

Network Physical Topology Discovery Algorithm for Ethernet Network

Sudip Sikdar¹, Amit Kr. Jain²

{sudipsikdar,amitkrjain}@bel.co.in

Central Research Laboratory, Bharat Electronics Limited, Ghaziabad (India)^{1,2}

Abstract: *To manage computer network with ease and hassle-free way, it is necessary for Network Management System (NMS) to have up-to-date information of physical network topology. It is not only very cumbersome to keep track of physical topology of computer network manually through NMS; it also becomes near impossible to maintain up-to-date information of the same when the computer network grows gradually. To address this problem, this paper proposes an algorithm for NMS to discover physical network topology automatically in Ethernet network comprising IP devices.*

Key Terms— NMS: Network Management System, MIB: Management information base, LLDP: Link Layer Discovery Protocol, ARP: Address Resolution Protocol, AFT: Address Forwarding Table, STP: Spanning Tree Protocol

I. INTRODUCTION

To manage any computer network consisting of various network elements (switches, routers, servers, workstations, printers etc.) in terms of locating fault, root cause analysis of fault etc., Network Management System (NMS) should have thorough information of how the network elements are connected to each other among themselves or in other words NMS should have information of network topology. Network topology information can be fed into NMS manually in which case it becomes very cumbersome and near impossible to keep track of connectivity information of network elements in large computer network. It also becomes very difficult to keep track of connectivity information of network elements when the network grows gradually to meet the business goal. There are many research papers [1],[2],[3],[4] which have already addressed this problem by proposing various algorithms to discover network topology in Ethernet network but none of them alone is sufficient for addressing the problem fully. To address the problem completely, this paper proposes an algorithm which combines those already proposed algorithm with new algorithm for NMS to discover physical network topology automatically in Ethernet network consisting of backbone network formed by switches & routers and End devices such as servers, workstations, printers, UPS, Laptops and other IP devices. Discovery of network topology begins with discovery of devices [1],[3] and finding the type of devices using & extending the methodology described in [3]. After that it involves finding inter connectivity among Routers, L3 and L2 switches. L2-L2 connectivity is derived using methodologies based on STP described in [2]. Router-Router connectivity is derived using methodologies based on OSPF protocol described in [4]. Methodologies described in [1] to find L2-L2, L2-L3 or vice-a-versa and L3-L3 connectivity does not produce accurate result in network formed by modern L2/L3 switches. Therefore, we have not considered the proposed methodology in [2] and instead adopted a new methodology based on Link Layer Discovery Protocol represented by LLDP MIB in this paper to find L2-L3 or vice-a-versa and L3-L3 connectivity

among L2 and L3 switches. Methodology described in [4] to find connectivity among Routers has been fine-tuned to discover connectivity in area other than backbone area in OSPF as [4] is capable to discover connectivity in backbone area only. Finally, discovery of topology ends with finding host connectivity with switches in which case we have used our own methodology to discover the same.

Remaining paper consists of following sections: section II describes the proposed algorithm along with flow chart. Section III includes experiments and results. Conclusion has been put in section IV and section V includes references.

II. PROPOSED ALGORITHM

In this paper we have proposed two algorithms – 1st one for discovery of topology of Ethernet network consisting of various elements such as Routers, L3 and/or L2 switches and hosts (servers, printers, Laptops etc.) and 2nd one for dynamic discovery of newly connected NE to the already discovered network.

A. Approach for discovery of topology of Ethernet network:

Topology discovery approach is mainly divided into three steps:

1. Discovery of network elements

Network elements are discovered using various methodologies described in [1], [3] in addition to using our own methodology which is based on directed broadcast in which case ping is sent to local broadcast address and replies from different network elements are captured.

2. Categorisation of network elements

Discovered elements are categorised into Router, L3 switch, L2 switch or hosts by finding the type of those elements using methodology described in [3]. Methodology in [3] does not categorise hosts further and tell which is Windows host or Linux host or Virtual Machine and that is why it has been modified further to categorise hosts into VmWare Virtual Machine, UPS and Windows host or Linux host.

3. Finding connectivity among network elements

After discovering network elements and categorisation of the elements based on their device type, connectivity among them is discovered in following ways:

Relevant MIBs are fetched according to the type of devices and then duplicate devices are identified and merged into one by analysing those MIBs. Thus different lists of devices (L3 device list, L2 device list and host list) are formed in this way and then connectivity among these devices are discovered by first finding L3-L3 connectivity among routers using OSPF information. If

OSPF information is not available, then routing information contained in ipRouteTable is analysed to find L3-L3 connectivity. Then L3-L2/L2-L3 & L2-L2 connectivity is discovered by analysing LLDP information if available otherwise STP information is analysed. If STP information is not available then AFT information is analysed to find L3-L2/L2-L3 & L2-L2 connectivity. Finally host connectivity is discovered by analysing AFT information.

OSPF MIB analysis:

There is a concept of area in OSPF protocol and at least one area which is known as backbone area should be configured as area zero. There could be multiple areas as well in which case the backbone area is connected to other area through ABR (Area Border Router). Algorithm to find L3-L3 connectivity using OSPF MIB is described in [4] which is capable to find L3-L3 and/or L3-L2 connectivity in backbone area only but not in other area in OSPF. That is why we have modified the [4] algorithm to discover L3-L3 connectivity in area other than backbone area also. ospfIfTable MIB object of a Router gives information about its interfaces in terms of interface IP address represented by ospfIfIpAddress, interface type represented by ospfIfType and interface area ID represented by ospfIfAreaId which tells the area to which this interface belongs. Finding connectivity of router Rx with other router Ry in a particular area involves several following steps:

1. First area ID is extracted from ospfIfTable MIB object of Rx and checked whether it is that particular area or not. If the area is that particular area, then ospfNbrTable MIB object of Rx is analysed and list of IP addresses of its neighbour routers is found from ospfNbrIpAddress in ospfNbrTable. Now the first IP address is picked from the list and is checked whether IP address of router Ry ($Ry \neq Rx$) is matched with this and if so then its area ID is matched with area ID of Rx by analysing ospfIfTable of Ry. If match is found for router Ry, then router Ry is found to be neighbour of router Rx in that particular area. To check further how Rx and Ry are connected to each other, interface type of Ry is derived from ospfIfType of ospfIfTable of router Ry. If the interface type comes out to be broadcast, then step 2 is followed. If the interface type comes out to be point2point then Ry is directly connected to Rx in that particular area in which case further ospfIfTable of router Rx is analysed to find out which interface of it is connected to router Ry. To find out the candidate interface of Rx, network ID is derived using ipAddressTable of Ry. And using the derived network ID & ospfIfTable of router Rx, IP address of connected point2point interface of router Rx is found out. In this way all other connected point2point interfaces of neighbour routers of Rx are found out for that particular area.

2. If the interface type comes out to be broadcast, then there must be a switch between these two routers in a properly designed network. To find out the candidate switch, first network ID and subnet mask of the subnet is found out from ipAddressTable of the router Ry. Now the candidate switch is found from the list of already found switch using network ID and subnet mask. Further using the same network ID and subnet mask and ipAddressTable of the router Rx, the connected interface of the router Rx with the switch is found. In this way all other connected broadcast interfaces of neighbour routers of Rx are found out for that particular area.

