

An Analytical Model for Evaluating Spanning Tree Based Layer 2 Routing Schemes

Zhen Luo*, Changjin Suh**

*School of Computer, Shenyang Aerospace University

**School of Computer Science and Engineering, Soongsil University

luozhen@sau.edu.cn, cjsuh@ssu.ac.kr

Abstract—Ethernet has spread its territory into metro and backbone networks. Since spanning tree protocol (STP) is impractical for backbone networks routing, many spanning tree based Layer 2 routing schemes have been proposed, such as multiple spanning tree protocol (MSTP) and MSTP based shortest path bridging (MSTP-SPB). We have developed three analytical models for evaluating these routing schemes, but they failed to analyze Intermediate System to Intermediate System protocol based shortest path bridging (ISIS-SPB), which forwards data from source to destination by multiple shortest path trees. This paper improves previous models by redefining the routing scheme variables, and presents an analytical model for evaluating Layer 2 routing schemes even for multi-path routing. Numerical results from this analytical model verify that the ISIS-SPB is much better than MSTP-SPB and STP.

Index Terms—Analytical model, Layer 2 routing, Spanning tree, STP, MSTP, MSTP-SPB, ISIS-SPB

I. INTRODUCTION

Ethernet is initially designed for data transport in LAN. Over the years, Ethernet has become the default Layer 2 (Data Link Layer) mechanism. To date, Ethernet has spread its territory into Metro Area Network (MAN) as well as Provider Backbone Bridged Network (PBBN).

A PBBN connects separate metro networks and other backbone networks. It consists of two types of bridges, Backbone Edge Bridges (BEB) and Backbone Core Bridges (BCB). BEBs are located at the edge of network, they are in charge of bridging metro networks or small backbone networks and reformatting frames. BCBs are located in a core network and forward frames in large scale. A BEB connects several small bridged networks, whereas a core network integrates BEBs. A BEB receives frames and forwards them to other BEBs via backbone core network. For simplicity, edge bridges, core bridges and backbone network are used for describing BEBs, BCBs, and PBBN respectively. Throughout the whole paper, the term node is used interchangeably with a bridge.

Ethernet starts with the Spanning Tree Protocol (STP)[1] for forwarding data in LAN. STP generates a single spanning tree for the whole network, which provides the shortest path for traffic destined to the root bridge. STP is impractical for backbone networks due to its poor link utilization and bad convergence time. All inactive links excluded from the spanning tree are not used at all. Furthermore, STP can not always guarantee the traffic to be forwarded along the shortest path. STAR[2], TRE[3] and TRE+[4] attempt to forward

frames via alternate paths which are shorter than the paths calculated by STP. However, they all run on the basis of STP, so they are still constrained by slow convergence, low utilization ratio and poor throughput.

MSTP, or Multiple Spanning Tree Protocol[5] extends STP by dividing a network into multiple regions and allowing multiple spanning trees in a region. A backbone network can be a central region and every independently bridged network can be a region. There is a common spanning tree that connects all bridges of all regions for control purpose. Data frames may be sent across multiple regions, in each of which a selected spanning tree is used. MSTP reduces failover latency by simply swapping currently selected spanning tree to another. Alternative Multiple Spanning Tree Protocol (AMSTP) proposes to establish spanning trees as many as the number of nodes in an Ethernet Backbone network, so as to maximize the delivery capability. As a result, each AMSTP node has a tree rooted at itself. A set of these trees is referred to as a tree set in this paper.

The IEEE 802.1aq Shortest path bridging (SPB) project started from how to generate the tree set, and proposed the MSTP-SPB[6], [7], [8]. When MSTP-SPB runs, each bridge generates its own shortest path tree (SPT) by acting itself as the root bridge, then each data package is forwarded along the shortest path. Edge Node Divided Spanning Tree (ENDIST)[9] extends the MSTP-SPB by creating multiple tree sets. ENDIST allows an edge node to build up trees as many as the number of its links connected to core nodes. About the same time, the SPB project proposed the ISIS-SPB[10] to generate multiple tree sets. ISIS-SPB uses the Intermediate System to Intermediate System (ISIS) protocol for the network topology discovery, it calculates 16 different equal cost shortest path trees for each bridge, so that multiple shortest paths routing from source to destination become possible, as a result, ISIS-SPB can achieve better link utilization and faster convergence time. Nowadays, ISIS-SPB has been standardized by the IEEE as the evolution of the various spanning tree protocols.

