From Trees to DAGs: Improving the Performance of Bridged Ethernet Networks

Chen Avin, Ran Giladi, Nissan Lev-Tov, Zvi Lotker
Department of Communication Systems Engineering
Ben Gurion University of the Negev
Beer Sheva 84105, Israel
Email: {avin, ran, zvilo}@cse.bgu.ac.il, levtovn@bgu.ac.il

***Abstract*—Ethernet is widely used in Local Area Networks (LANs) due to its simplicity and cost effectiveness. Today, a great deal of effort is being devoted to extending Ethernet capabilities in order to elevate it from a LAN technology to a ubiquitous networking technology, suitable for deployment in Metropolitan Area Networks (MANs) and even in core, Wide Area Networks (WANs). Current standardized Ethernet networks are based on a spanning tree topology, using the *Rapid Spanning Tree Protocol (RSTP)* or *Multiple Spanning Tree Protocol (MSTP)*. The spanning tree architecture is useful for avoiding forwarding loops, but may lead to low link utilization and long failure recovery time. In this paper we propose to shift from tree to Directed Acyclic Graph (DAG) topologies and offer a new bridged Ethernet architecture called *Orient*. Orient is based on assigning an orientation state to each port in the network in order to prevent loops. Thus, the Orient architecture enables a full utilization of all network links and ports, while maintaining simplicity of implementation and compliance with the standardized spanning tree protocols. We provide proofs of the correctness of our protocol and a set of simulations to establish its high efficiency.**

***Index Terms*—Spanning-tree, Ethernet, Bridge, RSTP, MSTP, MAN, VLAN, Forwarding.**

I. Introduction

Ethernet has undergone a significant development in many ways over the past 25 years, both in layer 1 (physical aspects, i.e., media, transmission rate and distance capabilities) and in layer 2 and above, i.e., the sophistication of the bridging system. Ethernet is developing into the preferred networking technology for recent deployments of *Metropolitan Area Networks* (MANs) due to its cost effectiveness, simplicity and inter-operability with *Local Area Networks* (LANs). Typically, MANs are a set of interconnected LANs that work together in order to provide access and aggregation within a metro region. *Virtual LANS* (VLANs) are used in LANs and MANs to separate groups of hosts and networks, and to create broadcast domains within the switched network.

Although Ethernet is a preferred technology for MANs, it has several shortcomings. One of the primary drawbacks is the active topology of a spanning tree, which by definition utilizes at most $n-1$ links in a VLAN of n nodes. This limited utilization causes an imbalance of load on links, which could be problematic in MANs from a performance perspective. Moreover, the use of a spanning tree means that failure of any single active link would disconnect the active topology, resulting in the disruption of network traffic until a new tree is constructed. The rebuilding process requires the activation of previously blocked links, and it can last for several seconds. This is not acceptable in MANs, since high availability is one of the major requirements, particularly in streaming and telecom applications.

Several solutions have been proposed to ease the problem of unbalanced link utilization and the sensitivity to a single spanning tree. In particular, the *Per-VLAN* solution enables the use of a separate spanning tree instance for each VLAN. However, a more scalable solution is required, since VLAN separation does not necessarily balance the utilization of the network links, and a typical MAN must support a large number of VLANs. The *Multiple Spanning Tree Protocol (MSTP)* [5] divides the network into regions, each of which can contain several (up to 65) spanning tree instances, one instance per non-overlapping group of VLANs. In addition to its clear advantages, the MSTP protocol suffers from high complexity, significant additional configuration, and scalability limitations, and the bandwidth across the network is still limited because traffic flows over a small number of superimposed trees [14]. Note also that a VLAN can use one instance of a spanning tree, thus the VLAN's links imbalance utilization and the sensitivity to one spanning tree instance remain unsolved. Solutions for the implementation of MSTP to optimize load balancing and performance issues are provided in [2], [10], [13]. Kern *et al.* [6] propose a traffic engineering framework where the MSTP trees are spanned taking both the traffic conditions and the QoS requirements into account. Sharman *et al.* [13] propose a multi-spanning-tree Ethernet architecture called Viking, which supports multiple spanning trees through VLANs. Their architecture was extended in [8] to enable autonomic adaptation to changing traffic loads.