3. If area ID of any interface of router Ry is different from area ID of router Rx, then it can be decided that router Ry is ABR (Area Border Router) whose one interface is in the area to which Rx

belongs and other interface is in another area say area 1. To find connectivity among routers in area 1 the procedure mentioned in step 1 is repeated.

If OSPF is not available, then routing table represented by ipRouteTable or ipCidrRouteTable is analysed to discover L3-L3 connectivity.

Analysis of Routing information:

If the value of ipRouteType in ipRouteTable of local router is 4 (i.e. indirect) then corresponding entry in ipRouteNextHop contains IP address of directly/indirectly connected remote router device. To further determine whether the corresponding ipRouteNextHop entry is of directly connected remote device or not, corresponding entry for ipRouteDest in ipRouteTable of local router is checked to see if it belongs to remote router or not. If so then ipRouteTable of remote device is analyzed and checked to see if ipRouteNextHop entries of ipRouteTable of remote device contains IP address of the local router. If so then corresponding entry for ipRouteDest of ipRouteTable of remote device is checked to see if it belongs to local router or not. If so then these two routers are connected directly.

Analysis of LLDP MIB:

If the value of MIB object "lldpRemPortIdSubtype" in lldpRemTable in lldpRemoteSystemData is 3 (i.e. MAC Address) then lldpRemPortId contains MAC address of the connected remote device. Thus the remote device which is physically connected to the local device is found. Once the remote device is found, its port number to which the local device is connected is derived by lldpLocPortId in lldpLocPortTable MIB object which contains the MAC address of its own particular port. lldpLocPortTable in LLDP MIB gives the mapping of the MAC address to port ifIndex. Now applying the similar logic the port number of the local device is derived by analysing lldpRemTable of remote device and its own lldpLocPortTable. Thus the physical topology information is derived from LLDP MIB.

If LLDP information is not available and STP information is available, then dot1dStpPortTable containing STP information is analysed to find topology information.

Analysis of STP information:

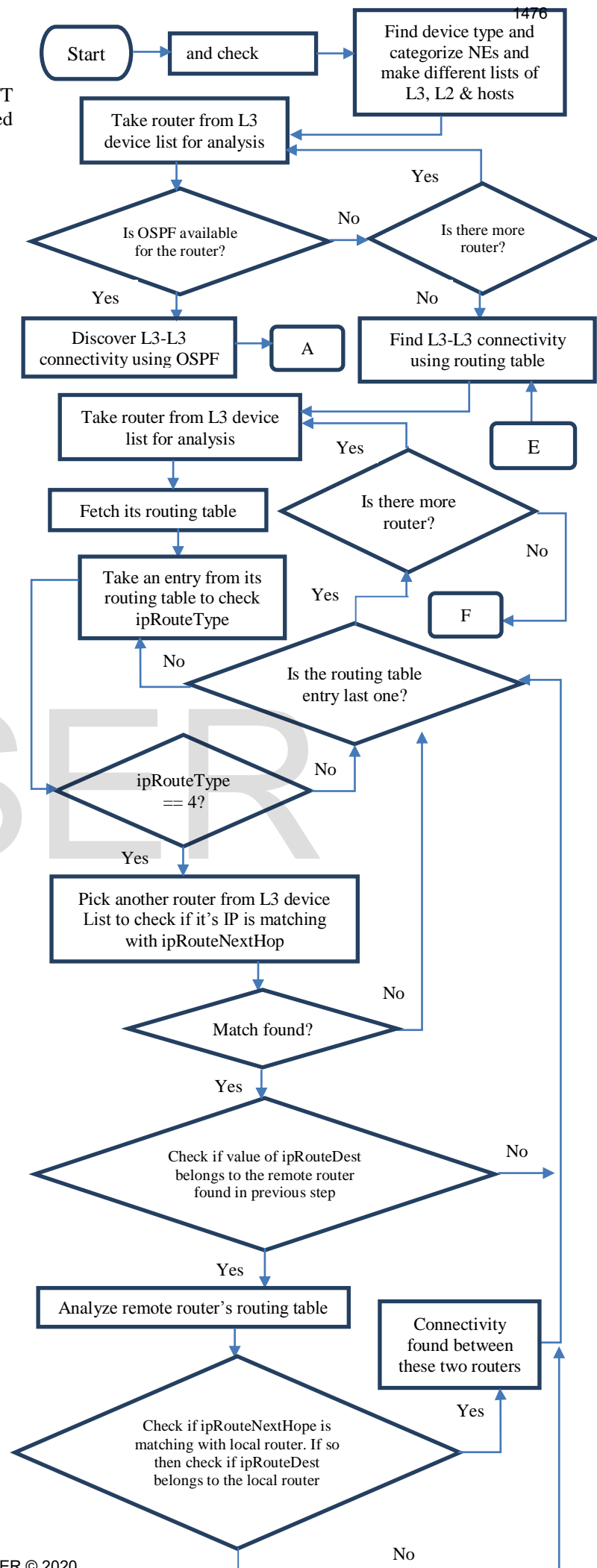
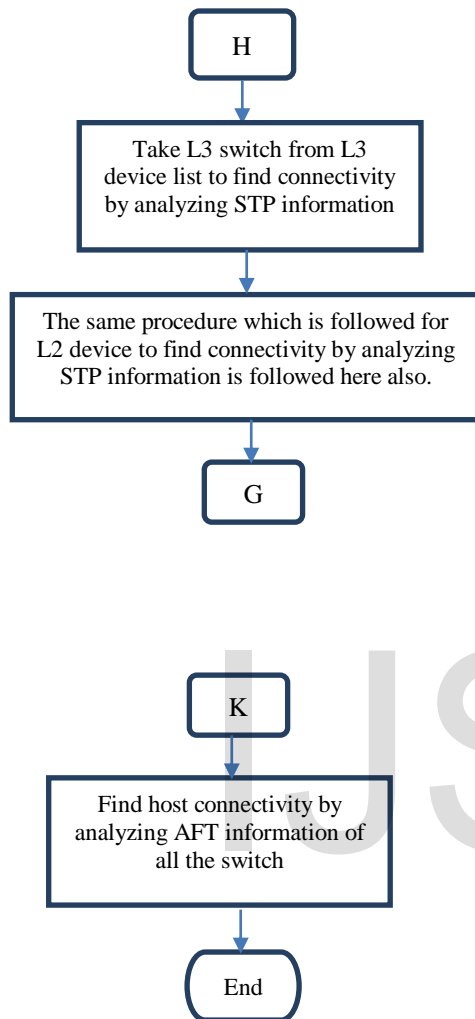
Analysis of STP information is described in [2]. If dot1dStpPortState of dot1dStpPortTable is in forwarding state, then dot1dStpPortDesignatedBridge contains the MAC address of the connected remote device. Once the connected remote device is found then its port number which is connected to the local device is derived from dot1dStpPortDesignatedPort. Next the port number of the local device is derived from the dot1dStpPort.

If LLDP information and STP information both are not available and AFT information for L2/L3 switch is available, then dot1dTpFdbTable containing AFT information is analysed to find topology information.

