Evaluation of Tree-Based Routing Ethernet

G. Ibáñez, *Member, IEEE*, A. García-Martínez, J. A. Carral, J. M. Arco, and A. Azcorra, *Member, IEEE*

***Abstract*—Tree-based Routing (TRE) revisits Tree-based Routing Architecture for Irregular Networks (TRAIN)—a forwarding scheme based on a spanning tree that was extended to use some shortcut links. We propose its adaptation to Ethernet, using a new type of hierarchical Ethernet addresses and a procedure to assign them to bridges. We show that compared to RSTP, TRE offers improved throughput. The impact of transient loops in TRE is lower compared to the application of the classical shortest path routing protocols to Ethernet. Finally, TRE is self-configuring and its forwarding process is simpler and more efficient than in standard Ethernet and shortest path routing proposals.**

***Index Terms*—Routing bridges, Ethernet, spanning tree.**

I. Introduction

ETHERNET is ubiquitous in backbones and campus networks due to its excellent price and performance ratio and configuration convenience. However, the use of Spanning Tree protocols (ST) that block all links exceeding the number of nodes minus one limits its scalability and performance severely. The application of the Shortest Path routing (SP) protocols to layer-2 networks is a hot topic, although problems such as mitigating the negative effect of transient loops are difficult to solve. Notorious examples of these SP architectures are RBridges [1], which are under standardization in the TRILL Working Group of the IETF, Shortest Path Bridging [2] being developed at the IEEE 802.1 Working Group and SEATTLE [3]. These three architectures rely on the IS-IS link-state routing protocol.

Tree-based Routing Architecture for Irregular Networks (TRAIN), [4]) presents an interesting alternative to the ST and SP routing paradigms. Its proprietary switching architecture relies on a spanning tree, but enables the use of transversal branches to improve network throughput. To do so, hierarchical identifiers must be assigned to each node. Despite its interest, the application of these ideas to Ethernet has only been enabled recently by the specification of a hierarchical format for Ethernet addresses and a protocol to automatically assign these addresses to Ethernet bridges. These components were defined in the HURP protocol [5] to be used with a different forwarding scheme.

In this letter, we describe a combination of functions that enables the application of the TRAIN architecture to Ethernet. We call the resulting architecture Tree-based Routing Ethernet (TRE). On the one hand, compared to the standard RSTP (Rapid Spanning Tree Protocol, IEEE 802.1D) protocol, TRE offers much improved throughput across different realistic topologies. The throughput (computed at bottleneck links) is improved by a factor between 2 and 5 for the scale-free and random network topologies and between 1 and 2 for meshed networks with lower average node degree and more uniform degree distribution. On the other hand, TRE outperforms SP in terms of protection against transient loops and reduced complexity of the forwarding implementation, although it provides lower throughput than SP (note that the gap in throughput performances is reduced for topologies with high-degree distribution such as scale-free and random).

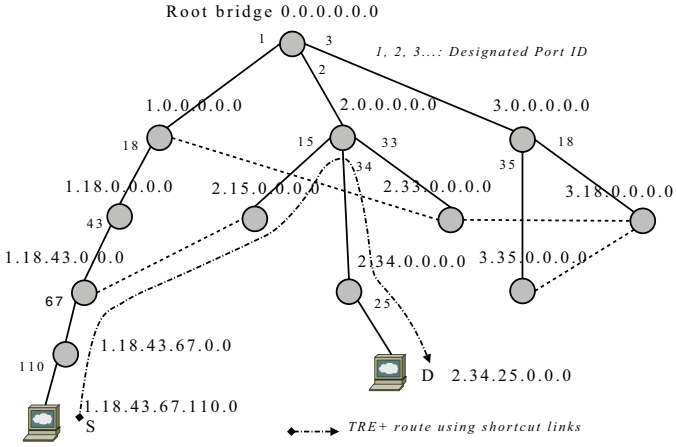

Fig. 1. TRE spanning tree and HLMAC address assignment.

II. Description of TRE Operation

TRE requires each bridge to be assigned a Hierarchical Local MAC (HLMAC) address as defined in HURP [5]. HLMAC addresses are local MAC addresses, i.e., addresses whose U/L bit is set to 1. The 46 bits available for addressing purposes (after removing the local or global bit and the multicast bit) encode by default up to 6 different hierarchical levels, with 6 bits for the first level and 8 bits for each other level. The HLMAC of a bridge is expressed in the dotted form a.b.c... as the chain of designated port IDs a, b, c, ... traversed in the descending path from the Root Bridge to the bridge to which the address is assigned.

