Journal of Computer and Communications, 2016, 4, 37-56
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Equal Preference Multi-Path Routing for L2 Hierarchical Networks

Ting-Chao Hou[1], Hsiang-Chi Tsai[2]

[1]Department of Communications Engineering & Center for Telecommunication Research, National Chung Cheng University, Taiwan
[2]Department of Communications Engineering, National Chung Cheng University, Taiwan
Email: ieetch@ccu.edu.tw, magt507@gmail.com

How to cite this paper: Hou, T.-C. and Tsai, H.-C. (2016) Equal Preference Multi-Path Routing for L2 Hierarchical Networks. *Journal of Computer and Communications*, **4**, 37-56.
http://dx.doi.org/10.4236/jcc.2016.414004

Received: September 8, 2016
Accepted: November 12, 2016
Published: November 15, 2016

Abstract

The layer 2 network technology is extending beyond its traditional local area implementation and finding wider acceptance in provider's metropolitan area networks and large-scale cloud data center networks. This is mainly due to its plug-and-play capability and native mobility support. Many efforts have been put to increase the bisection bandwidth in a layer 2 network, which has been constrained by the spanning tree protocol that a layer 2 network uses for preventing looping. The recent trend is to incorporate layer 3's routing approach into a layer 2 network so that multiple paths can be used for forwarding traffic between any source-destination (S-D) node pair. ECMP (equal cost multipath) is one such example. However, ECMP may still be limited in generating multiple paths due to its shortest path (lowest cost) requirement. In this paper, we consider a non-shortest-path routing approach, called EPMP (Equal Preference Multi-Path) that can generate more paths than ECMP. The EPMP is based on the ordered semi-group algebra. In the EPMP routing, paths that differ in traditionally-defined costs, such as hops, bandwidth, etc., can be made equally preferred and thus become candidate paths. We found that, in comparison with ECMP, EPMP routing not only generates more paths, provides higher bisection bandwidth, but also allows bottleneck links in a hierarchical network to be identified when different traffic patterns are applied. EPMP is also versatile in that it can use various ways of path preference calculations to control the number and the length of paths, making it flexible (like policy-based routing) but also objective (like shortest path first routing) in calculating preferred paths.

Keywords

Algebraic Routing, Multipath, ECMP, Policy-Based Routing, Datacenter Networks

1. Introduction

A layer 2 network by definition is a network where its protocol data units (or frames) can be transported from source to destination by using a data-link layer (the second layer in the Open System Interconnection (OSI) layering model) protocol, without the need of a network layer (the third layer in the OSI model) protocol. Ethernet has been the most popular layer 2 network technology, found not only in LANs (Local Area Networks), like campus networks and office networks, but also as the interconnection technology for metropolitan area networks. Ethernet's popularity comes from its "plug and play" capability, requiring minimum amount of configuration. In addition, the not-so-sophisticated transport functions provided by Ethernet enable it to be a commoditized technology with low cost, making it the switching product of choice. However, as Ethernet keeps on growing, with more end hosts from different organizations/tenants in a network, it becomes necessary to improve its scalability and traffic isolation capabilities.

IEEE 802.1Q [1], which defines VLAN (Virtual LAN), was introduced to separate traffic using the VLAN tag (C-VID in **Figure 1**). However 802.1Q VLAN tag is only-12-bit long, which allows only 4094 different VLANs and limits its applicability to service provider networks (in a metropolitan area) and data center networks. A subsequent amendment IEEE 802.1ad, also known as Provider Bridging (PB), then introduced a provider tag (S-VID in **Figure 1**) which is similar to the VLAN tag but is instead used for separating traffic at the provider network level. This hierarchical structure (Q-in-Q) allows each tenant/client's traffic to be carried in its own backbone VLAN while clients maintain control over their own VLANs at their sites. However the Provider Bridging still requires switches in the network to learn host MAC addresses, which becomes a more serious issue as the network size scales up. The next came the IEEE 802.1 ah amendment, known as the Provider Backbone Bridging (PBB). PBB uses service identifier (I-SID) to distinguish different VLAN services in the backbone network. It also adds another layer of MAC address, the backbone MAC address, on top of the host MAC address (MAC-in-MAC). In the provider backbone network, frame forwarding is based on the backbone destination address (B-DA). Therefore, core switches in the backbone network do not need to learn host MAC addresses, greatly increasing the scalability of the network.

Although these standards have increased Ethernet scalability, they still rely on the traditional network wide controlled flooding of frames for host discovery. The Spanning Tree Protocol (STP) and its subsequent variants (e.g., RSTP, MSTP) have long been used in preventing flooding loops in Ethernet networks. However, STP has also been known for several disadvantages. For example, it disables some links for preventing loops, thus does not allow source-destination traffic to take multiple paths; with the tree structure, its paths are inefficient for paths that do not end in the root node; and it converges slowly to a new solution if a node or link fails. Such characteristics are rather undesirable for provider networks or data center networks, where both ample bisection bandwidth and fast recovery are critical in their operations.

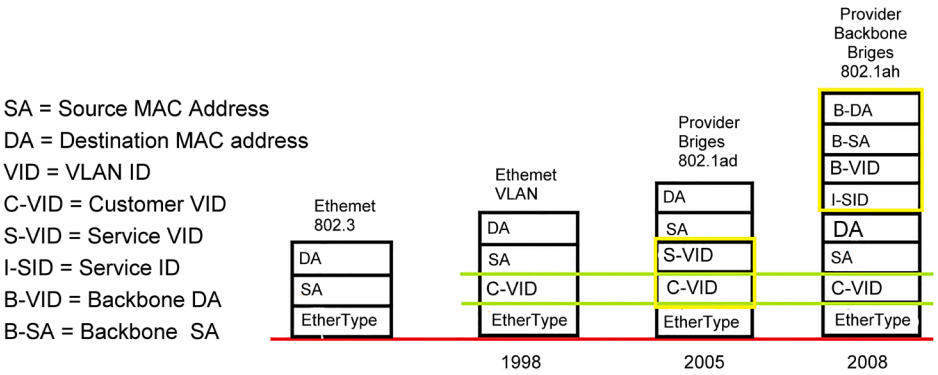


Figure 1. Evolution of layer 2 frame format.

