Sep. 30

International Journal of advanced studies in Computer Science and Engineering
IJASCSE, Volume 3, Issue 9, 2014

Improve Load Balancing In Provider Backbone Bridge Network Using 64 Shortest Path Trees

Pham Thi Quynh Anh
Department of Computer Science
Hue Industrial College
Hue, Vietnam

***Abstract*— Load balancing is a critical demand in Provider Backbone Bridge Network (PBBN). Equal Cost Multiple Path (ECMP) in Shortest Path Bridging (SPB), described in the IEEE 802.1aq standard, is known as a scheme to gain better load balancing comparing to other schemes by enabling bridge to use multiple equal shortest paths between it and other bridges. In ECMP, up to 16 shortest paths are available for traffic forwarding. However, in a big network, in which there are more than 16 shortest paths between some two bridges, ECMP becomes limited. Therefore, we proposed an improved method applying ECMP in a big network.**

Keywords-component; SPB; ECMP; load balancing;

I. INTRODUCTION

A PBBN [8] connects separated metro networks and other backbone networks. A PBBN consists two kinds of bridges: Backbone Core Bridges (BCBs) and Backbone Edge Bridges (BEBs). Each BEB connects a Provider Network to the backbone network. BCBs are located in the core of network forwarding the frames in a large scale. In a PBBN only BEBs are transferring (source) and receiving (destination) nodes. In this paper, *nodes* and *bridges* are interchangeably used.

Shortest Path Bridging (SPB), described in the IEEE 802.1aq standard, is a computer networking technology simplifying the creation and configuration of networks, enabling multipath bridging [2].

SPB is the replacement of the Spanning Tree Protocol (STP) which can prevent loops in bridging but does not ensure that the path between any two nodes is the shortest path. In fact, in STP, only paths from the root of the Spanning Tree (ST) to other nodes are ensured to be the shortest ones. In the other hand, to SPB, forwarding path between any two nodes is always guaranteed to be the shortest one.

SPB has two modes: the first is SPB for MACs (SPBM) for PBBN; the second one is SPB for VIDs (SPBV) for Q in Q of VLAN bridges [2]. In this paper, we consider the first modes of SPB.

SPB supports unicast, multicast and broadcast routing; all routing is on symmetric shortest paths.

The routing control plane of SPB in PBBN is based on Intermediate System to Intermediate System (IS-IS) protocol [5]. Therefore, in some documents SPB is referred as SPB-ISIS. IS-IS protocol enables every bridge in the network to learn the network topology. Each node then independently computes shortest paths from it to other possible destinations using Dijktra's algorithm; these shortest paths contribute a Shortest Path Tree (SPT) of that node.

Moreover, SPB supports Equal-Cost-Multiple-Path bridging (ECMP) [1] to achieve the load balancing for the network. ECMP allows multiple shortest paths between any two nodes to be active, provides faster convergence times, improving the use of the mesh topologies through increased bandwidth and redundancy between all devices [2].

II. 16-SPT METHOD

Currently, ECMP enables each BEB in PBBN to generate 16 SPTs to use in forwarding frames to other BEBs in the network. This method results almost every links in the network to be available, thus increases the using efficiency to the resources of the network.

Whereby, if there are two equal cost shortest paths (ECSPs) between two nodes, ECMP performs the tie-breaking algorithm as following: first, if one path has smaller number of hops than the other, the smaller path is chosen; otherwise, the path which has minimum BridgeID is chosen [1]. BridgeID of a node is contributed from BridgePriority (2 bytes) and either IS-IS SYSID (6 bytes) in the case that nodes are in different IS-IS level or MAC address (6 bytes) of node in case of same IS-IS level. In the paper, for simplification, we assume that equal cost paths are ones which have same number of hops; thus we use the second situation of tie-breaking algorithm only.

ECMP enables every BEB in the network to generate 16 SPTs with following steps:

Step 1:

Let E is the set of BEBs in a PBBN. To each bridge A belongs to E and every B in E, B is different from A, find every shortest path from A to each bridge B.

Let P(A,B) is the set of all shortest paths between node A and node B. If there are more than node path in P(A,B), then the tie-breaking method is used to choose the forwarding path between A and B from P(A,B).

Let PathIDp to be an unique identifier for a path in the network, made of BridgeIDs in path p, sorting in an increasing order; p $\in$ P(A,B).

The tire-breaking rule is that if PathIDx (x $\in$ P(A,B)) less than every PathIDi (i $\in$ P(A,B)) then x is the forwarding path from A to B.