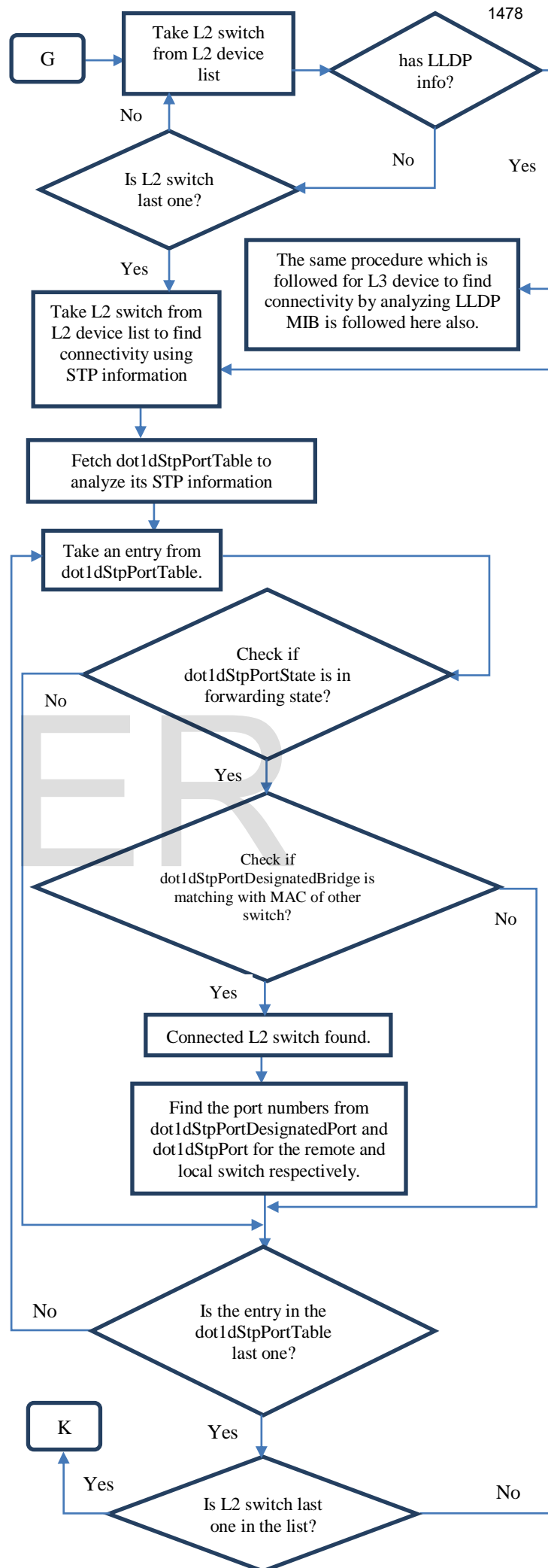
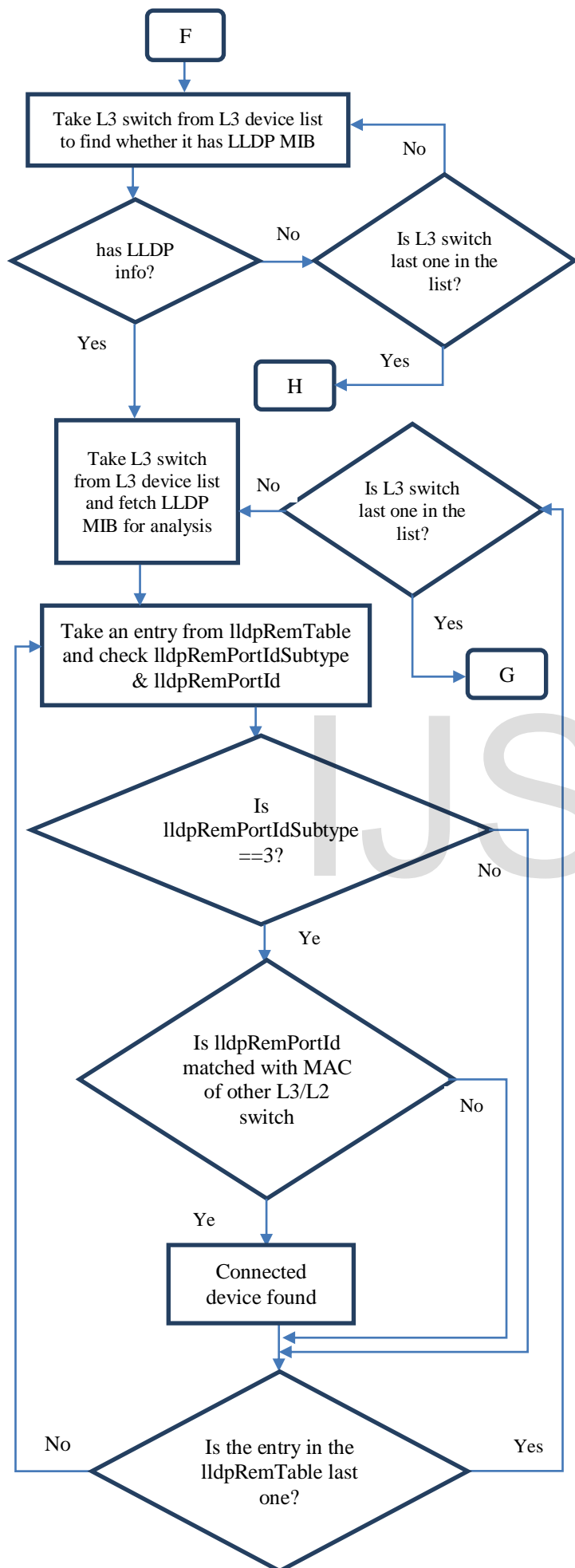
Analysis of AFT information:

AFT information is contained in the MIB object dot1dTpFdbTable. If the value of dot1dTpFdbStatus of dot1dTpFdbTable is 3 (i.e. learned) then dot1dTpFdbAddress of the table contains MAC addresses of connected (directly/indirectly) devices but dot1dTpFdbPort gives unique entry for the devices which are connected directly whereas dot1dTpFdbPort gives multiple entries for the devices which are not connected directly. Column dot1dTpFdbPort of dot1dTpFdbTable gives ifIndex of the local port at which remote device is connected. Thus physical topology is derived by finding unique entries in the dot1dTpFdbTable.

After discovering L2-L2, L2-L3/L3-L2 and L3-L3 connectivity, finally host connectivity is discovered by analysing AFT information of L2/L3 switches. Flow chart of the proposed algorithm is given in the following figure:



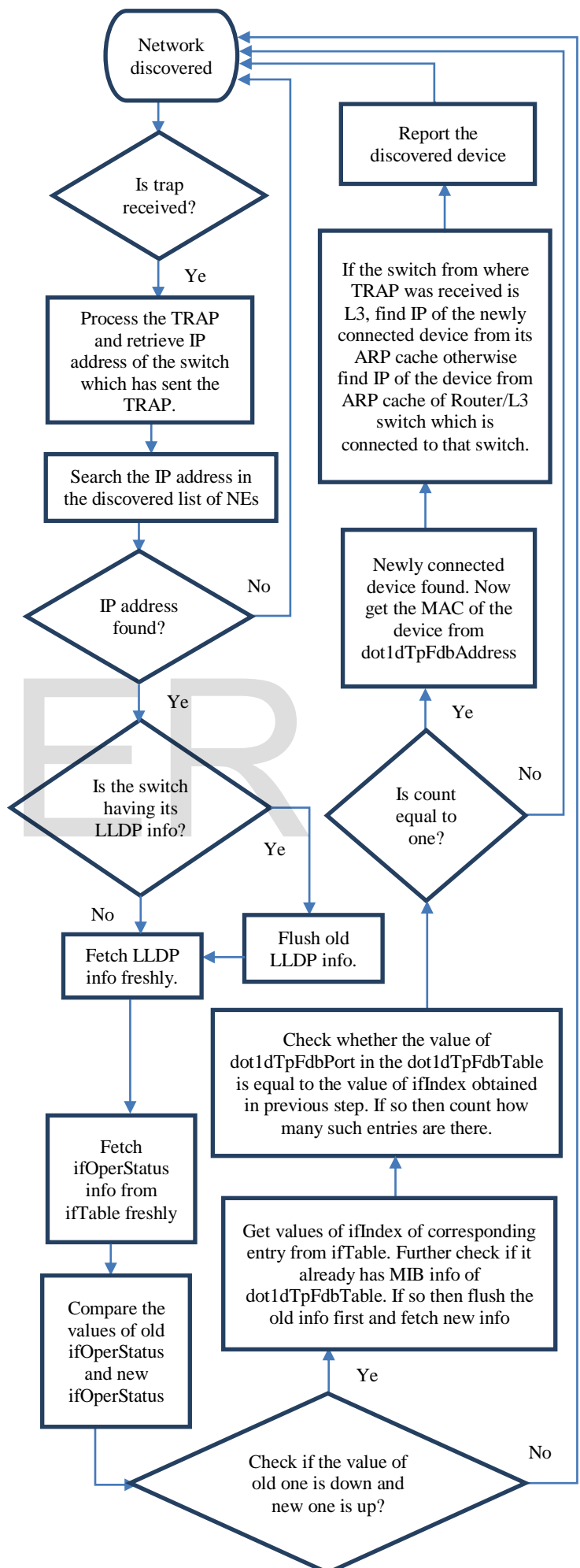




B. Approach for dynamic discovery of newly connected NE:

This approach works on the principle of SNMP trap which is received in case of any change in the network.

After a whole network is discovered if any new network element is plugged in to the network, the SNMP link up trap is generated from the switch to which the new element is plugged in. This trap is provided as an input to Dynamic Discovery algorithm which then extracts the IP address of the switch from which trap was generated. It then searches the IP address in the list of already discovered L2/L3 switch and eventually finds the candidate switch. Then it checks whether the switch's LLDP information is available or not. If the switch's LLDP information is already there in its data set, then it flushes out that old information and replaces it with new one else it will try to collect switch's LLDP information. After collecting LLDP information of the switch, connectivity of the switch with all other L2/L3-switch/Router is fetched. MIB object ifOperStatus of ifTable of the switch is further collected and compared with the old values of ifOperStatus. If the new value is "UP" and old value is "DOWN" then it retrieves the value of ifIndex of corresponding entry in the ifTable. It further checks whether the MIB information of dot1dTpFdbTable is available in its dataset or not. If the information is already available, then it flushes out that old information and replaces with the new one else it tries to collect the fresh information. The value of the dot1dTpFdbPort in dot1dTpFdbTable is compared with the value of ifIndex retrieved in earlier step. If the values are same then it counts for the number of such instances. If the count is one, then it decides that element discovered is new one and it retrieves the MAC of the newly discovered element. Now it tries to find out the IP of the newly discovered element and details of its connectivity with the switch. To find the IP of the new element it checks whether the switch from where trap was received, is L2 or L3. If it is L3 then it fetches the IP of newly discovered element from its ARP cache otherwise it fetches the IP of newly discovered element from the ARP cache of immediate router or L3 device. The algorithm to find connectivity of newly connected element is explained in following Figure:



III. EXPERIMENT AND RESULT

Our proposed algorithm was deployed and tested in an Ethernet network consisting of L3 switches and different host depicted in following diagram

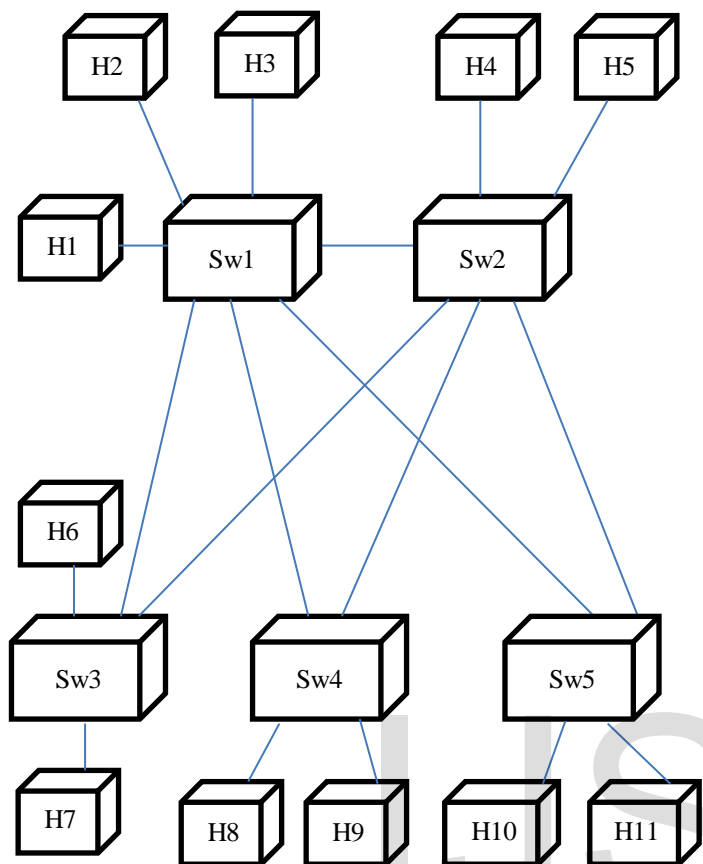


Fig1: Network Diagram 1

LLDP MIB object lldpRemTable of switch Sw1 as shown in Fig2 indicates that a device whose MAC is E8-05-6D-E7-68-17 is connected to it. Now the MIB object lldpLocPortTable of all the switches except Sw1 is fetched one by one to see whose MAC is E8-05-6D-E7-68-17. Once the candidate switch, which is Sw2 in this case, is found, then analysis of MIB object lldpLocPortId in lldpLocPortTable of it as shown in Fig3 indicates the port number 23 through which it is connected to Sw1.

Now analysis of MIB object lldpRemTable of switch Sw2 as shown in Fig4 and lldpLocPortTable of switch Sw1 as shown in Fig5 indicates that port number 23 of Sw1 is connected to Sw2. Similarly, further analysis of MIB object lldpRemTable of switch Sw1 and lldpLocPortTable of switch Sw3 as shown in Fig6 shows that the port number 22 of Sw1 is connected to port number 23 of switch Sw3. In the same way it is found out that the port number 20 and 21 of Sw1 is connected to port number 25 of switch Sw4 and port number 26 of switch Sw5 respectively.

Following the same procedure, it is found out that the port number 20, 21 and 22 of Sw2 is connected to port number 22 of switch Sw3, port number 24 of switch Sw4 and port number 25 of switch Sw5 respectively.