Another category of solutions are based on routing protocols, in which traffic can traverse least-cost paths rather than being aggregated on a spanning tree backbone, thus providing higher aggregate capacity and more resistance to link failures [14]. *Shortest path routing* has been suggested for Ethernet, e.g., [9], [11], [12]. Kim *et al.* [7] offered a solution based on distributed hash tables to accomplish the shortest path routing on Ethernet networks. The IETF promotes Transparent Interconnection of Lots of Links (TRILL) architecture,

Funded by the European Community's Seventh Framework Programme [FP7/2007-2013], grant agreement no. 215462.

which applies the IS-IS routing protocols at the link layer, transforming bridges to Rbridges (routing-bridges). TRILL provides optimal pair-wise and safe forwarding even during periods of temporary loops, while still enjoying the Ethernet simplicity and zero configuration. Rbridges can work with plain spanning tree bridges, thus allow gradual deployment, and they are also compatible with routers, being invisible to routers like bridges are. However, in order to function, Rbidges communicate using traffic encapsulation with a header that includes a hop count, and various TRILL protocol options. Additionally, the IEEE 802.1aq [4] standard suggests Shortest Path Bridging (SPB) alongside the spanning tree protocol (STP), by implementing the link state IS-IS routing protocol.

In this paper, we propose to increase link utilization by shifting from tree topologies to Directed Acyclic Graph (DAG) topologies which use all edges of the network. We then extend the tree-based forwarding mechanism to prevent loops when forwarding and broadcasting frames. In particular, we present a new Ethernet architecture called *Orient*, which enables multiple paths utilization in the Ethernet, and is based on the common *Rapid Spanning Tree Protocol* (RSTP) [3], using the same Bridge Protocol Data Unit (BPDU) messages. Orient does not use any routing protocol, is not using any encapsulation of the traffic, and is back compatible with the spanning tree bridges, thus is simple to implement, and can be deployed gradually. In essence, instead of limiting the active topology by placing some ports in the discarding state, the Orient protocol assigns an orientation state to each port, and enables utilizing of the port. The Root bridge of the underlying spanning tree will be called the network *Polaris* and all bridge ports will be oriented *north* or *south* according to the Polaris. This orientation assignment provides a directed connectivity over the network, where north ports are used to forward traffic toward the Polaris, while south ports are used to forward traffic toward the leaves of the underlying spanning tree. The Orient architecture utilizes all network links and ports and is capable of avoiding routing loops under what we call *legal path* forwarding. It is also able to provide automatic load balancing and Quality of Service (QoS) support. Unlike other solutions, multiple spanning trees are automatically constructed by the Orient protocol without the use of a heavy aggregated BPDU frames and without the extensive configuration work and management interference. The additional processing time caused by the Orient extension to RSTP is small, while the message complexity does not increase at all. Note that the Orient architecture does not necessarily replace the Multiple Spanning Tree solution or TRILL, as Orient can be implemented on each spanning tree instance of MSTP to increase its efficiency and scalability, and it can work with RBridges that enhance the pair-wise optimal connectivity.

The paper is organized as follows: In sections II-A and II-B we give an overview of the Ethernet bridging and STPs. Section III defines the the Orient architecture and proves its properties. In Section IV we discuss the learning and forwarding process on Orient. We provide two different schemes for broadcasting over Orient, prove their correctness and discuss their pros and cons. In Section V we present simulation results comparing the performance of our proposed mechanisms. Conclusions are then discussed in Section VI.

II. RSTP Architecture

A. Ethernet bridging

Bridges are layer-2 devices that were traditionally designed to partition LANs into LAN segments. Unlike routers, that run longest prefix matching algorithms in their next hop forwarding decisions, bridge forwarding decisions are based solely on table lookups and simple logical operations. Bridges maintain their forwarding information in a *filtering data table* whose entries map MAC addresses to the bridge ports that should be used to forward the relevant frames. The filtering tables are dynamic, where old entries expire and new entries are added on the fly.

When an Ethernet frame arrives at a bridge, two main processes take place; the *learning process* is responsible for adding a table entry for the frame, mapping its source MAC address to the port through which the frame was received. The MAC address is thus ”learned” by the bridge and this information will be used to forward successive frames. The *forwarding process* is responsible for the relay of frames to the port that should be used to forward them to their destination. During the forwarding process a table lookup is performed for the frame destination MAC address, and if the resulting port is active it will be the only candidate for the forwarding process. In case that the relevant MAC address does not appear in the filtering table, the frame is broadcasted, i.e., forwarded through all the active ports, except the port that received the frame. Entries are limited in their life time (the default is 300 seconds) and old entries are deleted from the table. Note that this mechanism fails in the presence of loops, and thus STPs were developed for the control plane of Ethernet bridging [3]. The STPs are responsible for the distributed construction and maintenance of the active topology of the network. The RSTP [3] and MSTP [5] protocols are discussed below. Bridge operations were enhanced in [5] to support the concept of VLANs, adding VLAN registration entries to the filtering tables.