To evaluate spanning tree based Layer 2 routing schemes, we have proposed three analytical models[11]. They are developed for different packet drop policies, and analyze the throughput of network under different routing schemes. Unfortunately, they did not consider the situation where the edge node uses multiple shortest path trees to forward data, so they cannot evaluate the ISIS-SPB routing scheme. This paper addresses this problem by improving the definition of routing

scheme variables, so that our previous proposed models can still be available even for multiple paths routing.

The rest of this paper is organized as follows. Section II gives a whole description of our analytical model. Then, the analytical model is implemented by an linear programming package in section III. Section IV evaluates the Layer 2 routing schemes for a given small PBBN. Finally, a conclusion is made in section V.

II. ANALYTICAL MODEL

Our analytical model is aimed to evaluate the spanning tree based Layer 2 routing schemes in terms of throughput, that means, once network topology, traffic distribution and packet drop policy are specified, then the analytic model is going to calculate the network throughput for different routing schemes. In this section, we describe the definition of our model from the aspect of network topology, traffic distribution, packet drop policies, and routing schemes.

A. Symbol definition

An analytical model is composed of a set of mathematical equations that describe relationships among physical characteristic variables, so we firstly define some basic variables for the network, which are itemized as below.

- N : a set of entire nodes ($N = N_E \cup N_C$).
- N_C : a set of core nodes ($|N_C| = n_c$).
- N_E : a set of edge nodes ($|N_E| = n_e$).
- u : the only gateway node to a upper level network, so $u \in N_E$.
- N_P : a set of busy edge nodes that attracts flow more than unbusy edge nodes ($|N_P| = n_p, N_P \subset (N_E - u)$).
- L : a set of entire links in a backbone network.
- $l(i, j)$: a directed link from node i to node j , so $l(i, j) \in L$, where $i, j \in N$.
- $c(i, j)$: the capacity of link $l(i, j)$.

We use variables a and z to denote source edge node and destination edge node respectively. Unless otherwise stated, a and z are two different edge nodes. $|A|$ is the cardinality of the set A .

B. Network traffic assumption

It is usually very difficult to model the network traffic, as there are too many events and too random quantities. However, it is simply easy to define steady state traffic.

We assume the steady state network traffic as follows. The edge bridge a has a constant inflow traffic $K(a)$ from outer bridged networks, it is going to send the traffic $g(a, z)$ to the destination edge bridge z , so $K(a) = \sum_{\{z|z \in N_E - a\}} g(a, z)$, where $a, z \in N_E, a \neq z$. Specially, each edge bridge sends data packets to gateway bridge u with the same upward traffic ratio r ($0 \leq r \leq 1$), then the rest $(1 - r)$ data packets will be sent to other edge bridges. The ‘busy’ edge bridge is assumed to receives data traffic m ($m \geq 1$) times as much as ‘unbusy’ bridge does. If there is no ‘busy’ edge bridge in the network, data traffic distribution is considered to be even, otherwise, it is considered to be uneven. For simplicity, it is assumed that

both busy edge bridges and unbusy edge bridges are loaded with same amount of input flow from outer bridged network. The gateway edge node u receives traffic from upper level network as much as the traffic it sends to upper level network.

Based on the traffic distribution stated above, the loaded traffic $g(a, z)$ for all pairs of a and z can be calculated and summarized in TABLE I.

TABLE I: Loaded traffic $g(a, z)$ for all pairs of a and z .