Name/OID	Value
lldpRemChassisIdSubtype.6499.13.4	macAddress (4)
lldpRemChassisIdSubtype.6518.1.3	macAddress (4)
lldpRemChassisIdSubtype.6574.25.1	macAddress (4)
lldpRemChassisIdSubtype.6575.26.2	macAddress (4)
lldpRemChassisId.6499.13.4	E8-05-6D-E7-68-00
lldpRemChassisId.6518.1.3	E8-05-6D-E7-84-00
lldpRemChassisId.6574.25.1	E8-05-6D-E7-81-00
lldpRemChassisId.6575.26.2	E8-05-6D-E7-80-00
lldpRemPortIdSubtype.6499.13.4	macAddress (3)
lldpRemPortIdSubtype.6518.1.3	macAddress (3)
lldpRemPortIdSubtype.6574.25.1	macAddress (3)
lldpRemPortIdSubtype.6575.26.2	macAddress (3)
lldpRemPortId.6499.13.4	E8-05-6D-E7-68-17
lldpRemPortId.6518.1.3	E8-05-6D-E7-84-17
lldpRemPortId.6574.25.1	E8-05-6D-E7-81-19
lldpRemPortId.6575.26.2	E8-05-6D-E7-80-1A
lldpRemPortDesc.6499.13.4	Port 23
lldpRemPortDesc.6518.1.3	Port 23
lldpRemPortDesc.6574.25.1	Port 25
lldpRemPortDesc.6575.26.2	Port 26

Fig2: lldpRemTable of Sw1

Name/OID	Value
lldpLocPortIdSubtype.25	macAddress (3)
lldpLocPortIdSubtype.26	macAddress (3)
lldpLocPortIdSubtype.27	macAddress (3)
lldpLocPortIdSubtype.28	macAddress (3)
lldpLocPortIdSubtype.29	macAddress (3)
lldpLocPortIdSubtype.30	macAddress (3)
lldpLocPortIdSubtype.31	macAddress (3)
lldpLocPortIdSubtype.32	macAddress (3)
lldpLocPortId.1	E8-05-6D-E7-68-01
lldpLocPortId.2	E8-05-6D-E7-68-02
lldpLocPortId.3	E8-05-6D-E7-68-03
lldpLocPortId.4	E8-05-6D-E7-68-04
lldpLocPortId.5	E8-05-6D-E7-68-05
lldpLocPortId.6	E8-05-6D-E7-68-06
lldpLocPortId.7	E8-05-6D-E7-68-07
lldpLocPortId.8	E8-05-6D-E7-68-08
lldpLocPortId.9	E8-05-6D-E7-68-09
lldpLocPortId.10	E8-05-6D-E7-68-0A
lldpLocPortId.11	E8-05-6D-E7-68-0B
lldpLocPortId.12	E8-05-6D-E7-68-0C
lldpLocPortId.13	E8-05-6D-E7-68-0D
lldpLocPortId.14	E8-05-6D-E7-68-0E
lldpLocPortId.15	E8-05-6D-E7-68-0F
lldpLocPortId.16	E8-05-6D-E7-68-10
lldpLocPortId.17	E8-05-6D-E7-68-11
lldpLocPortId.18	E8-05-6D-E7-68-12
lldpLocPortId.19	E8-05-6D-E7-68-13
lldpLocPortId.20	E8-05-6D-E7-68-14
lldpLocPortId.21	E8-05-6D-E7-68-15
lldpLocPortId.22	E8-05-6D-E7-68-16
lldpLocPortId.23	E8-05-6D-E7-68-17
lldpLocPortId.24	E8-05-6D-E7-68-18

Fig3: lldpLocPortTable of Sw2

Name/OID	Value
IldpRemChassisIdSubtype.6498.13.4	macAddress (4)
IldpRemChassisIdSubtype.6518.1.3	macAddress (4)
IldpRemChassisIdSubtype.6574.25.1	macAddress (4)
IldpRemChassisIdSubtype.6575.26.2	macAddress (4)
IldpRemChassisId.6498.13.4	E8-05-6D-E7-6C-00
IldpRemChassisId.6518.1.3	E8-05-6D-E7-84-00
IldpRemChassisId.6574.25.1	E8-05-6D-E7-81-00
IldpRemChassisId.6575.26.2	E8-05-6D-E7-80-00
IldpRemPortIdSubtype.6498.13.4	macAddress (3)
IldpRemPortIdSubtype.6518.1.3	macAddress (3)
IldpRemPortIdSubtype.6574.25.1	macAddress (3)
IldpRemPortIdSubtype.6575.26.2	macAddress (3)
IldpRemPortId.6498.13.4	E8-05-6D-E7-6C-17
IldpRemPortId.6518.1.3	E8-05-6D-E7-84-16
IldpRemPortId.6574.25.1	E8-05-6D-E7-81-18
IldpRemPortId.6575.26.2	E8-05-6D-E7-80-19
IldpRemPortDesc.6498.13.4	Port 23
IldpRemPortDesc.6518.1.3	Port 22
IldpRemPortDesc.6574.25.1	Port 24
IldpRemPortDesc.6575.26.2	Port 25

Fig4: IldpRemTable of Sw2

Name/OID	Value
IldpLocPortIdSubtype.29	macAddress (3)
IldpLocPortIdSubtype.30	macAddress (3)
IldpLocPortIdSubtype.31	macAddress (3)
IldpLocPortIdSubtype.32	macAddress (3)
IldpLocPortId.1	E8-05-6D-E7-6C-01
IldpLocPortId.2	E8-05-6D-E7-6C-02
IldpLocPortId.3	E8-05-6D-E7-6C-03
IldpLocPortId.4	E8-05-6D-E7-6C-04
IldpLocPortId.5	E8-05-6D-E7-6C-05
IldpLocPortId.6	E8-05-6D-E7-6C-06
IldpLocPortId.7	E8-05-6D-E7-6C-07
IldpLocPortId.8	E8-05-6D-E7-6C-08
IldpLocPortId.9	E8-05-6D-E7-6C-09
IldpLocPortId.10	E8-05-6D-E7-6C-0A
IldpLocPortId.11	E8-05-6D-E7-6C-0B
IldpLocPortId.12	E8-05-6D-E7-6C-0C
IldpLocPortId.13	E8-05-6D-E7-6C-0D
IldpLocPortId.14	E8-05-6D-E7-6C-0E
IldpLocPortId.15	E8-05-6D-E7-6C-0F
IldpLocPortId.16	E8-05-6D-E7-6C-10
IldpLocPortId.17	E8-05-6D-E7-6C-11
IldpLocPortId.18	E8-05-6D-E7-6C-12
IldpLocPortId.19	E8-05-6D-E7-6C-13
IldpLocPortId.20	E8-05-6D-E7-6C-14
IldpLocPortId.21	E8-05-6D-E7-6C-15
IldpLocPortId.22	E8-05-6D-E7-6C-16
IldpLocPortId.23	E8-05-6D-E7-6C-17
IldpLocPortId.24	E8-05-6D-E7-6C-18
IldpLocPortId.25	E8-05-6D-E7-6C-19

