 0;
   for (every OTHER_Router) {
      for (every "ipRouteNextHop" entry) {
         if ("ipRouteType" = indirect) {
            if (CURRENT_Router's IP address list.contains("ifRouteNextHop")) {
               ++current_count;
            }
         }
      }
   }

   if (current_count > max_count){
       core = CURRENT_Router;
       max_count = current_count;
   }
}

return core;
```

Figure 2. Algorithm for finding hypothetical network core in Java syntax.

With the core chosen, a neighbourhood structure around it is built up by finding directly connected branch points. Then for each of the neighbours, the same process continues until all branch points have been placed onto the core neighbourhood. There is also a “catch-all” function for branch points that cannot be reached from the core. Each of these branch points is expanded to reach other branch points that are already found on the core neighbourhood.

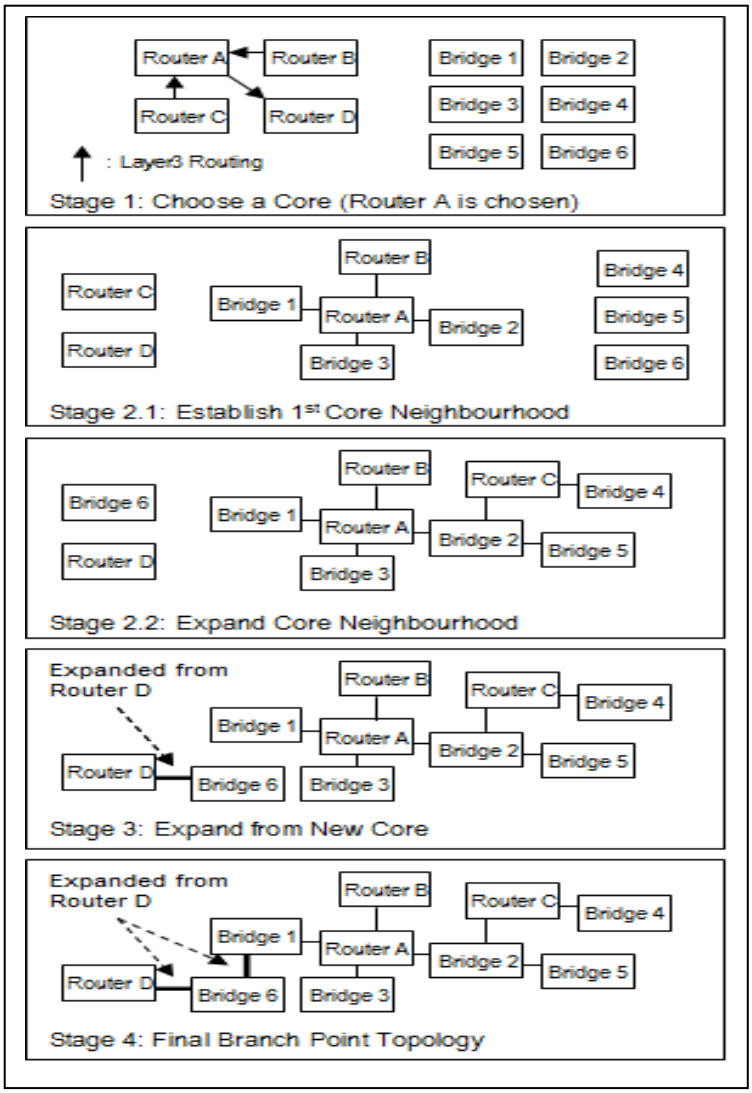

Figure 3. An example of finding the Branch Point Topology.

2.3.2 Specific Algorithms for Branch Point Connectivity

The backbone of a network consists of interconnected routers and bridges. Due to the complex nature of the layer2 connections, it is unrealistic to apply a universal algorithm with guaranteed accuracy and success rate. In order to be robust in terms of handling practical issues such as presence of a hub, incomplete MIB data or incorrect manual configurations, an element of fuzzy logic is introduced to the algorithm. Instead of looking for an exact match, the algorithm will first select a group of possible candidates against a certain threshold of criteria. Then it applies filters to the selected candidates to eventually find the most likely match.