Des. z Src. a	u	N_P	$N_E - N_P - u$
u	•	$\frac{m \cdot K(a)}{n_e + (m-1)n_p - 1}$	$\frac{K(a)}{n_e + (m-1)n_p - 1}$
N_P	$e \cdot K(a)$	$\frac{m \cdot (1-e) \cdot K(a)}{n_e + (m-1)n_p - m - 1}$	$\frac{(1-e) \cdot K(a)}{n_e + (m-1)n_p - m - 1}$
$N_E - N_P - u$	$e \cdot K(a)$	$\frac{m \cdot (1-e) \cdot K(a)}{n_e + (m-1)n_p - 2}$	$\frac{(1-e) \cdot K(a)}{n_e + (m-1)n_p - 2}$

C. Packet drop policy

We represent a network as a system of connected bridges. A bridge is regarded as a black-box with constant service time for incoming packets and infinite capacity buffers, and it consumes received packets immediately. Normally, a bridge processes the incoming packets with the below procedure.

- 1) Group the incoming packets by their outgoing links.
- 2) Sum the total traffic that flows to the same outgoing link, and compute the overflowed traffic to be dropped.
- 3) Apply the drop policy on the affected inflows.

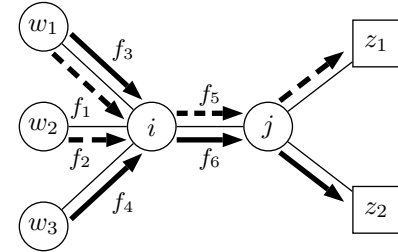


Fig. 1: Data forwarding at node i

Fig. 1 demonstrates how data flows forwarded, where node i is executing the packet drop procedure, the data flow denoted with dashed arrow line is destined to z_1 , the data flow drawn with solid arrow line is destined to z_2 . After step 1, inflows f_1, f_2, f_3 and f_4 are grouped together, as they are being forwarded to the same outgoing link $l(i, j)$. Then, step 2 calculates the total traffic as $f_1 + f_2 + f_3 + f_4$, and overflowed traffic as $f_1 + f_2 + f_3 + f_4 - c(i, j)$. Finally, step 3 will apply the packet drop policy on the affected flows.

Our previous three analytical models are defined for different packet drop policies. Both pathwise fair drop policy[11] and maximum throughput drop policy[12] allow packet dropping only at edge node and no packet loss at core node, while linkwise fair drop policy[11] allows packet dropping at any node. Take Fig. 1 for example, when node j takes either pathwise fair drop policy or maximum throughput drop policy, it should satisfy $f_5 = f_1 + f_2$ and $f_6 = f_3 + f_4$.

When linkwise fair drop policy is adopted, if the sum of f_1 , f_2 , f_3 and f_4 is larger than $c(j, k)$, then $f_5 = \frac{(f_1+f_2) \cdot c(j, k)}{f_1+f_2+f_3+f_4}$ and $f_6 = \frac{(f_3+f_4) \cdot c(j, k)}{f_1+f_2+f_3+f_4}$. All these policies are included in this model.

D. Routing scheme

For defining routing scheme, our previous models use variable $B(a, z)$ to describe a set of active links that belongs to the spanning tree used to deliver the traffic from a to z , variable $d(a, z)$ to represent the traffic successfully delivered from source a to destination z , and variable $p(i, j, z)$ to denote the sum of flow destined to z on the link $l(i, j)$. These variables can clearly define the routing scheme for STP and MSTP-SPB, but fail to define ISIS-SPB routing scheme. For making up, we introduce the concept of tree set. MSTP-SPB makes each node have a shortest path tree rooted at itself, all these trees are generated by the same tie-breaking algorithm, a set of these trees is referred to as tree set. ISIS-SPB can generate 16 tree sets with different tie-breaking algorithms, then each node has 16 trees for forwarding data. To clearly define the routing scheme constructed by these tree sets, we introduce some extra variables shown as below.

- T : a set that contains all the tree sets in the network.
- x : the index of a tree set in T , it ranges from 0 to $|T|-1$.
- $t(x)$: a tree set in T , and it is indexed by x , so $t(x) \in T$ and $T = \bigcup_{x=0}^{|T|-1} t(x)$.

Also the variables $B(a, z)$, $d(a, z)$ and $p(i, j, z)$ should be improved as below.