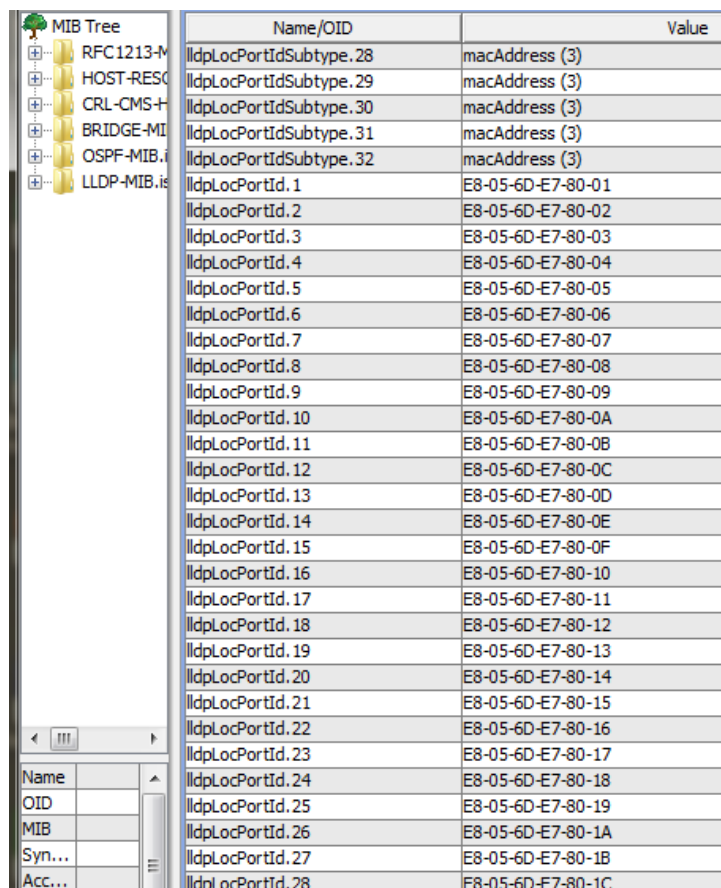
Fig5: IldpLocPortTable of Sw1

Name/OID	Value
IldpLocPortIdSubtype.25	macAddress (3)
IldpLocPortIdSubtype.26	macAddress (3)
IldpLocPortIdSubtype.27	macAddress (3)
IldpLocPortIdSubtype.28	macAddress (3)
IldpLocPortIdSubtype.29	macAddress (3)
IldpLocPortIdSubtype.30	macAddress (3)
IldpLocPortIdSubtype.31	macAddress (3)
IldpLocPortIdSubtype.32	macAddress (3)
IldpLocPortId.1	E8-05-6D-E7-84-01
IldpLocPortId.2	E8-05-6D-E7-84-02
IldpLocPortId.3	E8-05-6D-E7-84-03
IldpLocPortId.4	E8-05-6D-E7-84-04
IldpLocPortId.5	E8-05-6D-E7-84-05
IldpLocPortId.6	E8-05-6D-E7-84-06
IldpLocPortId.7	E8-05-6D-E7-84-07
IldpLocPortId.8	E8-05-6D-E7-84-08
IldpLocPortId.9	E8-05-6D-E7-84-09
IldpLocPortId.10	E8-05-6D-E7-84-0A
IldpLocPortId.11	E8-05-6D-E7-84-0B
IldpLocPortId.12	E8-05-6D-E7-84-0C
IldpLocPortId.13	E8-05-6D-E7-84-0D
IldpLocPortId.14	E8-05-6D-E7-84-0E
IldpLocPortId.15	E8-05-6D-E7-84-0F
IldpLocPortId.16	E8-05-6D-E7-84-10
IldpLocPortId.17	E8-05-6D-E7-84-11
IldpLocPortId.18	E8-05-6D-E7-84-12
IldpLocPortId.19	E8-05-6D-E7-84-13
IldpLocPortId.20	E8-05-6D-E7-84-14
IldpLocPortId.21	E8-05-6D-E7-84-15
IldpLocPortId.22	E8-05-6D-E7-84-16
IldpLocPortId.23	E8-05-6D-E7-84-17
IldpLocPortId.24	E8-05-6D-E7-84-18

Fig6: IldpLocPortTable of Sw3

Name/OID	Value
IldpLocPortIdSubt...	macAddress (3)
IldpLocPortIdSubt...	macAddress (3)
IldpLocPortIdSubt...	macAddress (3)
IldpLocPortIdSubt...	macAddress (3)
IldpLocPortIdSubt...	macAddress (3)
IldpLocPortId.1	E8-05-6D-E7-81-01
IldpLocPortId.2	E8-05-6D-E7-81-02
IldpLocPortId.3	E8-05-6D-E7-81-03
IldpLocPortId.4	E8-05-6D-E7-81-04
IldpLocPortId.5	E8-05-6D-E7-81-05
IldpLocPortId.6	E8-05-6D-E7-81-06
IldpLocPortId.7	E8-05-6D-E7-81-07
IldpLocPortId.8	E8-05-6D-E7-81-08
IldpLocPortId.9	E8-05-6D-E7-81-09
IldpLocPortId.10	E8-05-6D-E7-81-0A
IldpLocPortId.11	E8-05-6D-E7-81-0B
IldpLocPortId.12	E8-05-6D-E7-81-0C
IldpLocPortId.13	E8-05-6D-E7-81-0D
IldpLocPortId.14	E8-05-6D-E7-81-0E
IldpLocPortId.15	E8-05-6D-E7-81-0F
IldpLocPortId.16	E8-05-6D-E7-81-10
IldpLocPortId.17	E8-05-6D-E7-81-11
IldpLocPortId.18	E8-05-6D-E7-81-12
IldpLocPortId.19	E8-05-6D-E7-81-13
IldpLocPortId.20	E8-05-6D-E7-81-14
IldpLocPortId.21	E8-05-6D-E7-81-15
IldpLocPortId.22	E8-05-6D-E7-81-16
IldpLocPortId.23	E8-05-6D-E7-81-17
IldpLocPortId.24	E8-05-6D-E7-81-18
IldpLocPortId.25	E8-05-6D-E7-81-19