The routing operation which a layer 3 datagram forwarding relies on then comes to rescue. Layer 3 routers run routing protocols and exchange topological information with neighbour routers. A routing protocol has in itself a routing algorithm which calculates a suitable path based on the collected topological information and stores the result in the routing table (or forwarding table). A layer 3 datagram is delivered to its destination based on the routing table. The well-known Dijkstra shortest path first (SPF) algorithm in the link state routing protocols (OSPF [2], IS-IS [3]) is widely used. It can easily be extended to find multiple shortest paths. Compared to STP, these layer 3 routing protocols can provide equal cost multipath (ECMP) [4] and faster reconvergence time. On the other hand, layer 3 datagram forwarding is still undesirable in the sense that it requires IP address space management. When a host moves to a new subnet, its IP address needs to be re-configured, breaking all its on-going connections. In data center networks, where virtual machine (VM) migrations are frequent and service continuity is critical, layer 2 frame forwarding is clearly better than layer 3 datagram forwarding. For the above reasons, the IEEE 802.1 aq (Shortest Path Bridging; SPB) [5] [6] is the new standard that incorporates IS-IS into 802.1 ad and 802.1 ah [7]. Its two modes, SPBV and SPBM, correspond to 802.1ad (Q-in-Q) and 802.1ah (Mac-in-Mac), respectively. IETF also works on the TRILL (Transparent Interconnection of Lots of Links) [8] [9], where its Rbridges (Routing Bridges) runs on layer 2 with the help of the link state routing protocol (IS-IS). Both SPB and TRILL are emerging promising standards for provider networks and data center networks.

In this paper, we concentrate on the multipath routing in layer 2 networks. SPB, TRILL, and most existing data center network routing assume using ECMP for multipath frame/packet forwarding. However, ECMP routing only allows the use of paths with the same cost and this cost is usually expressed by a metric value (e.g., hops). In many network topologies, SPF-based ECMP does not necessarily provide enough path diversity. Non-SPF routing can open up a new horizon where bisection bandwidth can be increased with a smaller cost and alternative topology designs can be explored for different applications or traffic patterns. In addition, given a set of multipaths to use, how these paths are allocated to different clients/services are still an open problem. In

SPB, there is a set of ECT-algorithms [5] that map to the use of different equal cost paths. The mapping is random and does not necessarily evenly spread out the use of available paths [10].

Non-SPF path discovery and selection of feasible paths in a layer 2 network will be the main subjects of this paper. Our study starts with an ordered semi-group algebra along with simple destination-based hop-by-hop forwarding. In a non-SPF routing algorithm, paths are selected based on an attribute value that grades their preference. Paths with equal preference values might be different in the number of hops, bandwidth, etc. thus resulting in non-equal-cost multipaths. Paths values are calculated from link attributes that can be seen as policies characterizing the forwarding behaviour. Amaral et. al. proposed a multipath policy based routing in [11] [12]. It is based on the algebraic routing model in [13] which has been put to practice to model traditional shortest path protocols as well as policy based Border Gateway Protocol. Amaral's policy based routing can be shown to generate more paths than ECMP in most cases. However, we also find that it exhibits a few undesirable characteristics. For example, in certain topologies it might not find any paths while shortest paths do exist. It might also not find all qualified paths due to the semi-group characteristic, to be explained in the next section.

Non-SPF policy based routing has long been used in inter-AS routing protocols (BGP). Its concept or model has not been used in calculating multipaths until the past two or three years. There are still works to be done in policy based multipath routing model such as the trade-offs between the flexibility of the model, the amount of multiple paths that can be used simultaneously, and the network restrictions that must be applied. This paper would investigate a non-SPF routing algorithm that brings together the flexibility in policy based routing and the objectiveness in shortest path first routing.

The rest of this paper is organized as follow. In Section 2, the routing algebra fundamental and link/path attribute assignment are introduced. Based on the preference algebra, Section 3 defines the equal preference multi-path (EPMP) algorithm. Two different flavors of EPMP, EPMP-NH and EPMP-ES are explained. We use two hierarchical network topologies to evaluate the EPMP routing algorithms in comparison with the ECMP algorithm in Section 5. Path numbers and bisection bandwidth are the main performance measures. Finally, the conclusions are given in Section 13.

2. Routing Algebra

Routing in a computer network to most people means finding a shortest path. Indeed, the shortest path routing is used in many routing protocols, e.g., OSPF, RIP, and IS-IS. The term shortest path means a path with the lowest cost, where cost is often a measurable numerical metric like: hop count, path bandwidth, path reliability, latency, and others metrics. Solutions for the shortest path problem are well known [14].

The Internet is a huge interconnection of autonomous systems (AS) or domains, each one independently administered. We see shortest path routing being run within

each domain, so called intra-domain routing. For inter-domain routing, each AS might have its own view of which path should be used, and the path is selected based on policies. Unlike a simple numerical metric, these policies reflect a wider set of characteristics with semantically rich concepts, defining the nature of the paths and their relative preference. Given these characteristics policy routing provides great flexibility in defining route preferences [15]. Border Gateway Protocol (BGP) is a good example.

Since policy characteristic is not necessary a numerical value, the term best path is used instead of the shortest path. With best path, preference of the paths depends on characteristics such as relationships with other nodes, defining in this way a hierarchical order amongst the paths. However current policy routing models cannot take full advantage of the multiplicity of connections to a given destination and are single path in nature [15].

With policy routing two paths that are very different according to traditional numeric metrics can have the same policy characteristics and therefore have the same preference and are considered equally good. Multipaths become more available for policy based routing. Single-path routing protocols have a critical interval after a failure until the algorithm converges to a new solution. On the contrary, in the multipath case in the event of a failure, equally good alternative paths might still be available, therefore reducing the importance of the re-convergence process. Having multiple paths also means that traffic engineering can be achieved by carefully-designed distribution of traffic among those paths, instead of having to play with network metrics to obtain the desired result via routing state manipulations [15].

2.1. Path Preference Algebra

Let a network be represented by a directed graph $G(V,E)$ where V is the set of vertices (nodes) and E is the set of edges (links). Each edge is labelled with an attribute value (e.g., hop count, cost, preference, etc.). A routing algorithm is then characterized by 1) how link attributes are defined, 2) how path attributes are determined when being composed by links, and 3) how legal paths are determined based on the attribute value.

When preference is used as the path attribute, the determination of path attributes can be modelled by a preference algebra. In the preference algebra, a set Γ consists of different possible preference values for links and sub-paths (paths). A sub-path begins with a single link and it becomes a path when it connects the source and the destination. When a sub-path is extended by a link to form a new sub-path (a composition operation), an algebraic operation $\otimes$ composes the preference of the link and the preference of the original sub-path together to form the preference of the new sub-path. Multiple sub-paths between identical two end nodes are ranked (preferred to) by another operation $\oplus$ [12]. Both operations operate on values in Γ. An ordered semi-group is then formed by Γ endowed with two binary operations, $\otimes$ and $\oplus$: $(\Gamma, \otimes, \oplus)$.

As a contrast, the shortest path algorithm can be modelled by (a) setting Γ to $\Re$

(set of real numbers), (b) setting $\otimes$ to + (addition), and (c) setting $\oplus$ to the min (minimum) operation.