B. RSTP protocol

The spanning-tree algorithm was developed in order to eliminate the problems of loops in Ethernet networks. The protocol designates a loop-free subset of the network topology by placing some ports in a blocking condition. These inactive bridge ports can be activated in the event of a link failure, providing a new path through the network. The active topology is configured in a distributed manner following a distributed version of the Bellman Ford algorithm. It provides a shortest path spanning tree relative to the Root bridge which is calculated when the bridge is powered up and recalculated whenever a topology change is detected.

The tree calculation and maintenance require communication between the spanning-tree bridges, which is accomplished

through the BPDU configuration messages. Bridges exchange configuration messages at regular intervals called *hello time*, (typically one to four seconds). If a bridge or link fails (causing a topology change), neighboring bridges will detect the lack of incoming configuration messages and initiate a spanning-tree recalculation. Configuration messages are exchanged between neighboring bridges, and almost no central or administrator authority exists on network topology. Bridges have unique IDs which are composed from their MAC address and a pre-defined priority for determining the root bridge.

In the beginning of the process of building the spanning tree, each bridge "believes" that it is the root and thus its distance from the root is 0. The ID of the assumed root and the distance to it (denoted *root path cost*) are conveyed in the BPDUs sent by the bridge to its neighbors. In turn, the root bridge ID and the root path cost are updated according to the information received via incoming BPDUs. This is accomplished by the bridges that maintain variables called *priority vectors* and by comparing these priority vectors with the information received in incoming BPDUs. The bridge with the lowest ID is chosen to be the *root bridge*. Each bridge chooses a unique *Root port* which is the port with the minimal cost to the root. Also, a unique bridge is chosen for each LAN, called the *Designated bridge* and it is the bridge that provides the minimum cost to the root for that LAN. A LAN's Designated port is the port of the Designated bridge that connects it to the LAN. The Designated port is the only port allowed to forward traffic from the LAN towards the root. Similarly, the Root port is the only port allowed to forward traffic from the bridge towards the root. Any operational bridge port that is not a root or Designated port is a *Backup port* if that bridge is a Designated bridge for the port's attached LAN, and an *Alternate port* otherwise. An Alternate port offers an alternate path in the direction of the root bridge, whereas a backup port acts as a backup (for the path provided by a Designated port in the direction of the leaves of the spanning tree). Backup ports exist only where there are two or more connections from a given bridge to a given LAN.

Only the Root and Designated ports are placed in a forwarding state and considered part of the active topology. For each bridge the Root port is the port leading towards the root bridge, while the Designated ports lead towards the leaves of the spanning tree. The original spanning tree was enhanced to the Rapid Spanning Tree Protocol (RSTP) in [3] and later to the Multiple Spanning Tree Protocol (MSTP) in [5], providing the ability to have several parallel spanning trees. RSTP provides a faster convergence time and the functionality of detecting those ports that are located in the leaves of the spanning tree.

III. The Orient Architecture

We model our communication network as an undirected graph $G(V, E)$, where V represents the set of bridges and E the set of edges. An edge exists between two bridges if there is a point-to-point link between ports of the two bridges[1].

The result of the RSTP protocol is a spanning tree T of G. The ports of the edges that belong to T are part of the active topology, while the rest of the ports are in blocking mode. Under this topology, a node forwards traffic towards the Root bridge via its single Root port and traffic towards the leaves via its Designated ports. Thus, we can assign the non-blocking ports an *orientation state* with regard to the Root bridge, where the Root ports have a ***north*** orientation and the Designated ports have a ***south*** orientation. We direct each edge toward the Root bridge, and this provides a directed tree over the RSTP active topology.

The basic idea of the Orient architecture is to give orientations to all the ports in the network, not only the ports that belong to the RSTP active topology. We prove that if we give a south orientation to Designated ports[2] and a north orientation to all other ports (Root and Alternate), the resulting topology is a Directed Acyclic Graph (DAG). This implies that one can use the standardized spanning tree protocol (RSTP or MSTP) and assign orientation states to all ports, as described above, in order to maintain a full network topology and still avoid loops. Note that the orientation assigned to the ports does not mean that the communication is restricted to be unidirectional. Our architecture enables bi-directional communication while using the port orientation so as to avoid loops.