Fig7: IldpLocPortTable of Sw4

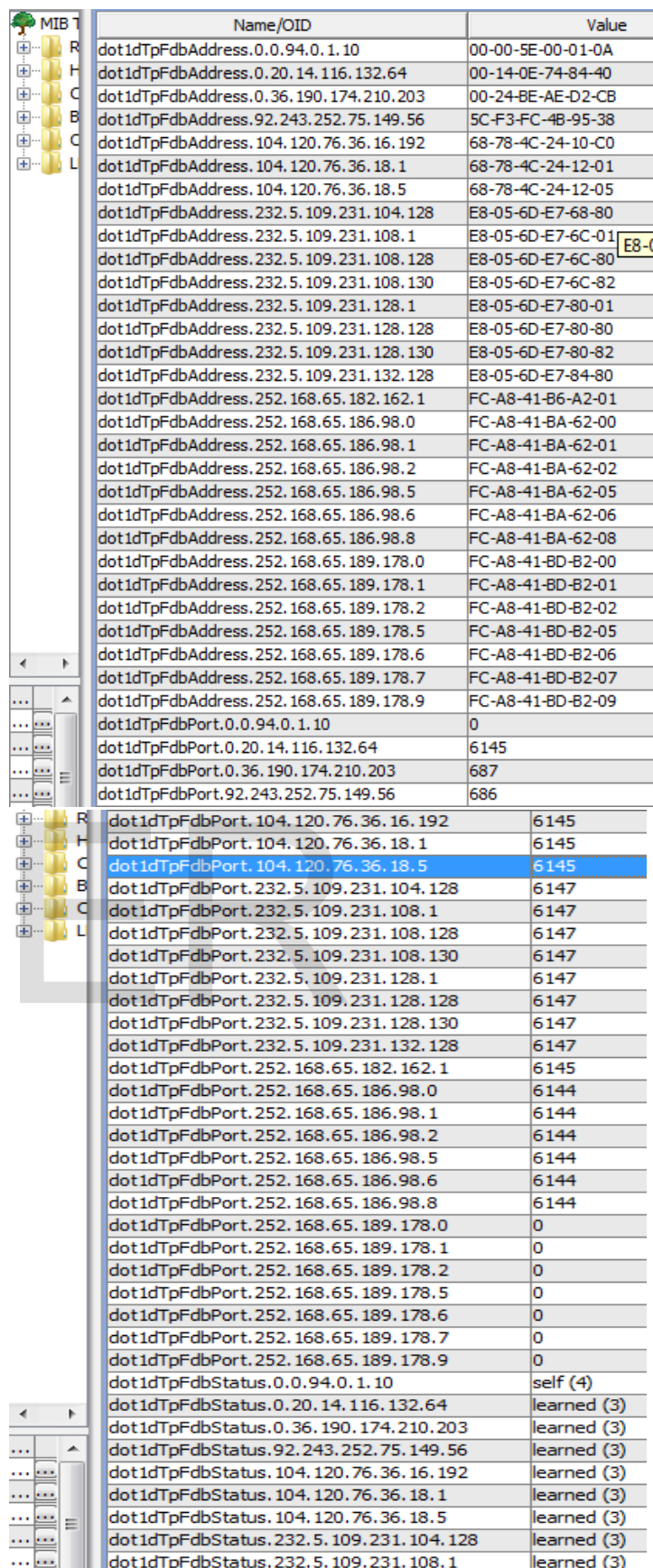


The image shows a MIB Tree on the left and a table of IldpLocPortTable of Sw5 on the right. The MIB Tree lists various protocols like RFC 1213-M, HOST-RES, CRL-CMS, BRIDGE-MIB, OSPF-MIB, and LLDP-MIB. The table lists IldpLocPortIdSubtype and IldpLocPortId values for ports 1 through 28.

Name/OID	Value
IldpLocPortIdSubtype.28	macAddress (3)
IldpLocPortIdSubtype.29	macAddress (3)
IldpLocPortIdSubtype.30	macAddress (3)
IldpLocPortIdSubtype.31	macAddress (3)
IldpLocPortIdSubtype.32	macAddress (3)
IldpLocPortId.1	E8-05-6D-E7-80-01
IldpLocPortId.2	E8-05-6D-E7-80-02
IldpLocPortId.3	E8-05-6D-E7-80-03
IldpLocPortId.4	E8-05-6D-E7-80-04
IldpLocPortId.5	E8-05-6D-E7-80-05
IldpLocPortId.6	E8-05-6D-E7-80-06
IldpLocPortId.7	E8-05-6D-E7-80-07
IldpLocPortId.8	E8-05-6D-E7-80-08
IldpLocPortId.9	E8-05-6D-E7-80-09
IldpLocPortId.10	E8-05-6D-E7-80-0A
IldpLocPortId.11	E8-05-6D-E7-80-0B
IldpLocPortId.12	E8-05-6D-E7-80-0C
IldpLocPortId.13	E8-05-6D-E7-80-0D
IldpLocPortId.14	E8-05-6D-E7-80-0E
IldpLocPortId.15	E8-05-6D-E7-80-0F
IldpLocPortId.16	E8-05-6D-E7-80-10
IldpLocPortId.17	E8-05-6D-E7-80-11
IldpLocPortId.18	E8-05-6D-E7-80-12
IldpLocPortId.19	E8-05-6D-E7-80-13
IldpLocPortId.20	E8-05-6D-E7-80-14
IldpLocPortId.21	E8-05-6D-E7-80-15
IldpLocPortId.22	E8-05-6D-E7-80-16
IldpLocPortId.23	E8-05-6D-E7-80-17
IldpLocPortId.24	E8-05-6D-E7-80-18
IldpLocPortId.25	E8-05-6D-E7-80-19
IldpLocPortId.26	E8-05-6D-E7-80-1A
IldpLocPortId.27	E8-05-6D-E7-80-1B
IldpLocPortId.28	E8-05-6D-E7-80-1C

Fig8: IldpLocPortTable of Sw5

Connectivity of host H4 and H5 with switch Sw2 is found by analyzing Sw2's FDB table which contains MAC address of different devices. MAC address of H4 and H5 is found by analyzing Sw2's ARP cache table represented by MIB object ipNetToMediaTable and MAC address of H4 & H5 is found to be 00-24-BE-AE-D2-CB and 5C-F3-FC-4B-95-38. FDB table of Sw2 as shown in Fig9a and Fig9b shows that MAC addresses of H4 and H5 are present against the port ifIndex number 687 and 686. Further analysis of the table reveals that there are no multiple entries for port ifIndex number 687 and 686 which concludes that H4 and H5 are connected to Sw2 and ifIndexes of the connected port of Sw2 are 687 and 686. Thus the proposed algorithm finds the topology of the network shown in Fig1.



The image shows a MIB Tree on the left and a table of dot1dTpFdbTable of Sw2 on the right. The MIB Tree lists various protocols like RFC 1213-M, HOST-RES, CRL-CMS, BRIDGE-MIB, OSPF-MIB, and LLDP-MIB. The table lists dot1dTpFdbAddress, dot1dTpFdbPort, and dot1dTpFdbStatus values for ports 1 through 28.