2.2. Attribute Values and Preference Ordering

Consider metropolitan area provider networks or data center networks. There typically exists a hierarchical structure in these networks. For metropolitan area networks, there are at least two levels. The outer/lower level has access switches which end hosts are connected to and the inner/upper level where core switches form the backbone network. For data center networks, the fat-tree network has at least three levels (or stages). The lowest level is the edge level, the middle level is the aggregation level, and the upper level is the core level. Figure 2 shows an example of a fat-tree network.

Given such hierarchical networks, it is natural to define three types of attributes in the set Γ:

- D: for links/edges in the downward direction of the hierarchy,
- U: for links/edges in the upward direction of the hierarchy,
- S: for links/edges that connects switches at the same level of the hierarchy.

Two addition attributes are needed to describe trivial paths and invalid/nonexistent paths.

- $\mathbf{1}$: a trivial link/path (self to self),
- $\mathbf{0}$: an invalid path or non-existent link.

Therefore, we have

$$\Gamma = \{\mathbf{1}, D, S, U, \mathbf{0}\}.$$

As to the path attribute resulting from the $\otimes$ operation on links and sub-paths, Table 1 shows a possible $\otimes$ operation proposed in [12]. The path preference ranking is then set to be

$$\mathbf{1} \prec D \prec S \prec U \prec \mathbf{0},$$

meaning downward path is preferred over same level path, and same level path is preferred over upward path.

Table 1 in fact defines a non-decreasing composition operation. Note that the composition operation here represents composing a new sub-path by appending a link a to a sub-path b (A sub-path starts as a link). The term non-decreasing refers to the new attribute in the preference order after the composition operation. For example, by appending a D link to a D sub-path, the new sub-path preserves the D attribute (unchanged in order); while appending a S or U link to a D sub-path results in a S or U sub-path (take the higher in order), respectively. When appending a D link to an S or U sub-path, the new sub-path becomes invalid (the highest in order but least preferred), not S or U, the higher in order. This is necessary in order to avoid forming loops. This non-decreasing property also helps increasing the number of available paths. Take the $D \otimes D = D$ for example. A path with two D links would be equally preferred as the path with three D links.

With the above rules, there however still exists a looping scenario. If the two opposite links connecting two same-level nodes are both labelled S as shown in Figure 3, then loops exist. In Figure 3(a), path 0-2-3-1, path 0-2-3-2-3-1 (having loop), path 0-2-3-2-3-2-3-1 (having loop), all are equally good and have path attribute *U*. In Figure 3(b), path 0-2-3-1, path 0-2-4-3-1, path 0-2-3-4-2-3-1 (having loop), path 0-2-3-4-2-4-3-1 (having loop), all are also equally good and have path attribute *U*.

One solution is to distinguish the two opposite same-level links as *R* and *L*, with *R* being preferred over *L*. The revised routing algebra is now:

$$\Gamma = \{\mathbf{1}, D, R, L, U, \mathbf{0}\}$$

$$\mathbf{1} \prec D \prec R \prec L \prec U \prec \mathbf{0} \tag{1}$$

with the composition operation shown in Table 2.

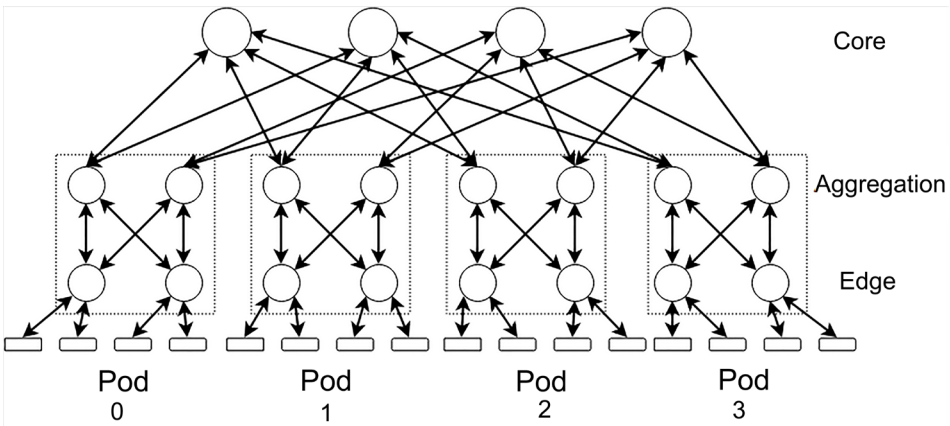
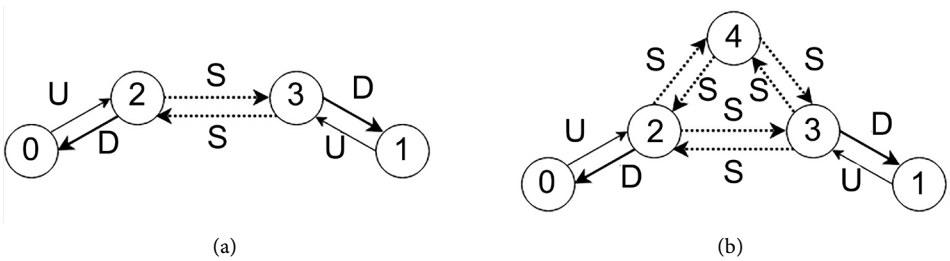


Figure 2. A three-level fat-tree network.

Table 1. The $\otimes$ composition operation.

$\otimes$	**1**	D	S	U	**0**
1	**1**	D	S	U	**0**
D	D	D	**0**	**0**	**0**
S	S	S	S	**0**	**0**
U	U	U	U	U	**0**
0	**0**	**0**	**0**	**0**	**0**



(a) (b)

Figure 3. Loops exist with link attribute S on opposite-direction same-level links.

When opposite-direction links are distinguished with *R* and *L*, loops can be avoided. With the new labelling as shown in Figure 4, only path 0-2-3-1 is valid in case (a), and only paths 0-2-3-1 and path 0-2-4-3-1 are valid in case (b), all with path attribute *U*.

There is another way of preventing loops from forming on same-level links. A same-level link attribute S1 (same-level once) can be used for it to not co-exist with other same-level links (including *R* and *L*) on the same path. Attribute S1 can also be used to prevent same level sub-path to extend too long. We will see its use later.

3. Equal Preference Multi-Path

3.1. EPMP-Next Hop

Equal preference multi-path calculation based on the above preference algebra can then be executed as follows [11] [15]. Given a network topology, there is a corresponding adjacency matrix A of dimension $|V|\times|V|$.