2.3.2.1 Determining Connection Type

Routers, especially multilayer switches, have three basic types of ports: routed ports, switch ports and trunk ports.

- A routed port is a physical interface that has an IP address assigned to it, similar to one on a host. A routed port could also be a management port of a bridge.
- A switch port is a physical interface that does not have an IP address but belongs to a virtual interface that has one. Switch ports configured for the same subnet, also known as a Virtual LAN (VLAN), function in exactly the same way as if on a bridge.

- A trunk port is a physical interface that carries traffic for multiple VLANs. This is a special type of port that only exists between multilayer switches for traffic aggregation.

For clarity, various types of connections are differentiated on the port-level.

- “L2-L2”: connections between two switch ports.
- “L2-L3”: connection from a routed port to a switch port.
- “L3-L2”: inverse connection of L2-L3.
- “L3-L3”: connection between two routed ports.

The algorithm for examining physical connectivity starts by examining the type of connection a port is likely to have. Then based on the connection type, an appropriate algorithm is used for finding the physical neighbour of this port.

For example if a port does not appear on the MAC address Forwarding Information Base (FIB), it is likely to be a routed port. Then looking at the ARP record of the port, if it has multiple data entries, the port is likely to be connected to a switch port of another branch point. Therefore the connection to this port is likely to be a L3-L2 connection.

TABLE II. DECISION TABLE FOR DETERMINING CONNECTION TYPES

Number of a Port’s :		Entries in Forwarding Information Base (FIB)		
		0	1	n
ARP Entries	0	* Invalid Port	L2-L2	L2-L2
	1	L3-L3	L2-L3	L2-L2
	n	L3-L2	L2-L2	L2-L2

* This could be the case of either an unconnected physical port or a virtual interface such as a VLAN

2.3.2.2 L2-L2 connectivity Analysis

This is the area heavily discussed by various other studies. All proposed methods involve processing data in the FIB and local MAC addresses of two bridges to determine whether they are physically connected. Although these methods are limited to networks with specially designed architectures, they point to the right direction, which this study has followed and developed a general algorithm to successfully work under complex situations involving trunk ports and daisy chains.

The FIB of a bridge or multilayer switch specifies the port to which a layer2 packet with an external destination MAC address should be forwarded. The FIB data is contained in “dot1dTpFdbTable” in BRIDGE-MIB, with the external destination MAC address of the layer2 packet stored in a “dot1dTpFdbAddress” object and the index of the forwarded port in “dot1dTpFdbPort”.

The set of all “dot1dTpFdbAddress” entries on a bridge with the same “dot1dTpFdbPort” is a collection of all the destination MAC addresses of all layer2 packets sent by this port.

On the other hand, all layer2 packets arriving at a port on a bridge will be forwarded onto the other ports. Therefore the set of all “dot1dTpFdbAddress” entries less those forwarded to a particular port, plus the MAC address of that port, is a collection of all possible destinations of the layer2 packet received by this port.

In a physical port-to-port connection, all packets sent by a port must be received by the connected port. Hence by comparing the layer2 packets sent and received by two ports, the L2-L2 connectivity between two ports can be successfully established.

LAYER2 CONNECTIVITY THEOREM:

"If port x on bridge A is connected to port y on bridge B, then the set of all the "dot1dTpFdbAddress" entries of A with their "dot1dTpFdbPort" value equal to x, less any entry equal to the 'ifPhysAddress' of B, should be equal to the set of all "dot1dTpFdbAddress" with their "dot1dTpFdbPort" values unequal to y."

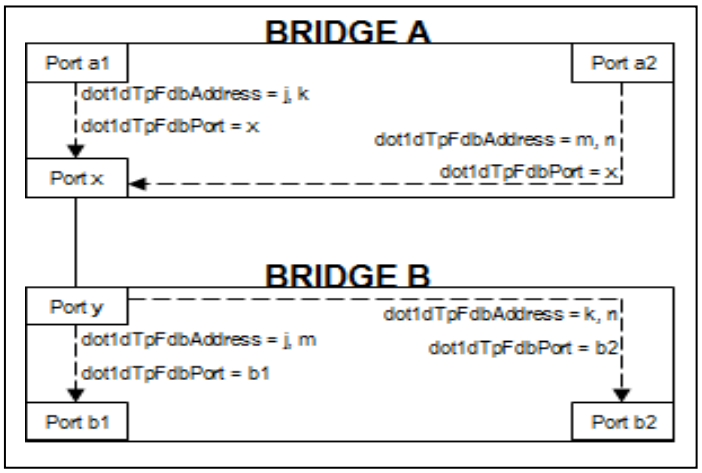

Figure 4. Layer2 Connectivity Theorem.