For example, suppose that there are three shortest paths between A and B. The PathIDs of these three paths are: ‘ABDHU’, ‘ABEGH’ and ‘ABDUV’. For the reason that ‘ABDHU’ is smallest one among three, the path with PathID ‘ABDHU’ is chosen as the forwarding path.

All forwarding paths rooted from node A contribute a SPT of node A.

After processing every node A belonging to E, every BEB in the network has its own SPT which is used to forward data from BEB to BEB in the network.

Step 2:

Sixteen SPTs of each BEB in the network are created by applying the step 1 after XOR-ing BridgeIDs of all nodes in the network with 16 different bit-masks: 0x00, 0x11,…0xFF.

With sixteen ways converting BridgeIDs, each BEB has up to 16 SPTs. ECMP enables many different shortest paths to be available between two nodes instead of using only one path as in STP. This method results much better load balancing in the network comparing to STP.

However, 16 ECTs of a node maybe overlap to each other. Hence, if there are 16 ECSPs between two nodes, these paths are not assured to be used all. Some of them maybe repeatedly used some times why the others are not used.

Moreover, in the case that a network has more than 16 ECSPs, 16-SPT method limits the capability of links in the network by enabling up to only 16 shortest paths to be available.

III. 64-SPT METHOD

Instead of create 16 SPTs, each BEB in network generates 64 SPTs for itself to forward the frames to other nodes. This method enables a BEB to use up to 64 ECSPs between itself to other BEBs in the network.

In the new idea, Step 1 is same as the 16-SPT method’s. Meanwhile, in the Step 2, instead of XOR-ing the BridgeID of each node with 16 bit-masks in turns, the algorithm XORs BridgeID of each node 64 bit-masks - 0x00, 0x01, 0x02,…, 0x0F, 0x10, 0x11,…, 0x3F - to convert to 64 different BridgeIDs. Since there are 64 different cases of converting BridgeIDs, each node can generates up to 64 SPTs for itself.

Thus, it is clearly can be seen that 64-SPT method enhances a better load balancing comparing to the 16-SPT method of ECMP.

The advantage of the proposed method is to improve the using capability of redundant links in a big PBBN which usually has many ECSPs between any two nodes.

In the case ECSPs are completely separated to each other, the improved method is 4 times better than the current method. However, in reality, for the reason that the efficiency of current method and the improved one depends on the BridgeIDs of nodes in the network and the ECSPs in network are overlap in a large number of links, the enhancement of new method from 16-SPT method varies in different cases.

IV. ANALYZE

A. The testing network

We have tested our improved method in the PBBN showed in Figure1 which has about 16-20 ECSPs between some couples of BEBs.

In Network 1, the BCBs are nodes marked with numbers 3, 4…, 16, 17, 21, 22, 23, 24; and the rest are BEBs.

We call link which connect a BEB to a BCB *edge link*; call link connecting BCBs to each other *core link*. We call bridges 21, 22, 23, 24 *level-1 core bridges*; links which connect 4 *level-1 core bridges* to other each are *level-1 core link*; other core links are *level-2 core links*.

We concern much about the *level-1 core links* because in the network a large amount of traffic goes through these links. The greater number of BEBs is, the heavier the traffic loading of core links is. With this reason, keeping these links balancing plays a key role in keeping balancing of whole network.

B. Measument and Comparison

To evaluate the affectivity of the two methods on load balancing of the network, we calculate the number of times each links in the network is used (T) when applying each method in a certain set of BridgeIDs. T(*l*) means the times link (*l*) is used when we applying 16-SPT method (or 64-SPT method) in a case of BridgeIDs distribution.

When a forwarding path goes through a link *l*, T(*l*) is added by 1.

$$T(l) = T(l) + 1 \qquad (1)$$

Assume that there are 4 equal cost paths from node E to node G: E-A-B-G, E-A-C-G, E-D-B-G, E-D-C-G. If after tie-breaking, the path E-A-B-G (which has PathID ‘AEBG’) is chosen as a forwarding path from E to G, if call *l1* = (A,E), *l2* = (A,B), *l3* = (B,G), then T(*l1*) = T(*l1*) + 1; T(*l2*) = T(*l2*) + 1; T(*l3*) = T(*l3*) + 1.