Name/OID	Value
dot1dTpFdbAddress.0.0.94.0.1.10	00-00-5E-00-01-0A
dot1dTpFdbAddress.0.20.14.116.132.64	00-14-0E-74-84-40
dot1dTpFdbAddress.0.36.190.174.210.203	00-24-BE-AE-D2-CB
dot1dTpFdbAddress.92.243.252.75.149.56	5C-F3-FC-4B-95-38
dot1dTpFdbAddress.104.120.76.36.16.192	68-78-4C-24-10-C0
dot1dTpFdbAddress.104.120.76.36.18.1	68-78-4C-24-12-01
dot1dTpFdbAddress.104.120.76.36.18.5	68-78-4C-24-12-05
dot1dTpFdbAddress.232.5.109.231.104.128	E8-05-6D-E7-80-80
dot1dTpFdbAddress.232.5.109.231.108.1	E8-05-6D-E7-80-01
dot1dTpFdbAddress.232.5.109.231.108.128	E8-05-6D-E7-80-80
dot1dTpFdbAddress.232.5.109.231.108.130	E8-05-6D-E7-80-82
dot1dTpFdbAddress.232.5.109.231.128.1	E8-05-6D-E7-80-01
dot1dTpFdbAddress.232.5.109.231.128.128	E8-05-6D-E7-80-80
dot1dTpFdbAddress.232.5.109.231.128.130	E8-05-6D-E7-80-82
dot1dTpFdbAddress.232.5.109.231.132.128	E8-05-6D-E7-84-80
dot1dTpFdbAddress.252.168.65.182.162.1	FC-A8-41-B6-A2-01
dot1dTpFdbAddress.252.168.65.186.98.0	FC-A8-41-BA-62-00
dot1dTpFdbAddress.252.168.65.186.98.1	FC-A8-41-BA-62-01
dot1dTpFdbAddress.252.168.65.186.98.2	FC-A8-41-BA-62-02
dot1dTpFdbAddress.252.168.65.186.98.5	FC-A8-41-BA-62-05
dot1dTpFdbAddress.252.168.65.186.98.6	FC-A8-41-BA-62-06
dot1dTpFdbAddress.252.168.65.186.98.8	FC-A8-41-BA-62-08
dot1dTpFdbAddress.252.168.65.189.178.0	FC-A8-41-BD-B2-00
dot1dTpFdbAddress.252.168.65.189.178.1	FC-A8-41-BD-B2-01
dot1dTpFdbAddress.252.168.65.189.178.2	FC-A8-41-BD-B2-02
dot1dTpFdbAddress.252.168.65.189.178.5	FC-A8-41-BD-B2-05
dot1dTpFdbAddress.252.168.65.189.178.6	FC-A8-41-BD-B2-06
dot1dTpFdbAddress.252.168.65.189.178.7	FC-A8-41-BD-B2-07
dot1dTpFdbAddress.252.168.65.189.178.9	FC-A8-41-BD-B2-09
dot1dTpFdbPort.0.0.94.0.1.10	0
dot1dTpFdbPort.0.20.14.116.132.64	6145
dot1dTpFdbPort.0.36.190.174.210.203	687
dot1dTpFdbPort.92.243.252.75.149.56	686
dot1dTpFdbPort.104.120.76.36.16.192	6145
dot1dTpFdbPort.104.120.76.36.18.1	6145
dot1dTpFdbPort.104.120.76.36.18.5	6145
dot1dTpFdbPort.232.5.109.231.104.128	6147
dot1dTpFdbPort.232.5.109.231.108.1	6147
dot1dTpFdbPort.232.5.109.231.108.128	6147
dot1dTpFdbPort.232.5.109.231.108.130	6147
dot1dTpFdbPort.232.5.109.231.128.1	6147
dot1dTpFdbPort.232.5.109.231.128.128	6147
dot1dTpFdbPort.232.5.109.231.128.130	6147
dot1dTpFdbPort.232.5.109.231.132.128	6147
dot1dTpFdbPort.252.168.65.182.162.1	6145
dot1dTpFdbPort.252.168.65.186.98.0	6144
dot1dTpFdbPort.252.168.65.186.98.1	6144
dot1dTpFdbPort.252.168.65.186.98.2	6144
dot1dTpFdbPort.252.168.65.186.98.5	6144
dot1dTpFdbPort.252.168.65.186.98.6	6144
dot1dTpFdbPort.252.168.65.186.98.8	6144
dot1dTpFdbPort.252.168.65.189.178.0	0
dot1dTpFdbPort.252.168.65.189.178.1	0
dot1dTpFdbPort.252.168.65.189.178.2	0
dot1dTpFdbPort.252.168.65.189.178.5	0
dot1dTpFdbPort.252.168.65.189.178.6	0
dot1dTpFdbPort.252.168.65.189.178.7	0
dot1dTpFdbPort.252.168.65.189.178.9	0
dot1dTpFdbStatus.0.0.94.0.1.10	self (4)
dot1dTpFdbStatus.0.20.14.116.132.64	learned (3)
dot1dTpFdbStatus.0.36.190.174.210.203	learned (3)
dot1dTpFdbStatus.92.243.252.75.149.56	learned (3)
dot1dTpFdbStatus.104.120.76.36.16.192	learned (3)
dot1dTpFdbStatus.104.120.76.36.18.1	learned (3)
dot1dTpFdbStatus.104.120.76.36.18.5	learned (3)
dot1dTpFdbStatus.232.5.109.231.104.128	learned (3)
dot1dTpFdbStatus.232.5.109.231.108.1	learned (3)

Fig9a: dot1dTpFdbTable of Sw2

R	dot1dTpFdbStatus.232.5.109.231.108.128	learned (3)
H	dot1dTpFdbStatus.232.5.109.231.108.130	learned (3)
B	dot1dTpFdbStatus.232.5.109.231.128.1	learned (3)
C	dot1dTpFdbStatus.232.5.109.231.128.128	learned (3)
C	dot1dTpFdbStatus.232.5.109.231.128.130	learned (3)
L	dot1dTpFdbStatus.232.5.109.231.132.128	learned (3)
	dot1dTpFdbStatus.252.168.65.182.162.1	learned (3)
	dot1dTpFdbStatus.252.168.65.186.98.0	learned (3)
	dot1dTpFdbStatus.252.168.65.186.98.1	learned (3)
	dot1dTpFdbStatus.252.168.65.186.98.2	learned (3)
	dot1dTpFdbStatus.252.168.65.186.98.5	learned (3)
	dot1dTpFdbStatus.252.168.65.186.98.6	learned (3)
	dot1dTpFdbStatus.252.168.65.186.98.8	learned (3)
	dot1dTpFdbStatus.252.168.65.189.178.0	self (4)
	dot1dTpFdbStatus.252.168.65.189.178.1	self (4)
	dot1dTpFdbStatus.252.168.65.189.178.2	self (4)
	dot1dTpFdbStatus.252.168.65.189.178.5	self (4)
	dot1dTpFdbStatus.252.168.65.189.178.6	self (4)
	dot1dTpFdbStatus.252.168.65.189.178.7	self (4)
	dot1dTpFdbStatus.252.168.65.189.178.9	self (4)

Fig9b: dot1dTpFdbTable of Sw2

IV. CONCLUSION

This paper describes an approach for finding the layer-2 topology of a network combining LLDP technique along with other technique described in [1], [2], [3] & [4]. The proposed approach first checks what MIB information is available and depending upon the availability of MIB information, it chooses a particular path of the algorithm and applies the information data to find the layer 2 path among network elements.

This paper also describes the approach to discover new network element with its connectivity with switch to which it is just plugged into the already discovered network. Correctness of the results of the algorithm is verified by checking the physical connection among network elements in different networks where the proposed algorithm was tested successfully.

Depending upon the network size and entries in router's route table, it takes several minutes to several hours to give output of physical topology.

V. REFERENCES

- [1] BBC Research White Paper WHP 188 2010.
<https://www.bbc.co.uk/rd/publications/whitepaper188>
- [2] Layer-2 Path Discovery Using Spanning Tree MIBs David T. Stott, Avaya Labs Research, Avaya Inc. 233 Mount Airy Road Basking Ridge, NJ 07920
https://www.researchgate.net/publication/2476855_Layer-2_Path_Discovery_Using_Spanning_Tree_MIBs
- [3] IP Network Topology Discovery Using SNMP
Suman Pandey#1, Mi-Jung Choi#2, Sung-Joo Lee#3, James W. Hong#4
Dept. of Computer Science and Engineering, POSTECH, Korea.
¹suman@postech.ac.kr, ²mjchoi@postech.ac.kr,
³forstar@postech.ac.kr, ⁴jwkhong@postech.ac.kr

https://www.researchgate.net/publication/224437374_IP_net_work_topology_discovery_using_SNMP

- [4] Efficient Physical Topology Discovery for Large OSPF Networks
Choonho Son, Junsuk Oh, Kyoung-Ho Lee, Kieung Kim, Jaehyung Yoo
Network Technology Laboratory
Korea Telecom (KT)
Daejeon, Korea 305-811
Email: {choonho,jsok,caza,gekim,styoo}@kt.co.kr
<https://www.ieeexplore.ieee.org/document/4575151>