Let D be the set of all destination MAC addresses of the packets leaving x, L the set of all "ifPhysAddress" on all ports of B, and D´ the set of all "dot1dTpFdbAddress" with their values unequal to y:

$$D - D \cap L = D' \quad (1)$$

This theorem applies for all L2-L2 connections including those between trunk ports or bridges with multiple redundancies enabled with Spanning Tree.

As mentioned earlier, all algorithms employ fuzzy logic to accommodate incomplete MIB data. Therefore instead of looking for the perfect match between both sides of the equation, a threshold can be specified and expressed as:

$$(D - D \cap L) \cap D' = (D - D \cap L) \,||\, D' \quad (2)$$

Hence the theorem could also be written as:

*"If port x… should be **A SUBSET OF** … or vice versa."*

However the fuzzy logic causes problem when determining connection involving 3 or more bridges in daisy chain or ring structures. In the example shown in Figure 5, a3 can be determined by the fuzzy algorithm as being connected to either port b1 or c1. This is because the sum of all possible destinations of the packets leaving a3 is a subset of all "dot1dTpFdbAddress" on port b1 as well as being subset of those on c1.

This problem is solved by further investigating the "dot1dTpFdbAddress" entries on the ports, as c1 is also forwarding packets to MAC address o, which can only be sent from b2.

This fuzzy algorithm also accommodates the presence of hubs. When a hub is present, one port on a branch point will seem to be connected to multiple hosts, as determined by the algorithm.

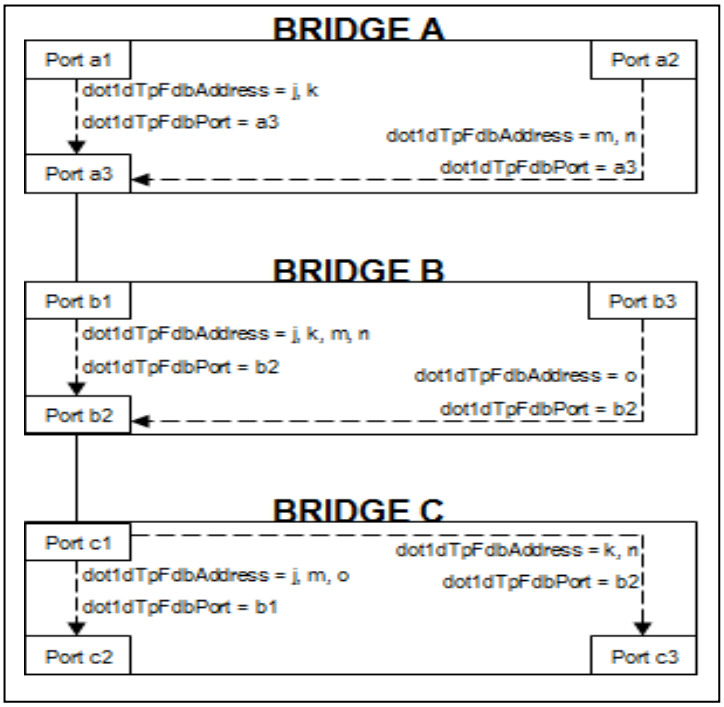

Figure 5. Bridges connected in a daisy chain.

2.3.2.3 L2-L3 and L3-L2 connectivity Analysis

A L2-L3 connection only exists between a switch port and a routed port, where the MAC address of the routed port is the only “dot1dTpFdbAddress” entry in the bridge’s FIB with the “dot1dTpFdbPort” value pointing to the switch port.

L2-L3 connectivity is discovered using the algorithm in Figure 6, which can be inversely applied for discovering the L3-L2 connectivity.