Applying 16-SPT method, we get of every links in the network; applying improved method, we get T_i of every links in the network. Then we calculate the Standard

Deviation (σ) of $\bar{}$ and T_i of all links in the network using the fomular (2) and (3).

$$\sigma_c = \sqrt{\frac{\sum_{j=1}^{N}(T_c(l_j)-\mu)^2}{N}} \quad (2)$$

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{N}(T_i(l_j)-\mu)^2}{N}} \quad (3)$$

in which:

N is the number of links in the network;

μ is the average value T of all links in the network.

$$\mu = \frac{\sum_{j=1}^{N} T_c(l_j)}{N} = \frac{\sum_{j=1}^{N} T_i(l_j)}{N} \quad (4)$$

Then, compute the Coefficient of Variation (CV) of the used times of all links in each group when the two methods are applied:

$$CV_c = \frac{\sigma_c}{\mu} \quad (5)$$

$$CV_i = \frac{\sigma_i}{\mu} \quad (6)$$

The less value of CV is, the better load balancing the network gains.

We have tested our proposed method with 16*C*4 sets of BridgesIDs. In which, the BridgeIDs of 4 *level-1 core bridges* is changed with every combination of 16 numbers (from set S = {0, 1,…,15}) choosing 4; while other BridgeIDs is not changed. Let *M* = 16*C*4.

To each BridgeID distributions, apply 16-SPT and 64-SPT method to calculate CV_c and CV_i. Hence, totally, we have *M* CV_c and *M* CV_i. Denote CV_{c_k} the CV_c in k^th case of BridgeID distribution; CV_{i_k} the CV_i in k^th case of BridgeID distribution.

We compute the enhancement (*e*) of 64-SPT method from 16_SPT method in every case k^th with formula (7)

$$e_k = \frac{(CV_{c_k} - CV_{i_k}) * 100\%}{CV_{c_k}} \quad (7)$$

Then we calculate the average enhancement (μ_e) of *M* cases:

$$\mu_e = \frac{\sum_{k=1}^{M} e_k}{M} \quad (8)$$

We also consider about the minimum and maximum of e_k.

$$max_e = \max(e_k);\ k = 1,2,\ldots, M \quad (9)$$

$$min_e = \min(e_k);\ k = 1,2,\ldots, M \quad (10)$$

Since *level-1 core links* and *core links* are most sensitive to the load balancing in the network, we consider much about the enhancement of new method in these link groups.

The result of our computation is showed in the Table1.

TABLE I. THE ENHANCEMENT OF 64-SPT METHOD FROM 16-SPT METHOD

Enhancement	All links (%)	Core links (%)	Level-1 core links (%)
Average	3.0	4.6	14.0
Min	2.2	3.4	-6.0
Max	4.4	6.9	50.2

From Table1, we can see that the new method enhances a better load balancing in all links comparing to 16-SPT, from 2.2% to 4.4%. Especially, the new method has a good effect on *core links* load balancing, from -6% up to 50.2%.

In the case that the ECSPs are distinct from each other, the advantage of new method becomes much clearer. That means the new method gains a higher load balancing enhancement from the 16-SPT method.

It is clear that 64-ECT method is also better 16-ECT method in case that there are more than 16 distinct equal paths between two nodes in the network. For the reason 16-ECT method can use not more than 16 paths while 64-ECT method enables a node to use more than 16 equal paths.

V. CONCLUSION

64-SPT method enhances load balance from 2% - 4% in whole network, from 3% - 7 % in *core link* group, from -6% - 50% in *level-1 core links* in the testing network. It is easy to see that the more distinct equal cost paths are, the better load balance will be gained when 64-SPT method applied; and the greater number of equal cost paths is, the better load balance of links is when improved method applied.

REFERENCES

[1] 802.1Qbp-2014 - IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks - Amendment 22: Equal Cost Multiple Path (ECMP), 978-0-7381-9001-3, pp. 1 – 127, April 7 2014

[2] IEEE Computer Society, "IEEE Standard for Local and metropolitan area networks−Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks− Amendment 20: Shortest Path Bridging", 3 Park Avenue New York, NY 10016-5997 USA 29 June 2012

[3] IEEE Std. 802.1aq D3.6, "IEEE Draft Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks – Amendment 9: Shortest Path Bridging", February 2011.