- $B(x, z)$: a spanning tree in tree set $t(x)$, it is adopted to forward packets to z . It is a set of links that construct the spanning tree.
- $d(x, a, z)$: the traffic successfully delivered from source a to destination z by the spanning tree whose tree set index is x .
- $p(x, i, j, z)$: the traffic forwarded to z by the link $l(i, j)$ through the spanning tree whose tree set index is x .

Base on these new variables, we can efficiently define any spanning tree based Layer 2 routing schemes, especially the ISIS-SPB routing scheme. Next section will give a detailed implementation when pathwise fair drop policy is adopted.

III. LINEAR PROGRAMMING MODELING

For pathwise fair drop policy, we are using linear programming modeling to implement the proposed model. All the equations needed for linear programming modeling are listed as below. Equation (1) is the objective function, and equations from (2) to (13) are the conditions to be satisfied, in which a , z_1 and z_2 are three different edge nodes, $z_1 \in (N_E - u)$ and $z_2 \in (N_E - N_P - u)$.

Maximize

$$\sum_{x=0}^{|T|-1} \sum_{\{z|z \in N_E\}} \sum_{\{a|a \in N_E-z\}} d(x, a, z) \quad (1)$$

subject to

$$K(z_1) = K(z_2) \quad (2)$$

$$g(a, z_1) = \begin{cases} g(a, z_2) & \text{if } z_1 \in (N_E - N_P - u) \\ m \cdot g(a, z_2) & \text{if } z_1 \in N_P \end{cases} \quad (3)$$

$$g(a, u) = r \cdot K(a) \quad (4)$$

$$\sum_{\{z_1|z_1 \in N_E-u\}} g(a, z_1) = (1-r) \cdot K(a) \quad (5)$$

$$K(u) = \sum_{\{a|a \in N_E-u\}} g(a, u) \quad (6)$$

$$g(u, z) = \begin{cases} \frac{K(u)}{n_e+n_p-1} & \text{if } z \in (N_E - N_P - u) \\ \frac{m \cdot K(u)}{n_e+n_p-1} & \text{if } z \in N_P. \end{cases} \quad (7)$$

$$d(x, a, z) = \sum_{\{j|l(a,j) \in B(x,z)\}} p(x, a, j, z) \quad (8)$$

$$\sum_{x=0}^{|T|-1} d(x, a, z) \leq g(a, z) \quad (9)$$

$$d(x, a, z_1) = \begin{cases} d(x, a, z_2) & \text{if } z_1 \in (N_E - N_P - u) \\ m \cdot d(x, a, z_2) & \text{if } z_1 \in N_P \end{cases} \quad (10)$$

$$d(x, a, u) = r \cdot \sum_{\{z|z \in N_E\}} d(x, a, z) \quad (11)$$

$$\sum_{\{j|l(j,i) \in B(x,z)\}} p(x, j, i, z) = \sum_{\{k|l(i,k) \in B(x,z)\}} p(x, i, k, z) \quad (12)$$

$$\sum_{x=0}^{|T|-1} \sum_{\{z|z \in N_E\}} p(x, i, j, z) \leq c(i, j) \quad (13)$$

According to the distribution of traffic mentioned above, we have equations (2) to (7). The successfully delivered traffic never exceed the offered load, so (9) is needed. In this model, we give a strict constraint, requiring the successfully delivered traffic from one edge node to other edge nodes are fair, that means (10) and (11). There is no data loss in core node during data forwarding, then traffic conservation (12) must be satisfied. Definitely, the traffic transferred on a link does not exceed the capacity of the link, so the constraint (13) are required.

IV. PERFORMANCE ANALYSIS

A. Experiment setting

Many metro and backbone networks have been designed and deployed with regular topology or quasi-regular topology[13], because regular topology naturally distribute traffic with balance and provides good network robustness and better throughput. Fig. 2 shows an example topology of backbone networks, in which a box represents an edge bridge and a circle denotes a core bridge. The number inside of either box or circle is bridge identifier. Bridge '11' is the unique gateway bridge in this backbone network, it is in charge of forwarding frames between this network and upper network. The line between two bridges denotes a physical link, each link is full duplex. The number close to each link represents the link capacity, and its unit is Gigabit per second (Gbps). Any link capacity higher than 10Gbps is achieved via link aggregations[14].