```
1) Find the number of entries in the bridge's FIB with
   "dot1dTpFdbPort" equal to the "ifIndex" of the port;

2) If only one such entry found, check the number of
   entries in the ARP table of the port;

3) If only one entry ARP found, check if the
   "ipNetToMediaPhysAddress" matches the "ifPhysAddress"
   of the other port;

4) If matches, the L2-L3 connection is determined.
```

Figure 6. L2-L3 Connectivity Discovery Algorithm.

3 End-to-end Physical Path Tracing

To trace the end-to-end physical path between two hosts on the network is the ultimate purpose for building a complete network diagram. Nevertheless without correctly utilizing the layer3 routing information on all the routers it is still impossible to accurately identify the path an IP packet must travel from source to destination.

Figure 7 is an example of a typical modern network of 3 VLANs. VLAN 1 and 2 are routed via Router A and VLAN 2 and 3 via Router B. Bridge 2 is a backbone switch with ports configured for all VLANs, where port 2-3 is connected to port 1-3 on Bridge 1 in VLAN 1 and port 2-4 to port 3-3 on Bridge 3 on VLAN 3.

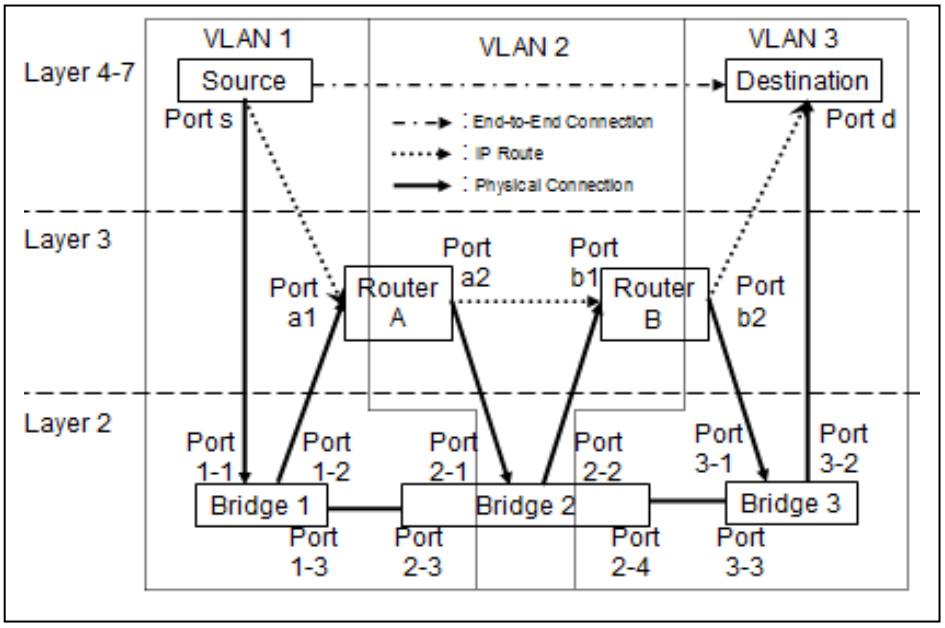

Figure 7. An example of a typical modern network of 3 VLANs.

In terms of physical ports, any IP packet from the source in VLAN 1 to the destination in VLAN 3 would follow this path:

s >>1-1>1-2 >>a1>a2 >>2-1>2-2 >>b1>b2 >>3-1>3-2 >>d

where ">>" indicates a physical cable and ">" indicates the internal switching or routing on a device from one port to another.

However on a simple physical topology diagram without layer3 information the path could easily be misinterpreted as:

s >> 1-1>1-3 >> 2-3>2-4 >> 3-3>3-2 >> d

3.1 Determining Layer3 Routes

To correctly identify the physical path for any end-to-end connection, it is important to identify the layer3 routes between the source and destination. By investigating the source's Default Gateway setting in the "ipRouteNextHop" object, the first router on the layer 3 route can be identified easily. For hosts without this setting, their network numbers can be deduced from the IP address in the "ipAdEntAddr" object and the subnet mask in "ipAdEntNetMask". The router can be found by comparing the host's network number with every router's locally routed subnets, stored in "ipRouteDest" entries with "ipRouteType" equal to 3.