A. The Orient Bridge Protocol Extension

The Orient bridge protocol needs only the following simple changes in the standardized RSTP or MSTP protocols.

1) Since our topology is not a tree, the Root bridge will be called the *Polaris*, and the best path cost for each bridge will be calculated according to the Polaris.
2) In the port role selection phase of the protocol, every Root and Alternate port is assigned a north orientation state, and every Designated port is assigned a south orientation state.

Optionally, the north oriented ports on each bridge can be locally ordered according to their accumulated cost to reach the Polaris. Note that all ports belong to the active topology, and each has an orientation; thus all ports are used for data forwarding. The *Orient active topology* is the set of all communication links with the orientation of each port.

B. Orient Active Topology Properties

We first give an observation of a basic property that is a result of the RSTP protocol and the uniqueness of Bridge IDs.

Observation 1. *The root path costs and IDs of the bridges impose a strict total order on the bridges with the Root bridge first in the order.*

[1]We assume an Ethernet network with full duplex point-to-point links between bridges and between bridges and hosts. From 10 gigabits and above, this is the only allowed Ethernet configuration. Our architecture also applies to multi-port LANs, but this setting makes the exposition more complex, since the network's model become a Hyper-graph.

[2]Since we consider point-to-point links, there are no Backup ports.

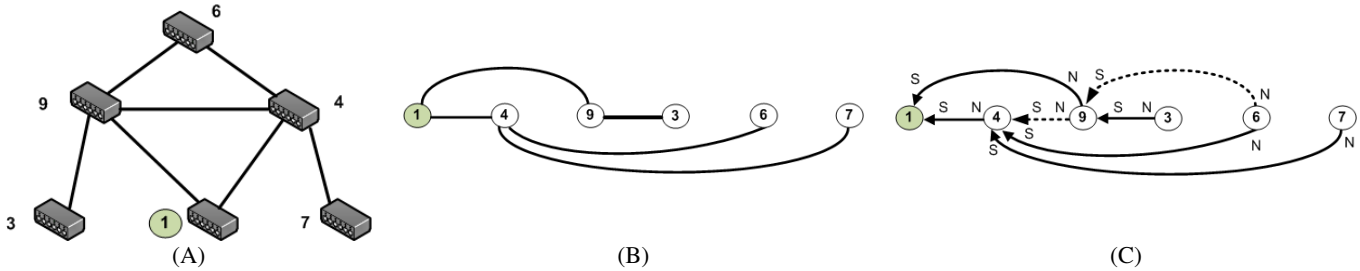

Fig. 1. (A) A communication network example with uniform link and port costs. (B) The result of the spanning tree protocol and its imposed strict total order by root cost paths and bridge IDs. The bridge with the lowest ID, Bridge 1 is the Root bridge. (C) The Orient topology. Dashed edges are edges that do not belong to the spanning tree. North ports are denoted by N and south ports by S. The bridge with the lowest ID, Bridge 1 is the Polaris.

Let $<$ be the following order on the bridges: $u < v$ (i.e., bridge u is smaller than a bridge v), if and only if either the root path cost of u is smaller than the root cost path of v, or the root path costs of u and v are equal and the bridge ID of u is smaller than the bridge ID of v. Clearly, this is a strict total order, in particular, for every two bridges, u, v either $u < v$ or $v < u$.

Fig. 1 (A) depicts an example of a communication network with a uniform cost on the links and ports. Bridge 1 (with the lowest ID) will become the Root bridge of the RSTP. Fig. 1 (B) presents the strict total order imposed by RSTP; bridges are ordered by their root path cost (in this example this is just hop count) and breaking ties by Bridge IDs.

Based on the above observation, we claim that the edges of the Orient active topology have a natural orientation.

Claim 1. *For every edge in G, one of its ports will be assigned north and the other south.*

Claim 1 follows from the fact that all ports in the RSTP start as Designated ports. Each edge connects two bridges, the port of the bridge that is higher in the order (i.e., farther away from the root) will change its role to Root or Alternate, while the other port will remain Designated. Therefore, the Designated port will be south and the other port north. Following Claim 1, the Orient topology assigns an orientation (direction) to each edge in G; the direction of each edge is toward the Polaris bridge. Formally, let $D = (V, A)$ be the directed graph representing the Orient topology, where a directed edge $< u, v > \in A$ if and only if the undirected edge $(u, v) \in E$ and $v < u$.