To build the spanning tree and assign hierarchical addresses to the bridges, TRE uses a modified version of RSTP, which is defined in HURP [5]. Once the root bridge has been elected according to the RSTP standard, it is assigned HLMAC 0.0.0.0.0.0, and the process of building the spanning tree from the root to the leaves starts. An iterative process starts in which the BPDUs sent by the parent bridge include the number of

Manuscript received February 28, 2009. The associate editor coordinating the review of this letter and approving it for publication was F. Granelli.

This work was partially supported by the Spanish Ministerio de Ciencia e Innovación project grant TIN2008-06739-C04-04 (T2C2).

G. Ibáñez, J. A. Carral, and J. M. Arco are with the Universidad de Alcalá, Madrid, Spain (e-mail: guillermo.ibanez@uah.es).

A. García-Martínez is with the Universidad Carlos III de Madrid.

A. Azcorra is with the Universidad Carlos III de Madrid and IMDEA Networks.

Digital Object Identifier 10.1109/LCOMM.2009.090469

the Designated Port. The bridges receiving these BPDUs are configured with the address resulting from substituting the first 0 element of the address of the parent bridge by the port number included in the BPDU. Note that TRE does not require the exchange of additional control frames apart from those required for building the spanning tree and assigning the HLMACs. In Fig. 1, Bridge 1.18.0.0.0.0 configures its address after receiving a BPDU sent from the bridge with HLMAC 1.0.0.0.0.0 through its Designated Port number 18.

Unicast forwarding is performed on each bridge as discussed in the following paragraphs.

The bridge considers the first element in the address to discover if the destination belongs to the same branch of the spanning tree. For example, address 2.34.25.0.0.0 belongs to the same branch as bridge 2.0.0.0.0.0., because the non-zero part of one address is included in the other (2. for the second address is included in the first one). If both bridges are located in the same branch, the path determined by the spanning tree is used. The particular port through which a packet is forwarded is the root port of the bridge if the destination is less specific than the address of the forwarding bridge, and the frame must ascend. Otherwise, the destination is located below the tree, and the port number to choose the designated port is encoded in the destination address as the first non-common element between the bridge address and the destination. For example, if bridge 2.0.0.0.0.0 receives a frame directed to 2.34.25.0.0, the path is descending, since the destination is more specific than the current bridge address and the next element after removing the common part (2.), 34, identifies the number of the forwarding port to use. It is worth noting that all the nodes belonging to the same branch of the spanning tree are connected through the shortest path, because paths that are part of the shortest path are also shortest paths. Therefore, this forwarding policy leads to shortest paths.

If the destination is not in the same branch as the forwarding bridge, the use of shortcut links is considered. To make such a decision, the distance between two HLMACs is defined as follows: the common prefix of both addresses is identified—if it exists—and removed. The distance is defined as the sum of all the remaining non-zero elements of both addresses. Then, the distance between 2.15.0.0.0.0 and 2.34.25.0.0.0 is 3 because the first element in the address, 2, is equal in both addresses, and after removal, 3 non-zero elements (15, 34, and 25) remain. Note that this distance represents the number of hops a frame should perform to travel from one HLMAC to the other by ascending in the spanning tree branch until a common bridge and then descending to the destination. In the previous example, a frame from 2.15.0.0.0.0 to 2.34.25.0.0.0 must go up to 2.0.0.0.0.0 and down through 2.34.0.0.0.0 to arrive at 2.34.25.0.0.0 (3 hops).

When a bridge B receives a frame sent to a destination D that is located in a different branch, the bridge computes the distance between the next neighbor up in the tree and D. Then, it considers the addresses of its directly connected neighbors (N1, N2, etc.) not belonging to the same branch, and obtains the distance from each neighbor to the destination (from N1 to D, N2 to D, etc). If the distance to the destination through any neighbor connected through shortcuts is lower than the distance traversing the spanning tree, the shortcut route is selected. Figure 1 illustrates the operation of the forwarding algorithm, showing (with a discontinuous line) the route followed by a frame from an originating host S with HLMAC address 1.18.43.67.110.0 to a destination host D with address 2.34.25.0.0.0. At bridge 1.18.43.0.0.0, the distance to the destination through the shortcut link is computed: 3 hops from 2.15.0.0.0.0 to 2.34.25.0.0.0 and one additional hop from 1.18.43.0.0.0 to its neighbor 2.15.0.0.0.0. Since the distance computed across the spanning tree is 6 hops, the shortcut is selected.