Once the router on the host network is found, by looking for the destination network number or the default route in its routing table, "ipRouteTable" or "ipCidrRouteTable", the next router can be found. Repeat this step until the router in the destination network is found.

It is important that information of the downstream interface to the next router is recorded for every router. For multilayer switches the downstream interface could refer to a VLAN with multiple switch ports. In this case, information about every individual switch port is to be collected and kept with the router.

3.2 Determining Layer2 Paths

Layer2 paths are to be determined individually between every two adjacent routers, as well as between routers and hosts at each end of the route. It is relatively easy to spot the path on a diagram between two points. Nevertheless an algorithm similar to Spanning Tree is required to find the path programmatically.

3.2.1 Layer2 Path between Routers

Within the same VLAN, using the physical topology information, the layer2 connection from the all the downstream ports of the first router is traced to the immediate neighbours, then to the neighbours' neighbours, and so on until the next router on the path is reached.

3.2.2 Layer2 Path between Router and Host

The same algorithm applies for finding the physical path between a router and a host, except that the starting device is always the router. This is to accommodate the fact that a host can be connected to a router via a hub and some of the hosts may not support SNMP.

4 Multicast Group Discovery and Routing Summarisation

4.1 Standard MIBs for Multicast Discovery

All useful multicast information can be obtained from the routers enabled with multicast routing. Some bridges with Internet Group Management Protocol (IGMP) contain information about multicast on a L3-L2 level, which does not contribute towards finding the number of multicast channels available on the network and the number of subscribers of each channel.

To extract the multicast groups and routing information, nine objects in the widely-adopted IPMROUTE-STD-MIB, defined in RFC 2932 [15], are required:

- "ipMRouteSource"
- "ipMRouteSourceMask"
- "ipMRouteUpstreamNeighbor"
- "ipMRouteInIfIndex"
- "ipMRouteRtAddress"
- "ipMRouteRtMask"
- "ipMRouteNextHopIfIndex"
- "ipMRouteNextHopState"
- "ipMRouteScopeNameString"

4.2 Multicast Groups

By extracting the "ipMRouteScopeNameString" data from all routers on the network, a full list of available multicast groups can be made. A multicast address is the textual name that uniquely identifies a multicast group.

Every multicast group contains a number of sources and receivers. It is possible to have multicast groups with no receivers. In this case there will not be any traffic of this group on the network. The IP address and subnet mask of the sources are stored in the "ipMRouteSource" and "ipMRouteSourceMask" objects respectively.

The router IP address and subnet mask for the Rendezvous Point (RP) of a multicast group is found in the "ipMRouteRtAddress" and "ipMRouteRtMask" objects respectively. The IP address of the router, from which a particular multicast stream is received, is found in the "ipMRouteUpstreamNeighbor" object.

4.3 Multicast Channels

For a particular multicast group, a complete channel structure can be established from the sources down to all the routers involved in forwarding the traffic of this multicast group.

By reading values of the “ipMRouteInIfIndex” and “ipMRouteNextHopIfIndex” the physical port where a multicast stream is received from, and the interface where the stream is forwarded to, can be obtained respectively. When a router is enabled with IGMP, the forwarding interface is a physical port; when not, the forwarding interface is a VLAN. In the case of forwarding to a VLAN, the multicast stream is broadcasted on all switch ports in that VLAN.

4.4 Physical Path for Multicast Traffics

Using the mechanism for end-to-end physical path discovery and the information on the channel structure of all multicast groups, every node involved in the delivery of a particular multicast stream can be identified.

The physical path for connections between routers, connections between the source and the RP, and connections between receivers and multicast routers can be determined in exactly the same manner as for an end-to-end connection.