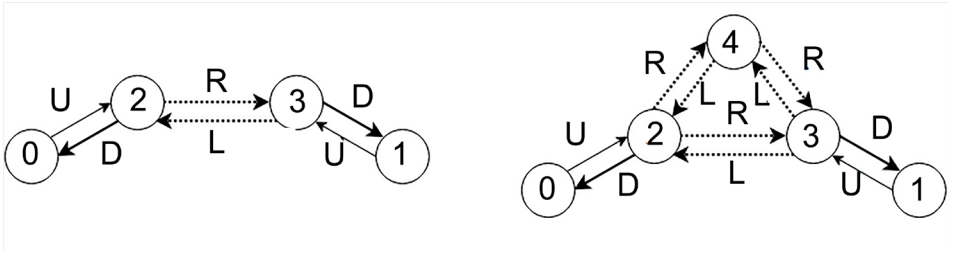
$$A=\left[a\left(e_{i,j}\right)\right], i, j \in V, e_{i,j} \in E$$

where $a_{i,j}=a\left(e_{i,j}\right)$ is the i, jth element of A and represents the link attribute of link $e_{i,j}$. The trivial link $e_{i,i}$ is labelled link attribute $\mathbf{1}$. Element $a_{i,j}$ of A gives the one-hop path attribute from node i to node j. The product of $A \odot A = A^2$ has its i, jth element defined as

$$\left(a_{i,0} \otimes a_{0,j}\right) \oplus \left(a_{i,1} \otimes a_{1,j}\right) \oplus \cdots \oplus \left(a_{i,|V|-1} \otimes a_{|V|-1,j}\right)$$

Table 2. The revised $\otimes$ composition operation.

$\otimes$	1	D	R	L	U	0
1	1	D	R	L	U	0
D	D	D	0	0	0	0
R	R	R	R	0	0	0
L	L	L	L	*L*	0	0
U	U	U	U	U	U	0
0	0	0	0	0	0	0



(a) (b)

Figure 4. Loops can be avoided with link attribute R and L on opposite-direction same-level links.

where $\oplus$ is the path selection operation according to the preference order $\prec$ defined in (1). The i, jth element of A^2 then specifies all preferred paths from node i to node j within 2 hops. By continuing left matrix multiplication (meaning the path is composed from destination node back to the source node)

$$A^{k+1} = A \odot A^k, \text{ for } k \in N$$

until it converges, we obtain attributes of all preferred paths from any source node to any destination node. During the process of matrix multiplication, it is also necessary to build the next-hop matrix that records all next-hop nodes to be used in constructing the preferred path. The routing can then be implemented as hop-by-hop routing by each node looking up its next-hop matrix for forwarding a packet (or frame). We call this method EPMP-NH (EPMP Next Hop).

As an example, consider a 5-node topology as shown in Figure 5. For ease of viewing, we use a double-arrowed link to represent the two opposite links. There are three levels in the topology. The adjacency matrix is

$$A = \begin{bmatrix} \mathbf{1} & U & R & D & D \\ D & \mathbf{1} & D & D & D \\ L & U & \mathbf{1} & D & \mathbf{0} \\ U & U & U & \mathbf{1} & L \\ U & U & \mathbf{0} & R & \mathbf{1} \end{bmatrix}.$$

The matrix multiplication converges at $k = 2$, *i.e.*, $A^2 \neq A$ but $A^3 = A^2$. The next-hop matrix for the topology is

$$\begin{bmatrix} - & 1 & 2 & 3 & 4 \\ 0 & - & 2 & \{0,2,3\} & \{0,4\} \\ 0 & 1 & - & 3 & 0 \\ \{0,1,2\} & \{0,1,2\} & \{0,1,2\} & - & 4 \\ \{0,1\} & \{0,1\} & \{0,1\} & 3 & - \end{bmatrix} \quad (2)$$

We make two observations: 1. EPMP-NH finds a total of 32 paths while ECMP finds a total of 24 paths. Among them, 21 paths are found by both. 2. There are three valid paths that EPMP-NH fails to find. They are path 3-0-1-2, path 3-2-1-0, and path

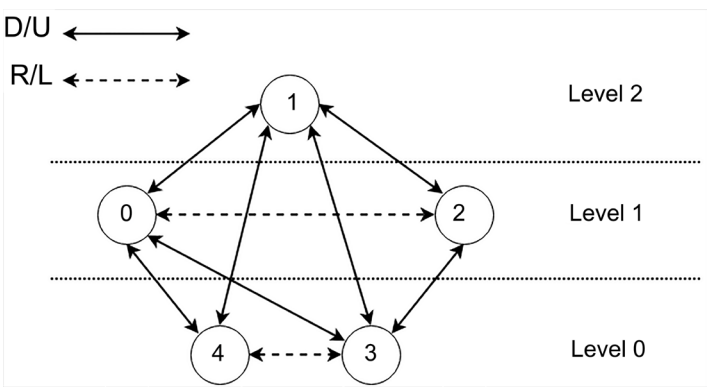


Figure 5. A 5-node topology with 3 levels.

4-0-1-2. To explain why these three paths are not found by EPMP-NH, we take the path 3-0-1-2 for example. From the next-hop matrix in (2), we know that for node 3 to reach node 2, node 3 can take nodes 0, 1, 2 as the next hop. For node 0 to reach node 2, node 0 can only take node 2 as the next hop. Therefore the only path from node 3 to node 2 via node 0 is path 3-0-2, which has a path attribute of U. Path 3-0-1-2 is equally good because it also has path attribute of U. Why it was not found is because path 0-2 (attribute R) is preferred over path 0-1-2 (attribute U) during the first matrix multiplication. This example shows that EPMP-NH may not find all valid paths because certain information is overridden during the path composition process.

3.2. EPMP-Exhaustive Search

It is possible to find all valid EPMP paths by using a brute-force approach. By starting from the destination node and branching out backward towards the source nodes via adjacent links, it allows for all possible paths to be examined. In appending an adjacent link to a sub-path using the algebra such as defined in **Table 2**, the resulting sub-paths that are at least as preferred as the final path preference value, which is available from the converged left matrix multiplication, are kept. Note that the search begins with the destination node because the composition algebra is defined for appending a link to a sub-path. We call this method EPMP-ES (EPMP Exhaustive Search). For the topology in **Figure 5**, EPMP-ES can find all 35 valid paths.

Note that with EPMP-ES, next-hop routing is not applicable. Instead, path routing should be used. SPB allows for path routing by utilizing the PATH_ID to B-VID mapping. The PATH_ID is a concatenation of IDs of switches along the path. An ECT-algorithm [5] [10] is used for identifying the path being chosen. The B-VID carried in the packet/frame header signals the switch which path to forward the packet onto.

3.3. EPMP Path Selection

In contrast to single best path (e.g., SPF) routing, where the forwarding decision is unambiguous, multipath routing needs to be concerned with which path is chosen for which packets/frames. To avoid out-of-order packet handling at the receiving end, that packets of the same flow are assigned the same path is commonly adopted. However, there is still the issue of assigning paths to different flows. For ECMP, with which hop-by-hop packet forwarding is the norm, [4] compares several methods (Round-Robin, Modulo-N Hash, and Highest Random Weight) in assigning flows onto outgoing ports at each immediate switch/router. For EPMP-NH, similar approaches can be adopted.