Claim 2. *The directed graph resulting from the Orient Protocol is a Directed Acyclic Graph (DAG).*

Proof: All directed edges in D are in the same direction according to the strict total order $<$ of the nodes; therefore there are no directed cycles in D. ■

Fig. 1 (C) presents the DAG representing the Orient topology of Fig. 1 (A), where S and N stand for south and north ports, respectively. Note that the Orient active topology itself is not directed and contains all the edges of the communication graph. We now turn our attention to establishing a loop-free bidirectional forwarding mechanism. Over the Orient topology, we define *legal* and *illegal forwarding* as follows:

Definition. *A frame forwarding is* **illegal** *if a frame entering a bridge through a north oriented port is forwarded through a north oriented port and otherwise it is* **legal**.

Thus, the legal forwarding transitions are: north to south, south to north and south to south. A ***legal path*** of a frame, is a sequence of legal forwarding transitions.

Legal forwarding can be seen as a basic property of the RSTP protocol:

Observation 2. *All forwarding transitions over the tree active topology of RSTP are legal, i.e., a frame entering a bridge on a Root port (i.e., north port) will never exit a bridge through a Root (north) port again.*

Obviously, legal forwarding on a tree is loop-free, we now extend this result to legal forwarding on the Orient topology and prove its two main properties: connectivity and loop-free.

Property 1 (Connectivity)**.** *There is a* legal path *connecting any two bridges on the Orient topology.*

Proof: Consider the spanning tree T resulting from the RSTP. All edges of T in the Orient topology will be oriented toward the Polaris (i.e., the Root bridge of the RSTP). Therefore from every bridge there is a directed path going north until it reaches the Polaris. For any two bridges u, v consider their directed paths to the Polaris, and let w be their least common ancestor. The path $u, \ldots w, \ldots, v$ is a legal path. ■

The second main Property is the following:

Property 2 (Loop-free)**.** *If a frame follows a* legal path *on the Orient topology then it will never be looped.*

Proof: If a legal path cannot cross an edge more than once in the same direction, then a frame following a legal path will never be looped, and we are done. Assume by contradiction that there is a legal path that moves along an edge in the same direction twice. Then the path must move along edges of the active topology that form a cycle in the communication graph G. Denote by c this sequence of nodes that creates a cycle on G . Since the graph D of the orient topology is a DAG, the sequence of nodes c does not form a direct cycle on D. Therefore it can be observed that c must contain at least one node v with two outgoing edges in D, to the previous and next nodes in the sequence (and at least one node with two

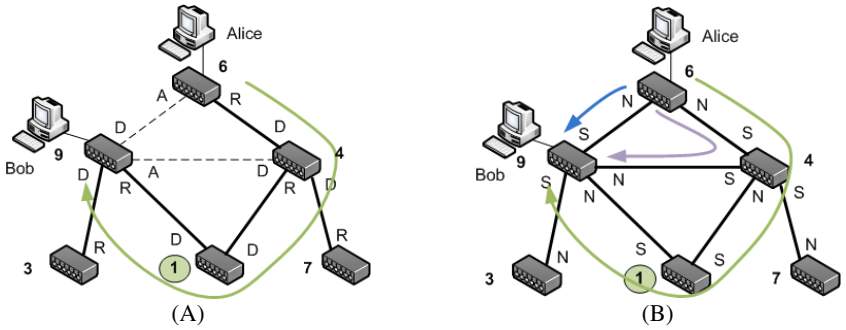

Fig. 2. Forwarding frames from Alice to Bob: (A) Spanning Tree forwarding - unique legal path. (B) Orient forwarding - 3 legal paths.

incoming edges). But then, both ports on v are north oriented and moving between these two ports is illegal, which makes the path illegal. Contradiction. ■

We now state without a proof a third property with significant practical implications for implementing the Orient architecture. An Orient bridge is a bridge that operates according to the Orient protocol.

Property 3. *Orient bridges can be added incrementally to an operating RSTP bridged network while maintaining the Connectivity and Loop-free properties of the network.*

We now prove another important property of the Orient topology, which we will soon use.

Property 4. *If each bridge in the Orient topology activates arbitrarily one of its North Ports, then the resulting topology is a spanning tree.*

Proof: A connected graph with $n-1$ edges is a tree. The number of edges in the resulting topology is $n-1$ (all but the Polaris choose one outgoing edge). Since every node is connected to the Polaris via a sequence of north ports, the resulting topology is connected, and we are done. ■

In the next section, we propose two types of forwarding mechanisms imposed by the path determined via the broadcasting of unlearned destinations.