Multicast and broadcast forwarding are performed across the spanning tree as it occurs in classical Ethernet.

III. TRE Analysis

Two characteristics must be highlighted about TRE when compared with ST protocols or with SP proposals like RBridges [1], Shortest Path Bridging [2], and SEATTLE [3]: Loop control and forwarding complexity.

TRE is loop-free in steady state (i.e. when addresses are stable). Unicast forwarding in TRE may follow the spanning tree; hence, it is loop-free, but it also uses shortcut links. To prove that the use of shortcut links is safe in terms of loops, we can reason that a shortcut is only selected when the distance to the destination is strictly shorter than the one through the spanning tree path. After each hop, performed either via the spanning tree or through a shortcut, a frame is at least one hop closer to its destination. Therefore, a frame should arrive at the destination in a finite number of hops. Since RSTP is loop-free under all circumstances, multicast and broadcast forwarding in TRE, relying on the spanning tree built by RSTP, are loop-free.

When a topology change occurs, i.e., when links or bridges fail or power up, TRE relies on the recovery mechanisms of RSTP both for reconfiguring the spanning tree and reassigning HLMAC addresses. Each bridge receiving a notification of a topology change disables the assigned HLMAC and immediately stops forwarding. Forwarding is not resumed until the spanning tree is reconfigured and the corresponding HLMACs are assigned. Therefore, there cannot be transient loops due to inconsistent decisions through shortcut links, and the impact of the unavailability of TRE is equivalent to RSTP. It is worth noting that in link state based architectures bridges may make forwarding decisions that are temporarily inconsistent; hence, they require either a TTL-like field [1] or some kind of complex synchronization to control loops.

Regarding the complexity of the forwarding process, TRE outperforms both backward learning Ethernet and SP since it does not need table lookups, but only address comparisons. Backward learning requires a lookup in a table containing A (A >> N) elements, A being the number of active hosts in a certain period (depends on the duration of the cache in the local node) and N the number of nodes of the network. SP forwarding requires a lookup in a table containing N elements to obtain the entry that exactly matches the destination address. Conversely, TRE port selection for destinations that belong to the same branch of the spanning tree of the node considered requires only 2 logical address comparisons, which are easily implemented in the hardware and are faster than a

TABLE I
PERFORMANCE OF SCALE-FREE (BARABASI-ALBERT) TOPOLOGIES

Topology		Average path length			Throughput (% of SP)		Th. Ratio
Nod.	Deg.	SP	TRE	ST	TRE	ST	TRE/ST
64	4	2.99	3.57	4.42	38.4	18.6	2.06
	6	2.5	3.04	3.9	31.4	12.2	2.57
	8	2.22	2.74	3.63	27.1	10.0	2.71
128	4	3.52	4.47	5.66	28.5	12.9	2.21
	6	2.86	3.71	4.89	22.2	7.44	2.99
	8	2.56	3.34	4.5	19.5	5.6	3.48
256	4	4.02	5.23	6.28	18.5	9.02	2.05
	6	3.26	4.36	5.36	12.9	5.13	2.51
	8	2.89	3.87	4.85	12.4	4.38	2.82

TABLE II
PERFORMANCE OF RANDOM (WAXMAN) TOPOLOGIES

Topology		Average path length			Throughput (% of SP)		Th. Ratio
Nod.	Deg.	SP	TRE	ST	TRE	ST	TRE/ST
64	4	2.99	3.57	4.42	38.4	18.6	2.06
	6	2.5	3.04	3.9	31.4	12.2	2.57
	8	2.22	2.74	3.63	27.1	10.0	2.71
128	4	3.52	4.47	5.66	28.5	12.9	2.21
	6	2.86	3.71	4.89	22.2	7.44	2.99
	8	2.56	3.34	4.5	19.5	5.6	3.48
256	4	4.02	5.23	6.28	18.5	9.02	2.05
	6	3.26	4.36	5.36	12.9	5.13	2.51
	8	2.89	3.87	4.85	12.4	4.38	2.82

table lookup. Port selection requires d comparisons for the rest of the destinations, d being the number of neighbors of the forwarding bridge belonging to different branches. Moreover TRE, unlike transparent bridging, does not require MAC address learning. Time and message complexity of TRE spanning tree computation and address assignment are equivalent to RSTP because RSTP messages are just extended to assign and reconfigure HLMAC addresses based on existing RSTP protocol components.