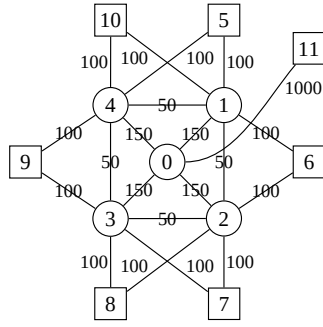


Fig. 2: A small backbone network

In the experiment, the value m is set with 2, the value of r is set with 0.2 and 0.5 for two different traffic distribution respectively. $m = 2$ means the busy edge node is going to attract flow twice than unbusy edge node. $r = 0.2$ implies the data packets is mainly forwarded to edge bridges, and $r = 0.5$ indicates a large amount of data packets flood into the gateway bridge. When network traffic distribution is uneven, bridges '9' and '10' are assumed to be 'busy' bridges. Lingo 11.0 package is used to program and run linear programming modeling.

B. Results and analysis

Fig. 3 to Fig. 6 show all the results with line charts, where the ISIS-SPB is compared with STP, MSTP-SPB and Perfect routing. Perfect routing allows all links in the network to be active for forwarding data frame.

In four figures, the vertical axis indicates the throughput of whole network, and the horizontal axis indicates the data traffic loaded to each edge bridge, denoted with percentage. Each edge bridge is connected to two 100Gbps links, so 100% means the loaded traffic is 200Gbps.

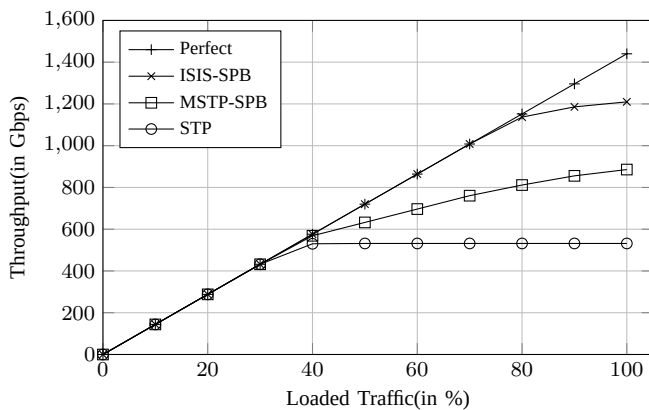


Fig. 3: Throughput when loaded traffic is even and upward traffic ratio r is 0.2

The curves shown in Fig. 3 and 4, demonstrate how the throughput changing with different loaded traffic when the data traffic distribution is even. On different upward ratios, perfect routing performs well without any packet loss, and the ISIS-SPB outperforms MSTP-SPB.

Fig. 5 and 6 shows the throughput as a function of different loaded traffic when the data traffic distribution is uneven. It