IV. The Learning and Forwarding Process

Recall that in case the destination of a transmitted frame has not yet been learned by the learning process of a bridge, the frame is broadcast over all the active topology ports, except the port receiving the frame.

In the Orient architecture, we must provide methods for broadcasting frames in order to set up a unique legal path from source to destination for a given session. We propose the following two broadcasting options and discuss their benefits.

A. Tree per class

This broadcasting scheme is based on bridges dividing the frames among their north oriented ports according to a certain classification. This classification could be global and uniform, such as having a class for each priority tag, or could be local and vary from bridge to bridge, like classifying the frames according to the last bits of their source MAC addresses, where each bridge can have a different source type classification. Each bridge has a local table that contains a mapping between its North Ports and the possible frame classes, such that each class is mapped to exactly one North Port (but a North Port can be mapped to several classes). Recall that the North Ports are ordered according to the cost of their best path to the Polaris. Thus the assignment of ports to priority classes will be according to this order, with the highest priority assigned to the first port. Note that frames having the highest priority will be forwarded along the spanning tree would have been derived by RSTP. In the Tree per class scheme each frame class will be allowed to be transmitted and received through a single North Port on every bridge. This leads to the following broadcasting scheme.

When a frame enters a node through a port p, if the destination address of the incoming frame does not appear in the filtering table the bridge will do the following: if p is north oriented and the frame class is not in the classes of p, then the frame is discarded. Otherwise, the frame is broadcasted through all the South Ports and through the single North Port that corresponds to the frame class (excluding the port from which the frame entered).

The next Property immediately follows from Property 4

Property 5. *The Tree per class broadcasting scheme restricts traffic of each class to a single spanning tree.*

The logical classification of frames according to their VLAN tags can be nicely implemented over the Orient topology using the Tree per class forwarding scheme. Here, a bridge will map a VLAN class only to a port registered for that VLAN. This implies that each VLAN is not a full spanning tree but a subtree on the Orient topology.

The advantages of the Tree per class scheme is its ability to support QOS aspects but its limitations are that the number of classes must be at least the number of North Ports in order to have a full link utilization. Fig. 2 depict the load-balancing benefits of the Orient forwarding compared to the spanning tree architecture. (A) presents the spanning tree topology of Fig. 1 with the port roles: R, D, A which stand for Root, Designated and Alternate respectively. There is only one unique (legal) path between Alice and Bob. In (B) the Orient architecture is presented. All ports belong to the active topology and assigned N or S, north or south orientation. Here,

frames of different classes could follow 3 different legal paths to the destination.

B. Safe BFS

In the Safe BFS scheme unicast traffic is forwarded as follows. When a frame enters a node through a port p the *source* MAC address of the frame examined. If the source MAC address appears in the filtering table with a port other than p, then the frame is discarded. This means that there already exists a path to the source through another port. Otherwise, if necessary, the source MAC address is added to the filtering table with the receiving port p and the frame is legally forwarded. That is, if the destination address of the incoming frame does not appear in the filtering table then the frame is broadcasted through all the other legal ports. (Namely, if the frame is received on a North Port, it is forwarded only on the South Ports). Otherwise, if the destination MAC address appears in the filtering data table then the frame is forwarded through the associated port.

Note that the Safe BFS mechanism has two main differences from the standard BFS. The first difference is that in the standard BFS algorithm, the broadcasted message is only allowed to enter a node once, regardless of the port on which the frame was received. In our mechanism we do not have an indicator that the specific broadcasted frame has entered a bridge, we can only know whether a frame from the same source has entered the bridge. Thus, we cannot discard any frame for which the source has already reached the bridge because a source can transmit many frames, and all of them should reach their destination. The Safe BFS mechanism keeps the port on which the source was received for the first time and allows frames from that source to enter the bridge only through that port. The second difference is that Safe BFS is restricted to the Orient topology and constructs only legal paths. The correctness of the Safe BFS forwarding is established in the following statement.