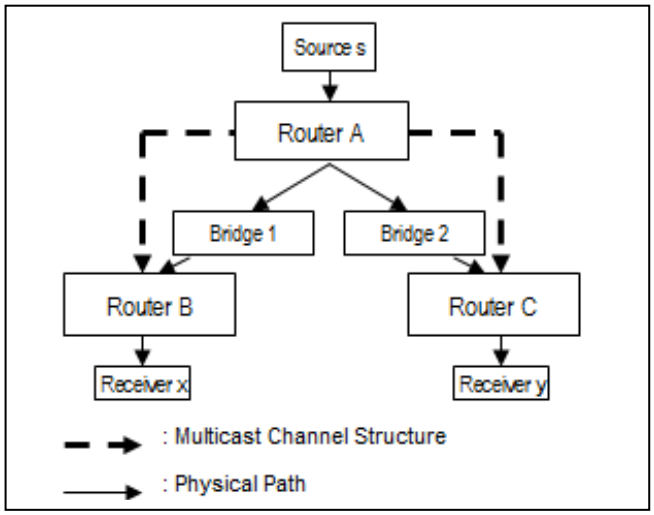

Figure 8. Finding Physical Path for Multicast Traffics.

5 Software Implementation and Test Results

The theorem, algorithms and methods proposed in this study have been implemented in software and tested on various networks with different structures.

The software engine is developed in Java for future web deployments and only uses one external package, snmp4j.jar, for sending and receiving SNMP messages. It uses JGraph.jar and JGraphT.jar for generating visual diagrams of the network.

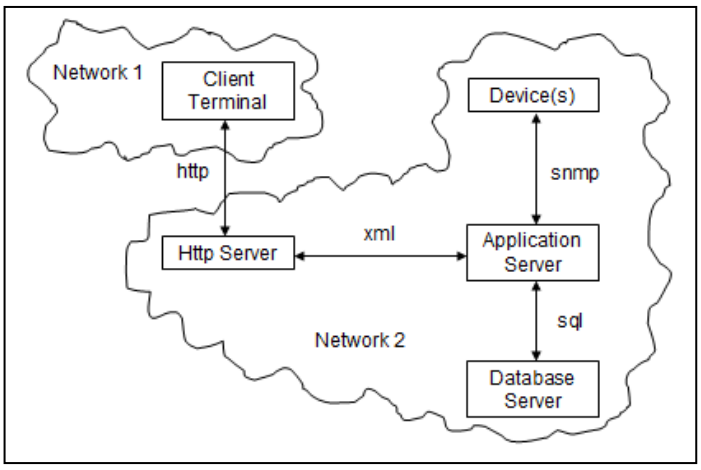

Figure 9. Web service architecture for remote monitoring

5.1 Test 1 – IT network

The software was developed using the Agile Software Development method and rigorously tested on the BBC R&D network on every iteration cycle.

Figure 10. Topology of BBC R&D network

As shown in Figure 10, the network consists of a single router and many bridges and hosts. Every transparent box indicates a device that does not support SNMP. The software is able to correctly analyze the network in just over 30 minutes. This is due to the fact that a large number of hosts do not support SNMP and therefore the software has to spend the longest possible time on timing out. It also took the software a longer to fully analyze the connections.

5.1 Test 2 – Media distribution network

The software was also tested on a media distribution network consisting of 37 routers and over 300 hosts.

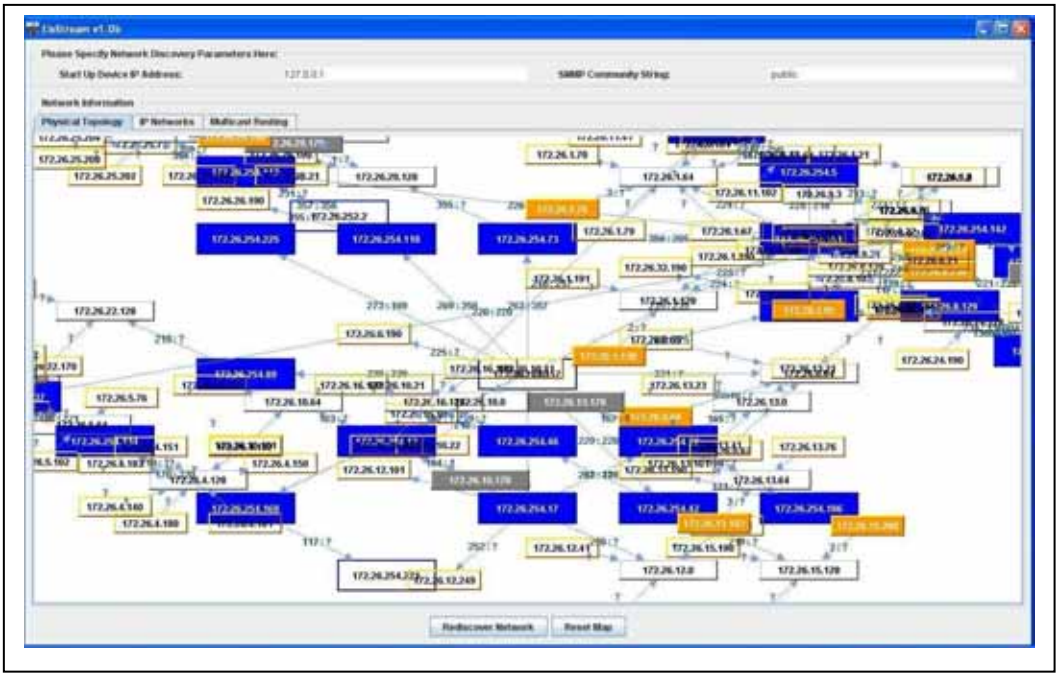

Figure 11. Distribution Network Topology