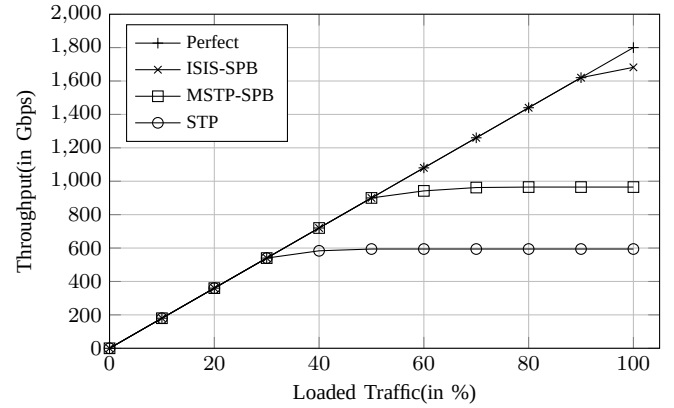


Fig. 4: Throughput when loaded traffic is even and upward traffic ratio r is 0.5

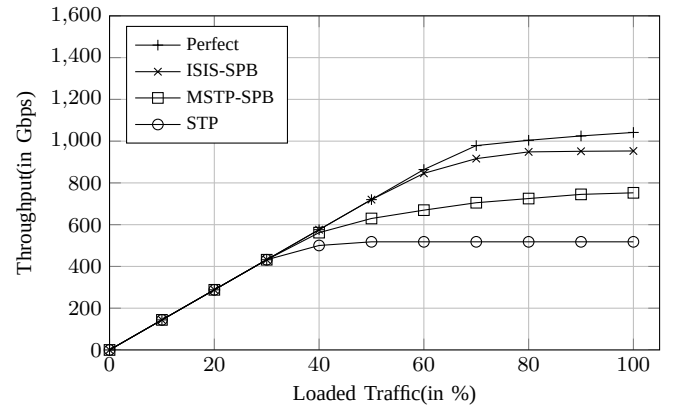


Fig. 5: Throughput when loaded traffic is uneven and upward traffic ratio r is 0.2

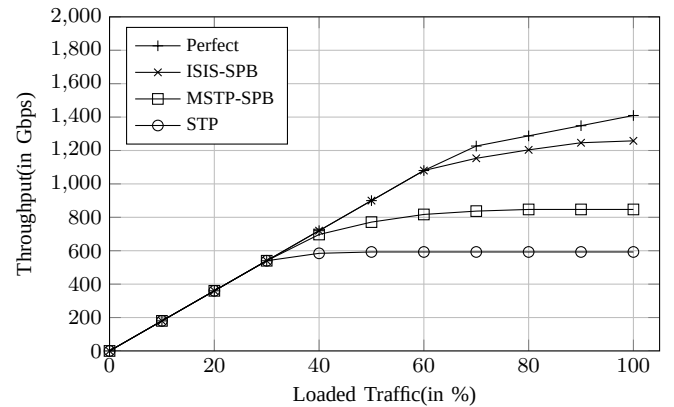


Fig. 6: Throughput when loaded traffic is uneven and upward traffic ratio r is 0.5

also shows that the ISIS-SPB outperforms MSTP-SPB. Due to uneven data traffic distribution, the throughput achieved by perfect routing does not change in linear mode with loaded traffic.

TABLE II compares the ISIS-SPB with MSTP-SPB from the numerical value standpoint when the loaded traffic is at 90%, it take the throughput produced by perfect routing as the reference, so all the values in the 'Perfect' row are all

100%. It is obvious that the ISIS-SPB performs much better than MSTP-SPB, no matter when the network traffic is even or uneven and upward traffic ratio is high or small. The bottom line of the table shows that, ISIS-SPB can achieve much better throughput than MSTP-SPB in the range of 27.7% to 67.9% increase.

TABLE II: Throughput comparison when loaded traffic is at 90% and perfect routing is used as reference

	Even		Uneven	
	$e = 0.2$	$e = 0.5$	$e = 0.2$	$e = 0.5$
Perfect	100%	100%	100%	100%
ISIS-SPB	91.5%	100%	92.8%	92.4%
MSTP-SPB	66.0%	59.6%	72.7%	62.8%
STP	41.0%	36.7%	50.5%	43.9%
$\frac{\text{ISIS-SPB}}{\text{MSTP-SPB}} - 1$	38.6%	67.9%	27.7%	47.1%

V. CONCLUSION

We proposed an analytic model for evaluating the spanning tree based Layer 2 routing schemes in terms of throughput. It introduces tree set variables, and improves the definition of routing scheme variables, so that our previous models can still be available even for multiple paths routing. It also implements a linear programming modeling for pathwise fair packet drop policy. Given a backbone network, we program and run the analytical model by Lingo 11.0 package. Numerical results obtained by this model verify that ISIS-SPB routing is much better than MSTP-SPB in the range of 27.7% to 67.9% increase.