Property 6. *For each source x and destination y, Safe BFS sets a unique simple communication path from y to x. Moreover, In case all the network links are equally delayed, Safe BFS will set the shortest legal path from y to x.*

Proof: By Theorem 1, there exists at least one legal path from x to any other node on the orient topology. Thus, after the broadcasting stage of Safe BFS, the MAC address of x will appear on every bridge filtering data table. Moreover, the MAC address of x will appear exactly once on each bridge, according to the first time the frame has reached the bridge. This provides a unique legal path from every bridge to the source x. Also,the corresponding ports represent the contemporary fastest legal paths to x. Therefore, if the links are equally delayed the Safe BFS broadcasting will provide the shortest path from y to x. ■

V. Simulation Results

The Orient system was evaluated for its performance via simulations and was compared to the performance of single RSTP spanning tree. The network topology was assumed to be a grid topology which can represent metro Ethernets and cluster networks. As in [13], we assumed various grid sizes to establish the robustness and scalability of the Orient architecture over increasing network sizes. We considered grid of sizes 5×5, 6×6 , . . . , 20×20 and for each grid size we performed 10 trials of our simulations. All the figures

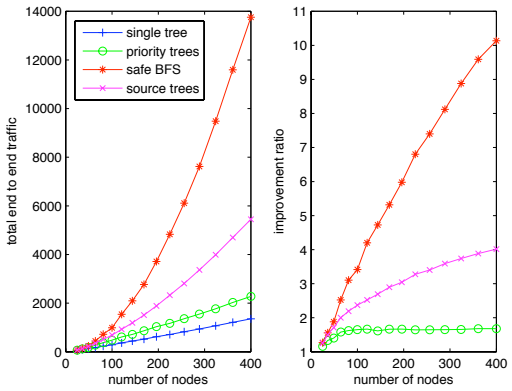

Fig. 3. Peer to peer total end to end traffic.

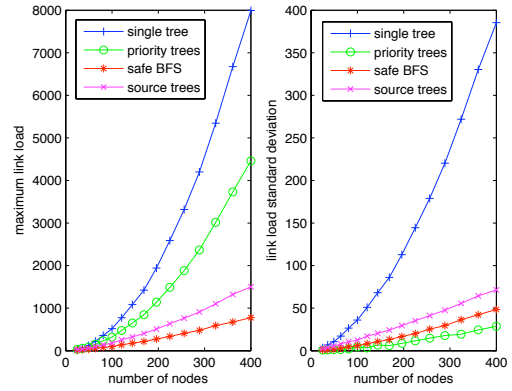

Fig. 4. Peer to peer maximum and standard deviation of link load.

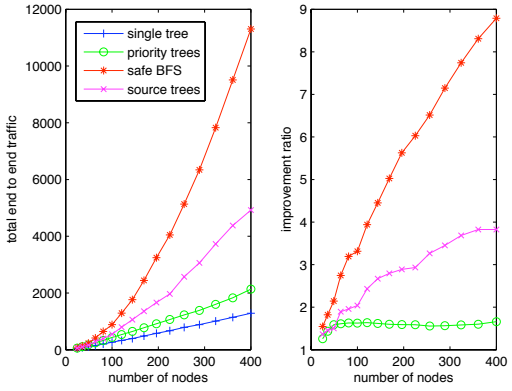

Fig. 5. Client server total end to end traffic.

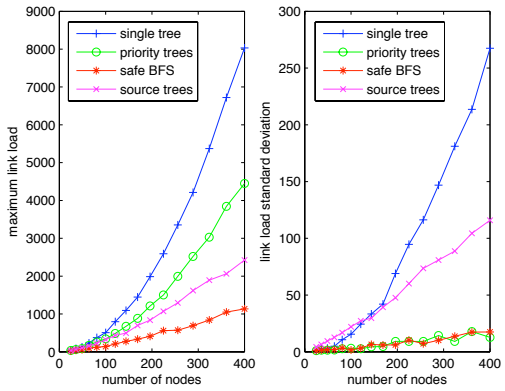

Fig. 6. Client server maximum and standard deviation of link load.

below present the average of the 10 trials where in each trial a different selection of the random source-destination pairs was performed as follows. In the peer-to-peer scenario, we assumed that each node randomly chooses 10% of the other nodes as its peers and for each one of its peers the node transmits data at 1 mbps. In the client-server scenario, we assumed that 10% of the network nodes are randomly selected as servers, and each node transmits data at 1 mbps to each server.

These scenarios were simulated to evaluate several performance issues. We measured the total network capacity (end-to-end traffic) for uniform links having limited capacity of x mbps, where x is the number of network nodes. We also considered the case of uniform links with unlimited capacity, and in this case we measured the traffic on the most loaded link (the network bottleneck), and the standard deviation of the loads among the network links, as a load-balance indicator.