For EPMP-ES, since it is to be used in path based routing, routing decision is made at the source switch. The methods for next-hop forwarding (Round-Robin, Modulo-N Hash) can still be adopted for load balancing. However, different flows generate different traffic volumes. Different paths may also have overlapping links. Round-Robin or Modulo-N Hash cannot guarantee traffic will be evenly distributed over all paths/links. Instead, if the source switch can have some information about the current

loading of candidate paths, it can choose the least loaded path for a newly arrived flow.

There are several studies on how link loading can be measured. For example, each switch continuously accumulates the byte count of packets that it has forwarded as a basis for calculating link loading [16], and some controller can be employed to collect the link loading in the network [17].

We consider a distributed architecture where no centralized controller is available. When a switch in the core network detects that one of its link is congested (say, near 90 percent in utilization), it will broadcast the link congestion information to all other switches in the network. All switches that receive this congestion notification can then invalidate the paths that contain the congested link. When the link becomes uncongested, (say, below 80 percent in utilization), another notification message will be broadcast so that previously affected paths can be restored to be valid.

The congestion notification can be signalled via IS-IS extension [18]. A new sub-TLV value CONGESTED_LINK can be defined for this purpose. It can be carried in the IS-IS Hello (IIH) packets.

4 . Evaluation of EPMP

We next use two topology examples to evaluate the Equal Preference Multi-Path routing as described in the previous section, in particular in comparison with ECMP.

The first network topology example is a frequently referenced constellation topology in Layer 2 network discussions. As shown in **Figure 6**, it has 36 nodes (switches) in three levels. Among them, 16 are edge switches, 16 are aggregation switches, and 4 are core switches. There are 172 directional links. For this evaluation, links between same level switches all have attribute of S1 to prevent long paths and to control the number of total paths.

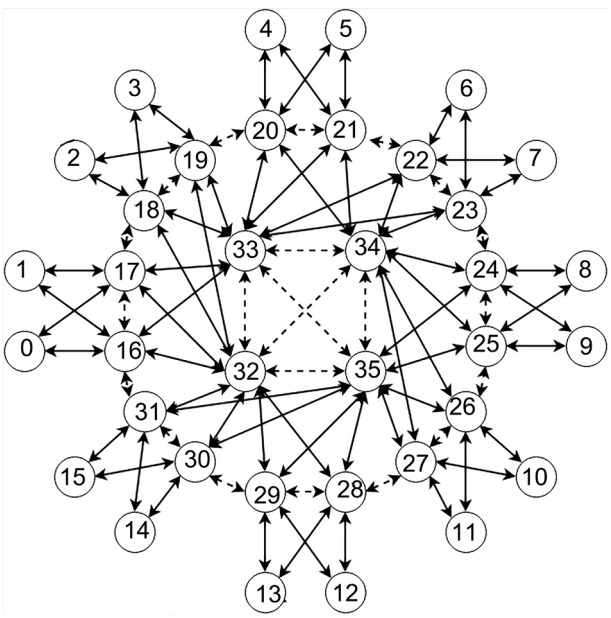


Figure 6. A 36-node constellation topology with 3 levels.

The second network topology example is meant to simulate a small-scale data center network topology with symmetric structure. As shown in Figure 7, it has 52 switches. Among them, 36 are edge switches divided into 4 clusters, 12 are aggregation switches, and 4 are core switches. There are 220 directional links. Attributes R and L are assigned to links between core switches and intra-cluster links. The 4 inter-cluster links are labelled S1 to control the length of paths. They, to some extent, forbid inter-cluster traffic at the aggregation level.

4.1. Path Diversity

Table 3 shows the total number of paths found by ECMP, EPMP-NH, and EPMP-ES, respectively. It also lists the number of source-destination pairs that have only a single path available. Here, we only consider paths between edge switches.

For the constellation topology, EPMP-NH finds 2592 paths while ECMP finds 1760 paths. There are however a total of 3840 valid paths available for EPMP, which are all found by EPMP-ES. Out of the 240 source-destination (S-D) pairs, 64 pairs have only single path by ECMP, while all S-D pairs have at least 2 paths by EPMP-NH and EPMP-ES. For EPMP-ES, every S-D pair has at least 12 paths. For the cluster topology, EPMP-NH finds 16,704 paths while ECMP finds 9900 paths. There are however a total of 49,248 valid paths available for EPMP, which are found by EPMP-ES. Out of the 1260 S-D pairs, 648 pairs have only single path by ECMP, while all S-D pairs have at least 2 paths by EPMP-NH and EPMP-ES. For EPMP-ES, every S-D pair has at least 12 paths.

The SPF based ECMP not only produces less paths, it also does not guarantee multiple paths between all S-D pairs.

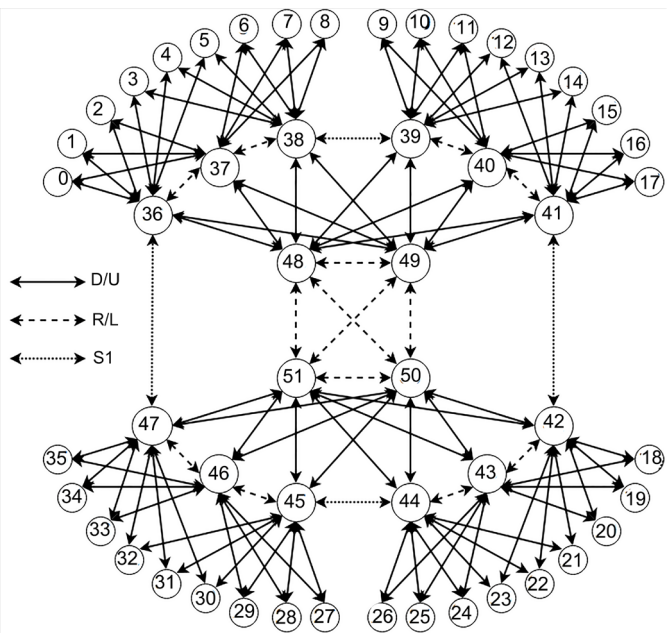


Figure 7. A 52-node cluster topology with 3 levels.

Table 3. Path numbers and single-path SD pairs.

		ECMP	EPMP-NH	EPMP-ES
Constellation	Total paths	1760	2592	3840
	1-path SD	64	0	0
Cluster	Total paths	9900	16704	49248
	1-path SD	648	0	0

4.2. Network Bisection Bandwidth

Throughput is usually the most important performance measure of a network. Network topology, routing algorithm, link capacity, etc. all influence the network throughput. In this section, we use the bisection bandwidth [17] to measure the network throughput. Given a network topology and the link capacity, the total amount of traffic flows that the network can deliver from the source hosts to the destination hosts is defined as the bisection bandwidth. Different traffic patterns and/or routing algorithms will result in different network bisection bandwidth.