[4] Chiabaut, J., Allan, D.,Bragg, N., Ashwood Smith, "P. Tie-breaking in shortest path determination." U.S. Patent 7,911,944 B2, Mar. 22, 2011

[5] David Allan, Peter Ashwood-Smith, Nigel Bragg, Don Fedyk,…, “Shortest Path Bridging: Efficient Control of Larger Ethernet Networks”, IEEE Communication Magazine, 0163-6804/10, pp. 128-135, Oct.2010

[6] David Allan, Janos Farkas, Scott Mansfield, “Intelligent load balancing for Shortest Path Bridging”, Communications Magazine, IEEE Vol. 50, pp. 163-167, July 2012

[7] David Allan, Peter Ashwood-Smith, Nigel Bragg, and Don Fedyk, Nortel, “Provider Link State Bridging”, IEEE Communication Magazine, 0163-6804/08, pp. 110-117, Sept. 2008

[8] Bob Sultan, Ben Mack-Crane, “PBBN in the Data Center”, IEEE 802.1 Interim Meeting Pittsburgh, PA., USA, May 2009

[9] [9] Ming Tian, Ju-long Lan, Xuan-young Zhu, Jian-mei Huang, “A routing Optimization of Equal-Cost-Multi-Paths based on link Criticality”, IEEE 978-1-4244-5848-6, pp. 203-207, 2010

[10] M. Dzida, M. Zagozdzon, M. Pioro, A. Tomaszewski, “Optimization of the Shortest-Path routing with Equal-Cost-Multi-Path load balancing”, ICTON 2006, IEEE 1-4244-0236-0, pp. 9-12, 2006

[11] P802.1aq/D4.5, “IEEE Approved Draft Standard for Local and Metropolitan Area Networks: Bridges and Virtual Bridged Local Area Networks - Amendment 9: Shortest Path Bridging”, 978-0-7381-7397-9, 1 – 359, April 3 2012

[12] Gero, B.P. ; Farkas, J. ; Kini, S. ; Saltsidis, P. “Upgrading the metro Ethernet network”, Communications Magazine, IEEE , Vol. 51, pp. 193 – 199, May 2013

[13] Nemeth, K., Korosi, A., Retvari, G., “Optimal OSPF traffic engineering using legacy Equal Cost Multipath load balancing”, IFIP Networking Conference, pp. 1-9, May 2013

[14] Lee, S.S.W. , Po-Kai Tseng , Chen, A., Cheng-Shong Wu, “Non-Weighted Interface Specific Routing for Load-Balanced Fast Local Protection in IP Networks”, Communications (ICC), 2011 IEEE International Conference, pp. 1-6 June 2011 .

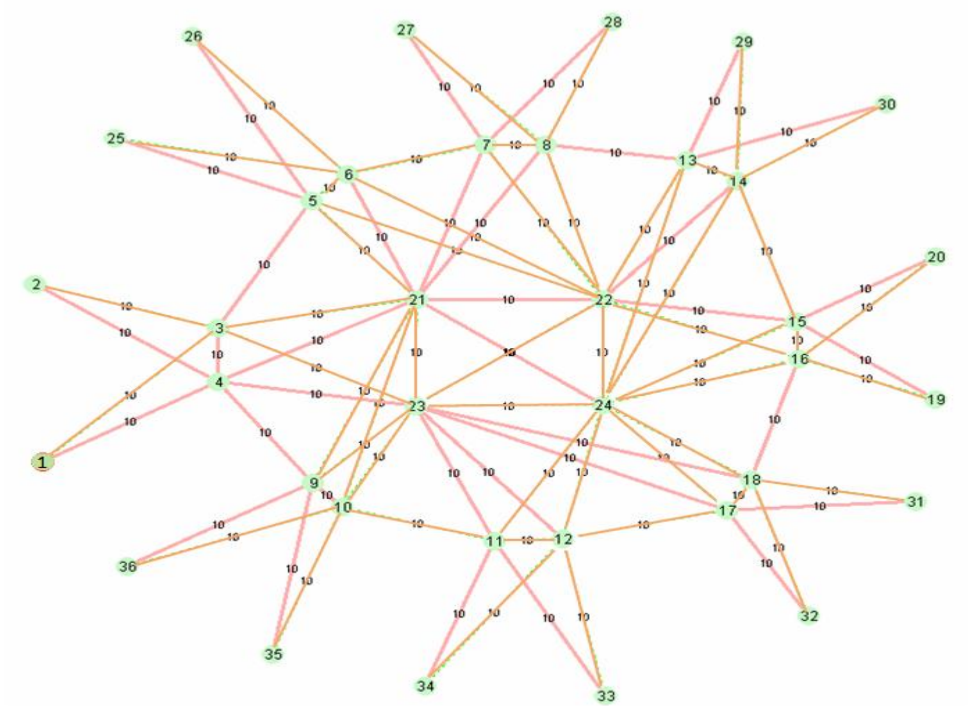

Figure 1. The network 1