As the performance of Orient depends on which broadcasting scheme is implemented to set the communication paths, we measured its performance separately for three broadcasting schemes and we present the resulting graphs. First, we considered the Safe BFS mechanism. Note that in this setting where the network is a grid, the paths set by Safe BFS on the Orient topology are shortest paths in the unrestricted network, thus its performance in this context will be at least as good as the performance of shortest path suggested architectures such as Rbridges [11]. Next, we considered the Tree per class mechanism for two types of classes. The Tree per priority sets two priority trees, one for each North Port. We assumed that the traffic priority is uniformly distributed among the two classes. The Tree per source sets one random tree for each source node.

Fig. 3 depicts the total network capacity measures, for the peer to peer scenario, We measured the total end-to-end traffic, i.e., the amount of traffic that can be transmitted over the network without overloading any individual link. The right hand side graph shows the improvement ratio against the single spanning tree results. [13] present a similar simulation scenario but for networks of up to 64 nodes. For the scenario with 49 nodes which has similar parameters our results are quite similar to those of [13] for the Safe BFS scheme, about 2.4 improvement ratio. Note that the improvement ratio for Safe BFS and source tree increases with the network size. Fig. 5 depicts the same situation for the client server scenario.

We also measured the statistics on a single link when the links are not limited in their capacity. Figures 4 and 6 depict the maximum value and standard deviation of the link load in the peer to peer and client server scenarios, respectively. For the single tree (i.e., RSTP), the most loaded links are the one close to the root, which also results in high standard deviation. All of the Orient schemes show an improvement for the bottleneck load, where the best performance is measured for Safe BFS. The standard deviation of the loads over the network links was calculated to measure the traffic load balance. Here, the Tree per priority scheme yields the smallest standard deviation (recall that we assumed a uniform priority distribution of traffic).

VI. Conclusions

We have presented Orient, a new architecture for Ethernet bridging, which can be implemented by simple modifications to the RSTP or MSTP. We have shown that Orient architecture provides a loop free topology while utilizing all network links and ports. We have proposed several broadcasting schemes in order to set paths between sources and destinations. Simulation results indicate a substantial improvement in comparison the single spanning tree topology.

References

[1] Gallo, G., Longo, G., Pallottino, S., and Nguyen, S. Directed hypergraphs and applications. *Discrete Appl. Math. 42*, 2-3 (1993), 177–201.

[2] He, X., Zhu, M., and Chu, Q. Traffic engineering for metro ethernet based on multiple spanning trees. *icniconsmcl 0* (2006), 97.

[3] IEEE. *Local and Metropolitan Area Networks: Common Specifications: Media Access Control (MAC) Bridges. ANSI/IEEE Std 802.1D-1998.*

[4] IEEE. *Local and Metropolitan Area Networks: Shortest Path Bridging. IEEE Std 802.1aq-2006.*

[5] IEEE. *Local and Metropolitan Area Networks: Virtual Bridge Local Area Networks. IEEE Std 802.1Q-2003.*

[6] Kern, A., Moldovan, I., and Cinkler, T. Scalable tree optimization for qos ethernet. *iscc 0* (2006), 578–584.

[7] Kim, C., Caesar, M., and Rexford, J. Floodless in seattle: a scalable ethernet architecture for large enterprises. *SIGCOMM Comput. Commun. Rev. 38*, 4 (2008), 3–14.

[8] Lin, S., Sharma, S., and Chiueh, T. Autonomic resource management for multiple-spanning-tree metro-ethernet networks. *nca 00* (2007), 239–248.

[9] Lui, K., Lee, W. C., and Nahrstedt, K. Star: a transparent spanning tree bridge protocol with alternate routing. *SIGCOMM Comput. Commun. Rev. 32*, 3 (2002), 33–46.

[10] Padmaraj, M., Nair, S., Marchetti, M., Chiruvolu, G., and Ali, M. Traffic engineering in enterprise ethernet with multiple spanning tree regions. *icw 00* (2005), 261–266.

[11] Perlman, R. J. Rbridges: Transparent routing. In *INFOCOM* (2004).

[12] Rodeheffer, T. L., Thekkath, C. A., and Anderson, D. C. Smartbridge: a scalable bridge architecture. In *SIGCOMM '00: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication* (New York, NY, USA, 2000), ACM, pp. 205–216.

[13] Sharma, S., Gopalan, K., Nanda, S., and Chiueh, T. Viking: a multi-spanning-tree ethernet architecture for metropolitan area and cluster networks. *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies 4* (March 2004), 2283–2294 vol.4.

[14] TOUCH, J., AND PEARLMAN, R. *Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement. IETF draft-touch-trill-prob*, 3 2009.