First, we assume that two hosts are connected to each edge switch. A fixed number of flows are generated at each source hosts and terminated at some destination hosts. Various flow patterns are considered [17]. They can be either deterministic or random.

- Stride (i): flows from host x are destined to host $(x+i)$ mod N, where N is the total number of hosts.
- Random: flows from host x are destined randomly to one of other $N-1$ hosts.
- Hotspot(n): flows from host x are destined to one of n hosts, where $n < N$.

To evaluate the bisection bandwidth, we note that the concept of a non-blocking network needs to be clarified. In the circuit switching fabric (or interconnection network) design, a non-blocking fabric refers to the one that owns the capability of always finding a path connecting an idle input port and an idle output port. Cross-bar and Clos network [19] are well-known non-blocking switching networks, while Banyan network is a blocking network. Such a non-blocking definition precludes the traffic patterns in which more than one input ports contend for the same output port. Similarly, in a packet switching network, the bisection bandwidth of the network will be reduced if the traffic pattern is imbalanced, *i.e.*, there are hotspots on the destination end. Therefore, for a given flow demand pattern, we need to do a demand shaping first so that a perfect non-blocking switching network can deliver all shaped flow demands. If a target network cannot deliver all shaped flow demands, we know that it is due to the network topology or the routing algorithm. For a given network topology under evaluation, it is then the routing algorithm that is responsible for the difference in bisection bandwidth.

The demand estimation in [17] is exactly a realization of the demand shaping we just discussed. In demand estimation, an $N \times N$ matrix, where N is the number of hosts, stores flow information for each S-D host pair. **Figure 8** shows an example with 4 hosts. Host 0 generates one flow to each of host 1, host 2, and host 3. Host 1 generates two flows to host 0 and one flow to host 2. Host 2 generates one flow to each of host 0 and

host 3. Host 3 generates two flows to host 1. There is an underlying assumption that all hosts have identical network interface with one unit of bandwidth, in and out. For host 0, since the three generated flows compete for the same outgoing interface, each of them can only get one third of the outgoing bandwidth. The reason that they equally split the outgoing bandwidth is due to the assumption that flows are of TCP nature. The AIMD (Additive Increase Multiplicative Decrease) behaviour and fair queueing in the TCP congestion control mechanism will tend to achieve max-min fairness among contending flows. Similarly, flows from hosts 1, 2, and 3 will equally split the outgoing bandwidth as indicated in the flow matrix on the left of Figure 8.

The flow demand matrix will go through the next round of modification by considering the incoming bandwidth contention at the destination hosts. The bandwidth demands on the receiving end of host 0 and host 1 both exceed the receiving interface capacity, one unit. Three flows compete for the bandwidth in both cases. Therefore flows that demand more than one third of the bandwidth will be shaped so that their demands are reduced to one third unit, again based on the TCP fairness mechanism. The results are shown with boldface in the flow matrix on the right of Figure 8. The square bracket indicates that the flow demand has been constrained by the destination host's network interface.

The flow demand matrix then goes through a new round of modification by considering the outgoing bandwidth contention at the source hosts. For this example, since the flow from host 2 to host 0 is constrained by host 0's interface capacity, and the two flows from host 2 do not use up its outgoing interface capacity, the flow from host 2 to host 3 can then increase its demand to two-third unit, as shown in Figure 9.

The flow demand matrix iteratively checks the constraints set by the source host interface and the destination host interface until no more demand modification occurs. The converged flow demands are then used to test the bisection bandwidth of the network.

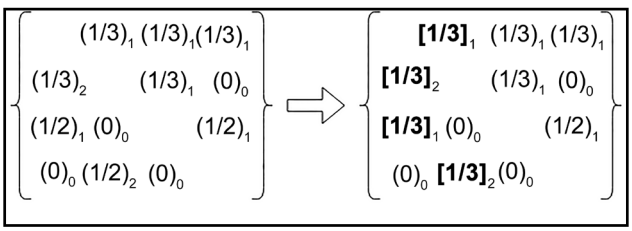


Figure 8. Demand shaping: iteration 1.

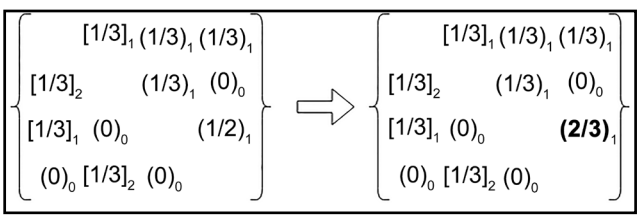


Figure 9. Demand shaping: iteration 2.

Figure 10 compares the bisection bandwidths resulting from employing different routing algorithms in the constellation network with stride traffic patterns. Figure 11 compares the bisection bandwidth with random and hotspot traffic patterns. The three routing algorithms, ECMP, EPMP-NH, EPMP-ES, all use the Modulo-N Hash to decide which path to forward the packets. The fourth routing algorithm, EPMP-ES (least-loaded), is the path routing that chooses the least-loaded path. We also provide the bisection bandwidth for when an ideal non-blocking network is used. It is for reference purpose.

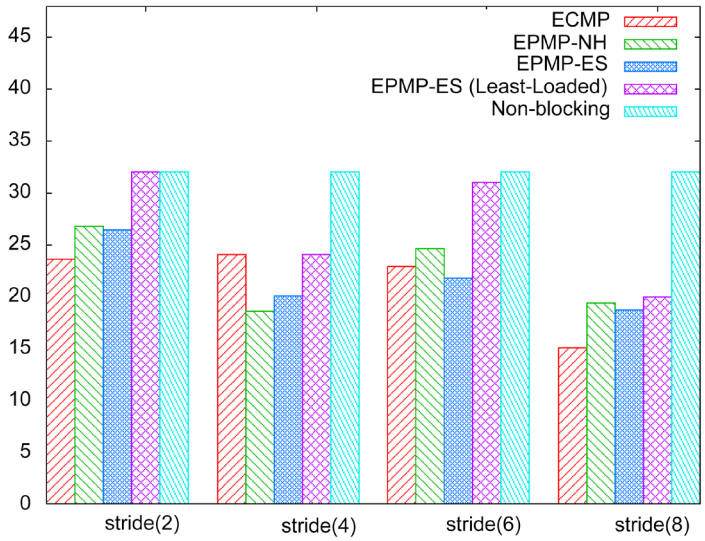


Figure 10. Bisection bandwidths for different routing algorithms with the 36-node constellation topology (stride traffic patterns).