IV. Performance Evaluation

The network throughput and path length, in number of hops, are computed for ST, SP, and TRE in different topologies. For throughput estimation, it is assumed that every node establishes a flow with N-1 clients, each one located at every other node. Then, the bottleneck link, i.e., the link shared by the highest number of flows in the topology, is determined. The relative throughput is computed by dividing the number of flows with the considered protocol at the bottleneck link by the number of flows obtained for SP. Note that SP may not be optimal, since obtaining the maximum throughput requires solving a load distribution optimization problem.

Scale-free (Barabasi-Albert model following power-law distribution) and random (Waxman) topologies were generated with BRITE [6], with varying average node degree (4, 6 and 8). To remove the dependency on the particular root node elected, N iterations per topology, with a different root bridge elected on each iteration, are performed to obtain an average. Table 1 and Table 2 show the results for scale-free and random topologies respectively.

The relative throughput ratio of TRE versus ST (right column value) increases with higher average node degrees and shows low dependency on network size. However, the improvement with the average node degree does not keep pace with SP, because of TRE limitations: only shortcuts to the branch in which the destination is located can be used, and each bridge may choose a shortcut even though a better shortcut could be selected later in the spanning tree.

TABLE III
PERFORMANCE OF REFERENCE TOPOLOGIES

Topology			Average path length			Throughput (% of SP)		Th. Ratio
Name	Nod.	Deg.	SP	TRE	ST	TRE	ST	TRE/ST
Enterprise	34	3.18	3.26	3.45	3.92	79	71	1.1
Pan-Euro	36	3.11	3.62	4.3	5.24	55	41	1.3
EBONE	87	3.7	4.53	5.26	6.39	52	38	1.4
Tiscali	161	4.07	4.2	4.7	5.77	54	36	1.5
Sprint	315	6.17	3.97	4.76	6.14	44	23	1.9

The TRE/ST ratio is slightly lower in random topologies because in power-law distributions, there are nodes with high node degree acting like "hubs" whose links are very likely to be selected as shortcuts by TRE.

The results obtained for a set of topologies used as reference for real networks are shown in Table 3. We use the enterprise campus [7] model to represent usual campus networks. This topology physically mimics a tree; hence, any kind of shortest path computation provides limited advantage over ST operation. The rest of the topologies are close to flat meshes: Pan-European reference network described in [8] and three real topologies obtained with Rocketfuel [9], ranging from a small access provider network consisting of 87 routers to a network with 315 routers. The ratio of average throughput of TRE is better in scale-free and random networks than in flat networks. This is because the links in flat networks are connected to close nodes, which are also close among them, reducing the number of destinations to which these shortcuts are useful.

We conclude that the improvement of TRE over ST both in terms of throughput and path length is high in random networks but moderate –though significant– in meshes of lower degree variation. The improvement increases with the average node degree due to the higher number of shortcuts. Although SP offers better performance, TRE is a good alternative due to its lower complexity and its loop-free forwarding mechanism.

References

[1] R. Perlman, “Rbridges: transparent routing," in *Proc. IEEE Infocom 2004*, Mar. 2004.
[2] IEEE 802.1 Working Group 802.1aq, Shortest Path Bridging. [Online]. Available: http://www.ieee802.org/1/pages/802.1aq.html
[3] C. Kim, M. Caesar and J. Rexford, “Floodless in seattle: a scalable ethernet architecture for large enterprises," in *Proc. ACM SIGCOMM 2008*, Aug. 2008.
[4] H. Chi and C. Tang, “A deadlock-free routing scheme for interconnection networks with irregular topologies," in *International Conference on Parallel and Distributed Systems*, 1997.
[5] G. Ibáñez, A. García-Martínez, A. Azcorra, J. A. Carral, and J. M. Arco, “Hierarchical up/down routing architecture for Ethernet backbones and campus networks," *High Speed Networks Workshop (HSN 2008), INFOCOM 2008*, Apr. 2008.
[6] “Boston University Representative Topology Generator - BRITE." [Online]. Available: http://www.cs.bu.edu/brite/.
[7] CISCO (1999), “Gigabit campus Ethernet—principles and applications." [Online]. Available: http://www.cisco.com/warp/public/cc/so/neso/lnso/cpso/gcnd_wp.pdf.
[8] UGent-IBCN, “NRS - Reference Networks." [Online]. Available: http://www.ibcn.intec.ugent.be/INTERNAL/NRS/index.html.
[9] N. Spring, R. Mahajan, and D. Wetherall, “Measuring ISP topologies with Rocketfuel," in *Proc. ACM SIGCOMM*, Aug. 2002.