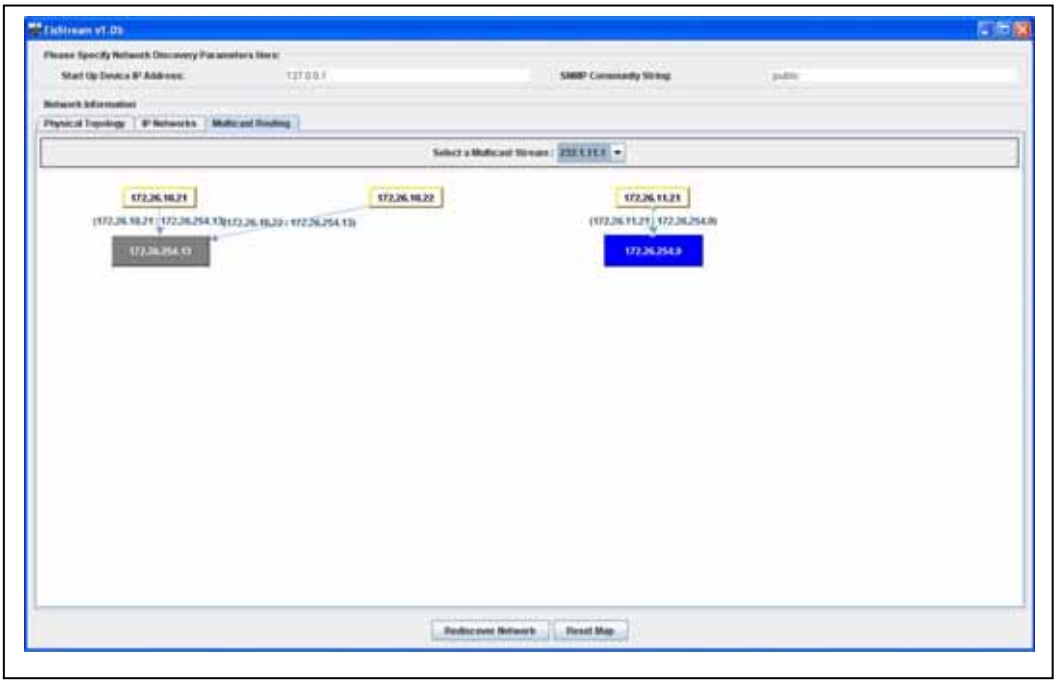

Figure 12. Multicast Channel Structure

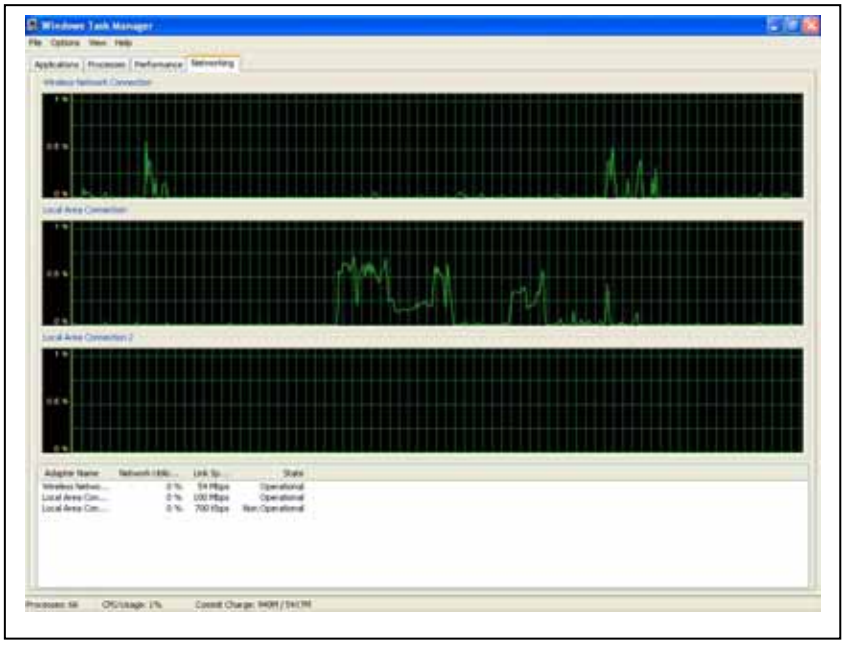

Figure 13. Network Utilization during Discovery

6 Further Developments and Conclusion

This project was originally commissioned as part of the efforts by the IP Measurement Group of the European Broadcasting Union (EBU-IPM) and the Research & Development Department of the British Broadcasting Corporation (BBC R&D) to address the problems in monitoring media streams.

Through the course of investigating the methods for basic functionalities such as device discovery and topology analysis, it became clear that more robust algorithms need to be developed for complex modern networks.

This study successfully established a framework for network inventory and discovery, and path tracing for end-to-end and multicast streams, which sets the foundation for higher-level performance monitoring and troubleshooting.

The key contributions of this study are:

- Enhanced device discovery algorithm;
- Layer2 Connectivity Theorem;
- Physical path tracing;
- Multicast information gathering and summary.

The result is an integrated and comprehensive solution that is based solely on SNMP and four widely-adopted standard MIBs but works on nearly all multi-vendor networks.

A software implementation phase of the solution has finished with beta version released in open source in April 2010. There is potential to develop the software into an API or web service with database access.

7 References