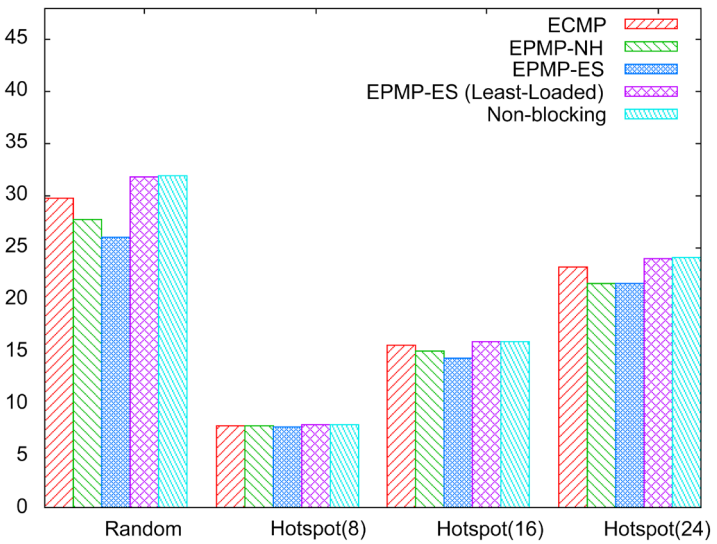


Figure 11. Bisection bandwidths for different routing algorithms with the 36-node constellation topology (random and hotspot traffic patterns).

From Figure 10 and Figure 11, we observe that EPMP-NH and EPMP-ES do not necessarily result in higher bisection bandwidth than ECMP even though they provide more paths. Those additional paths are longer paths than the shortest ones. If used arbitrarily, the average link utilization becomes higher, causing some highly demanded links to become congested earlier.

EPMP-ES (least-loaded) always results in higher bisection bandwidth than the other three. This is because it uses the longer paths with discretion. In certain traffic patterns, it achieves the same result as that of the non-blocking network. For Stride (4) and Stride (8), there is a bigger gap between them. This is mainly due to the intrinsic weakness in the 36-node constellation topology, where it does not have adequate links/paths for Stride (4) and Stride (8) traffic patterns. The Stride (i) traffic patterns in combination with EPMP-ES (Least Loaded) routing method serve as a useful tool to identify the bottleneck links in network topology design.

For hotspot traffic patterns, the differences in bisection bandwidth between different routing algorithms are smaller. This is because flow demands have been shaped to lower values due to traffic contention. The networks have relatively more bandwidths to carry the flow demands, making the routing algorithm a less significant factor.

Figure 12 compares the bisection bandwidths resulting from employing ECMP, EPMP-NH, and EPMP-ES (least loaded path routing) algorithms in the 52-node cluster network with stride traffic patterns. Since there are 36 edge switches and two hosts are connected to each edge switch, stride (i) for i around 36 is a cross traffic pattern and stride (i) for i close to 1 or 71 is a local traffic pattern. We observe that the two EPMP algorithms give higher bisection bandwidth than ECMP with local traffic, but the other way around with cross traffic. In both cases, EPMP-ES (least loaded path routing) is always better than EPMP-NH in providing higher bisection bandwidth. The reason that EPMP is worse than ECMP with cross traffic patterns is due to the insufficient links bandwidth in the network core. This is evident from the lower bisection bandwidth with cross traffic patterns than that with the local traffic patterns for all three routing algorithms. When there are no extra links for non-SPF routing, the longer paths in EPMP use up link capacity faster than the SPF based ECMP and suffer lower bisection bandwidth consequently.

With the random traffic pattern, the average bisection bandwidth for EPMP-ES (least loaded path routing), EPMP-NH, ECMP are 43.87, 38.03, and 37.88, respectively. EPMP-ES can achieve 15 percent more bisection bandwidth than ECMP.

4.3. EPMP vs. Network Topology

EPMP in general should generate more paths than ECMP, but not always. Consider a 4-node network topology with 2 levels as shown in Figure 13. The adjacency matrix is

$$A = \begin{bmatrix} \mathbf{1} & R & R & U \\ L & \mathbf{1} & L & U \\ L & R & \mathbf{1} & \mathbf{0} \\ D & D & \mathbf{0} & \mathbf{1} \end{bmatrix}.$$

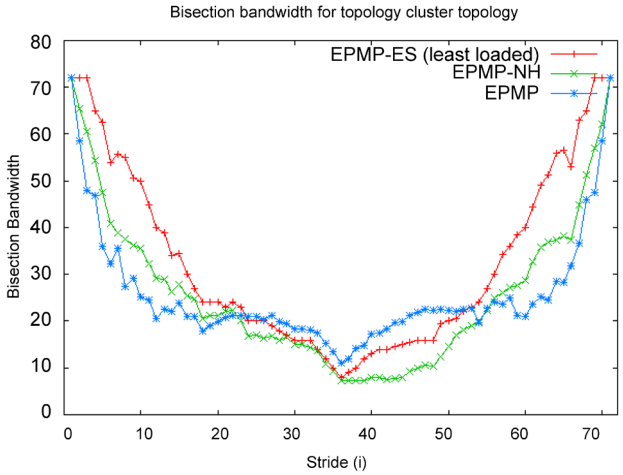


Figure 12. Bisection bandwidths for different routing algorithms with the 52-node cluster topology (stride traffic patterns).

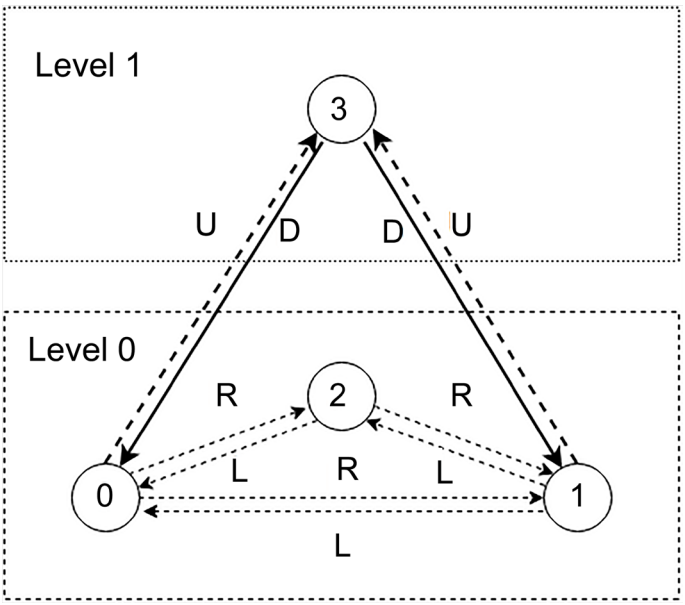


Figure 13. A 4-node topology with 2 levels.

In the first iteration of matrix multiplication, we obtain $A^2 = A$, and therefore finding the solution. The next-hop table for the topology is

$$N = \begin{bmatrix} - & \{1,2\} & 2 & 3 \\ \{0,2\} & - & \{0,2\} & 3 \\ 0 & 1 & - & \{\} \\ 0 & 1 & \{\} & - \end{bmatrix}$$

We make two observations: 1. There is no valid path between node 2 and node 3 based on EPMP. 2. EPMP-NH finds a total of 13 paths while ECMP finds a total of 14 paths. They both find the following 10 paths: 1-0, 2-0, 3-0, 0-1, 2-1, 3-1, 0-2, 1-2, 0-3,

and 1-3. EPMP-NH allows additional 3 paths: path 1-2-0, path 0-2-1, and 1-0-2; while ECMP finds additional 4 paths: path 3-0-2, path 3-1-2, path 2-0-3, and path 2-1-3.