REFERENCES

- [1] R. Seifert, *The Switch Book: The Complete Guide to LAN Switching Technology*. New York, USA: John Wiley & Sons, Inc., 2000.
- [2] K.-S. Lui, W. C. Lee, and K. Nahrstedt, "Star: a transparent spanning tree bridge protocol with alternate routing," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 3, pp. 33–46, 2002.
- [3] G. Ibáñez, A. García-Martínez, J. A. Carral, J. M. Arco, and A. Azcorra, "Evaluation of tree-based routing ethernet," *Comm. Letters*, vol. 13, no. 6, pp. 444–446, 2009.
- [4] J. A. Carral, G. Ibáñez, A. García-Martínez, M. A. López-Carmona, and I. Marsa-Maestre, "Tre+: Extended tree-based routing ethernet," *ETRI Journal*, vol. 32, no. 1, pp. 157–159, February 2010.
- [5] I. T. G. of IEEE 802.1, *IEEE Standard for Local and Metropolitan Networks, Multiple Spanning Trees*, IEEE 802.1s task force Std., 2002.
- [6] J. Farkas and Z. Arato, "Performance analysis of shortest path bridging control protocols," in *GLOBECOM*. IEEE, 2009, pp. 1–6. [Online]. Available: <http://dx.doi.org/10.1109/GLOCOM.2009.5425776>
- [7] N. Finn. (2005, 09) Shortest path bridging. IEEE802.1aq. [Online]. Available: <http://www.ieee802.org/1/files/public/docs2005/aq-nfinn-shortest-path-0905.pdf>
- [8] Z. Luo and C. jin Suh, "Shortest path bridging with euclidean distance," *Telecommunications Review*, vol. 21, no. 6, pp. 1019–1032, December 2011.
- [9] C. Suh, K. Kim, and J. Shin, "Endist: Edge node divided spanning tree," in *10th International Conference on Advanced Communication Technology, ICACCT 2008*, vol. 1, Feb 2008, pp. 802–807.
- [10] "Ieee standard for local and metropolitan area networks—media access control (mac) bridges and virtual bridged local area networks—amendment 20: Shortest path bridging," *IEEE Std 802.1aq-2012*, pp. 1–340, June 2012.
- [11] C. Suh, Z. Luo, D.-y. Kim, and B.-s. Joo, "An analytic model for spanning tree based routing methods," in *ICACCT'09: Proceedings of the 11th international conference on Advanced Communication Technology*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 605–610.

- [12] C. Suh and S. woong Jung, "Layer-2 routing analytic model by linear programming," in *Proc. 10th International Conference on Advanced Communication Technology ICACCT 2008*, vol. 1, Feb. 17–20, 2008, pp. 561–566.
- [13] S. L. Cecilio, J. M. Pedersen, J. M. G. Lopez, M. T. Riaz, and O. B. Madsen, "Using regular topologies for wimax backbone : A case study," in *Proceedings of the International Conference on Image Processing & Communication Challenges (IP&C-2009)*. Warsaw, Poland: Academy Publishing House EXIT, 2009, pp. 466–473.
- [14] I. T. G. of IEEE 802.1, *IEEE Standard for Local and Metropolitan Area Networks: Link Aggregation*, IEEE 802.3ad task force Std., 11 2008.



Zhen Luo was born in China in 1981. He received his B.Sc. and M.Sc. from Shenyang Aerospace University, China in 2003 and 2006 respectively, and obtained his Ph.D. in computer communication from Soongsil University, South Korea in 2012.

He has worked as an Assistant Professor at School of Computer Science and Engineering at Soongsil University, South Korea in 2011. He is now a faculty member in School of Computer at Shenyang Aerospace University. His current research interests focus on Carrier Ethernet, switching and routing

theory, and embedded systems.



Changjin Suh was born in South Korea in 1959. He received his B.Sc. and M.Sc. from Seoul National University, South Korea in 1982 and 1984 respectively, and obtained his Ph.D. in electric and computer engineering from University of Massachusetts in Amherst, U.S.A. in 1996.

From 1985 to 1990, he has worked at ETRI (Electronics and Telecommunications Research Institute), South Korea. In 1997, he joined the School of Computer Science and Engineering in Soongsil University, South Korea. His current research interests focus on sensor networks, Ethernet, switching and routing theory.