[1] Glenn Mansfield, M. Ouchi, K. Jayanthi, Y. Kimura, K. Ohta, Y. Nemoto, “Techniques for automated Network Map Generation using SNMP”, infocom, pp.473, Fifteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Vol. 1-3), 1996.

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

[3] A. Bierman, K. Jones, "Physical Topology MIB”, RFC2922, IETF, September 2000.

[4] Suman Pandey , Mi-Jung Choi , Sung-Joo Lee , James W. Hong, “IP network topology discovery using SNMP, ” Proceedings of the 23rd international conference on Information Networking, p.33-37, January 21-24, 2009, Chiang Mai, Thailand.

[5] 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.

[6] 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.

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

[8] F. Baker, “IP Forwarding Table MIB”, RFC 2096, IETF, January 1997.

[9] Cisco Systems, “Cisco ASR 9000 Series Routers MIB Specifications”, www.cisco.com, March 2009, pp. 3.5, OL-19099-01.

[10] Alcatel, “Alcatel 7750 SR”, www.alcatel-lucent.com, 2006, pp. 6, 3CL 00469 0397 TQZZA Ed.10 20761

[11] F. Baker, “IP Forwarding Table MIB”, RFC 1354, IETF, July 1992

[12] K. Norseth, Ed., “Definitions of Managed Objects for Bridges”, RFC 4188, IETF, September 2005.

[13] E. Decker, P. Langille, A. Rijsinghani, K. McCloghrie, “Bridge MIB”, RFC 1493, IETF, July 1993.

[14] E. Bell, A. Smith, P. Langille, A. Rijsinghani,K. McCloghrie, “Q-BRIDGE-MIB”, RFC 2674, IETF, August 1999.

[15] K. McCloghrie, D. Farinacci, D. Thaler, "IPv4 Multicast Routing MIB”, RFC 2932, October 2000.