What is special about this example is that node 2 needs to rely on other same-level nodes (nodes 0 and 1) to reach a higher level node 3. In addition, the paths in consideration can end in different levels, unlike what we have been assuming in the constellation and cluster networks where paths end on the lowest level (edge switches). This is to illustrate a potential deficiency in EPMP if paths in consideration do not terminate on edge switches.

5. Conclusions

We have investigated the Equal Path Multi-Path routing algorithm, which is a non-SPF routing algorithm based on ordered semi-group preference algebra. We showed that its use in hierarchical networks, like the data center networks, provides several benefits. EPMP can provide higher throughput (bisection bandwidth) than ECMP because it allows more paths to be used. By comparing EPMP with ECMP, we showed that EPMP not only provides more paths (up to 2 times for the constellation topology and 5 times for the cluster topology) to increase network bisection bandwidth (by 10 percent on average for the constellation topology and 15 percent on average for the cluster topology), allows a variety of policies to be exercised (by manipulating how same-level links are used), but also can be used to identify bottleneck links in the network topology as different traffic patterns are applied.

However, the use of EPMP needs certain caution. For example, EPMP may not find any path for an S-D pair when paths are allowed to terminate at non-edge switches in a hierarchical network. In addition, the original EPMP algorithm (EPMP-NH, which was called multipath policy-based routing in [11] [15]) does not necessarily find all valid paths. EPMP-ES (Exhaustive Search) performs better than EPMP-NH (Next-Hop) because the next-hop path finding approach cannot find all valid paths due to the ordered semi-group algebra. We showed that the EPMP-ES guarantees all valid paths to be found. Path based routing with EPMP-ES can provide higher bisection bandwidth than ECMP and EPMP-NH, and in many scenarios can make the network behave like an ideal non-blocking network. These findings indicate that there is a great potential in using EPMP routing for data center networks and metropolitan networks.

For future work, the flexibility of EPMP can be further investigated. In a hierarchical network, the labelling of same-level links (R, L, S1) and the order of their preference in the composition operation can create a lot of routing policy possibilities for the network administrator to maximize the network bisection bandwidth. In addition, EPMP can be a helpful tool to network topology design in identifying weak links in the existing data center networks and metropolitan networks.

Acknowledgements

This work was supported by grants from MOST (Most 105-2221-E-194-022, MOST 104-3115-E-194-001, MOST 104-2218-E-194-008), Taiwan.

References

[1] IEEE Std. 802.1Q-2011 (2011) IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks.

[2] Moy, J. (1998) OSPF Version 2. IETF RFC 2328.

[3] Callon, R. (1990) Use of OSI IS-IS for Routing in TCP/IP and Dual Environments. IETF RFC 1195.

[4] Hopps, C. (2000) Analysis of an Equal-Cost Multi-Path Algorithm. IETF RFC 2992.

[5] IEEE Std. 802.1aq-2012 (2012) IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment 20: Shortest Path Bridging.

[6] Allan, D., Ashwood-Smith, P., Bragg, N., Farkas, J., Fedyk, D., Ouellete, M., Seaman, M. and Unbehagen, P. (2010) Shortest Path Bridging: Efficient Control of Larger Ethernet Networks. *IEEE Communications Magazine*, **48**, 128-135. http://dx.doi.org/10.1109/MCOM.2010.5594687

[7] Fedyk, D., Ashwood-Smith, P., Allan, D., Bragg, N. and Unbehagen, P. (2012) IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging. IETF RFC 6329.

[8] Perlman, R. (2011) Routing Bridges (Rbridges): Base Protocol Specification. IETF RFC 6325.

[9] Touch, J. and Perlman, R. (2009) Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement. IETF RFC 5556.

[10] Tsao, Y.-C. and Hou, T.-C. (2015) Bridge Priority Provisioning for Maximizing Equal Cost Shortest Path Availability. *Proceedings of* 2015 *IEEE* 16*th International Conference on High Performance Switching and Routing* (*HPSR*), Budapest, Hungary, 1-4 July 2015. http://dx.doi.org/10.1109/hpsr.2015.7483078

[11] Amaral, P., Bernardo, L. and Pinto, P. (2013) Multipath Policy Routing Using Destination Based Hop-by-Hop Forwarding. *Proceedings of* 2013 21*st IEEE International Conference on Network Protocols* (*ICNP*), Gottingen, Germany, 7-10 October 2013. http://dx.doi.org/10.1109/icnp.2013.6733668

[12] Amaral, P., Bernardo, L., Pinto, P. and Julio, F. (2014) An L2 Policy Based Multipath Fabric. *Proceedings of* 2014 *IEEE International Conference on Communications* (*ICC*), Sydney, Australia, 10-14 June 2014. http://dx.doi.org/10.1109/icc.2014.6883881

[13] Griffin, T. and Gurney, A. (2008) Increasing Bisemigroups and Algebraic Routing. In: Berghammer, R., Möller, B. and Struth, G., Eds., *Relations and Kleene Algebra in Computer Science*, Springer, Berlin, 123-137. http://dx.doi.org/10.1007/978-3-540-78913-0_11

[14] Cormen, T., Leiserson, C., Rivest, R. and Stein, C. (2008) Introduction to Algorithms. The MIT Press, Cambridge, Massachusetts.

[15] Chalaca, J. (2013) Multipath Policy Routing in Packet Switched Networks. PhD Thesis, Universidade Nova de Lisboa.

[16] Tso, F., Hamilton, G., Weber, R., Perkins, C. and Pezaros, D. (2013) Longer Is Better: Exploiting Path Diversity in Data Center Networks. *Proceedings of* 2013 *IEEE* 33*rd International Conference on Distributed Computing Systems* (*ICDCS*), Philadelphia, USA, 8-11 July 2013. http://dx.doi.org/10.1109/icdcs.2013.36

[17] Al-Fares, M., Radhakrishnan, S., Raghavan, B., College, W., Huang, N. and Vahdat, A. (2010) A Scalable, Commodity Data Center Network Architecture. 7*th USENIX Symposium on Networked Systems Design and Implementation*, San Jose, USA, 28-30 April 2010.

[18] Smit, H. and Li, T. (2008) IS-IS Extensions for Traffic Engineering. IETF RFC 5305.

[19] Clos, C. (1953) A Study of Non-Blocking Switching Networks. *Bell System Technical Journal*, **32**, 406-424. http://dx.doi.org/10.1002/j.1538-7305.1953.tb01433.x