TRE+: Extended Tree-Based Routing Ethernet

Juan A. Carral, Guillermo Ibáñez, Alberto García-Martínez,
Miguel A. Lopez-Carmona, and Ivan Marsa-Maestre

Tree-based routing Ethernet (TRE) is a recent Ethernet architecture that enables shortcut links to improve performance compared to spanning tree protocols. However, TRE can only use shortcuts that arrive directly at bridges located in the branch of the destination. TRE+ extends the topology knowledge of a bridge to 2 hops away, thus unveiling new shortcuts to the destination branch. Simulations show a major performance improvement of TRE+ compared to TRE, with results close to shortest paths in some topologies.

Keywords: Routing bridges, Ethernet, spanning tree, hierarchical addresses.

I. Introduction

Ethernet performance is limited by the use of spanning tree (ST) protocols, which block all links exceeding the number of bridges minus one. To overcome this restriction, proposals like RBridges [1] use shortest path (SP) routing protocols like IS-IS in layer two. However, these SP protocols are complex because they need full topology knowledge to perform the shortest path computations using the Dijkstra algorithm.

An interesting alternative to the ST and SP routing paradigms is the tree-based routing architecture for irregular networks (TRAIN) [2]. Its switching architecture complements an ST topology with shortcut links to improve throughput. Hierarchical tree-based addresses assigned to each bridge are used to control the forwarding of frames while taking advantage of available shortcuts. Tree-based routing Ethernet (TRE) [3] restates TRAIN ideas for Ethernet by defining a hierarchical local MAC (HLMAC) addressing scheme and an automatic address assignment mechanism that can be implemented as a rapid spanning tree protocol (RSTP) extension. To transport frames using regular (non-hierarchical) addresses, a learning mechanism in the edge bridges of the network as well as tunneling or MAC-to-HLMAC address translation techniques are required as with RBridges [1] or HURBA [4]. TRE provides higher throughput than RSTP and simpler forwarding but is limited to use links that are directly connected to the branch in which the destination is located.

In this letter, we extend TRE to provide bridge topology information up to 2 hops away to increase the number of alternative paths (shortcuts), thus yielding a performance improvement. To do so, a TRE+ bridge must inform all its neighbors of the neighbors to which it is directly connected. This distance vector-like routing exchange is only propagated to neighbors that are 1-hop away to prevent transient loops resulting from count-to-infinity situations. Evaluations show that the throughput is improved by more than 60% for scale-free networks and by more than 200% for random networks.

II. TRE+ Operation

TRE+ is an extension of TRE aimed to improve the ability of TRE bridges to detect and take full advantage of possible shortcuts to the destination of a given frame. The operation of both TRE and TRE+ requires the following: a spanning tree active forwarding topology, provided by a standard protocol such as RSTP; a hierarchical tree-based addressing scheme, used to control the forwarding decisions; an automatic mechanism to assign a hierarchical address to each bridge, (RSTP can be extended to support the address assignment as described in [4]); and a new set of forwarding rules based on the hierarchical addressing scheme. Furthermore, TRE+

Manuscript received Sept. 15, 2009; revised Oct. 29, 2009; accepted Nov. 12, 2009.
This work was supported by Comunidad de Castilla-La Mancha, Spain (PII1I09-0204-4319, EMARECE).
Juan A. Carral (phone: +34 91 8856625, email: juanantonio.carral@uah.es), Guillermo Ibáñez (email: guillermo.ibanez@uah.es), Miguel A. Lopez-Carmona (email: miguelangel.lopez@uah.es), and Ivan Marsa-Maestre (email: ivan.marsa@uah.es) are with the Department of Computer Engineering, University of Alcala, Madrid, Spain.
Alberto García-Martínez (email: alberto@it.uc3m.es) is with the Department of Telematic Engineering, University Carlos III of Madrid, Madrid, Spain.
doi:10.4218/etrij.10.0209.0388

bridges require a new mechanism to detect neighboring bridges located within 2 hops and to learn their HLMAC addresses.

The procedure to discover the bridges located in the 2-hop neighborhood is similar to a distance vector reachability exchange, although in this case, the scope for the advertisements is limited to 1 hop. In other words, each bridge periodically advertises the HLMAC addresses of the bridges to which it is directly connected to all its neighbors with distance 1. When a bridge loses communication with a directly connected neighbor, it immediately advertises distance 2, the infinite distance in our setup.

Each TRE+ bridge is assigned an HLMAC address as in TRE [3]. This address is the sequence of the designated port numbers (coded in 8 bits) traversed from the root bridge to the addressed node descending the ST, expressed in the dotted form a.b.c.d.0.0. In Fig. 1, the HLMAC 8.9.0.0.0.0. (shortened to 8.9.) is assigned to the bridge receiving a BPDU with address assignment information from the designated (parent) bridge, 8., which sent the BPDU through its designated port, 9.

At each bridge, an incoming frame is forwarded through the ST unless the distance through a shortcut is strictly lower. Therefore, the bridge must first compute the distance to the destination through the ST and then via possible shortcuts and choose the port leading to the better route to the destination.

The hop distance between two nodes through the ST is computed by removing the common prefix of both HLMAC addresses and counting the remaining non-zero elements. For example, the distance between bridge 8.6. and bridge 8.9.1. is 3 hops because 8. leads both addresses, and after removal, 3 non-zero elements (6, 9, and 1) remain.

To compute the distance to a given destination through a shortcut offered by a neighbor bridge N (in its 2-hop neighborhood), a bridge must first compute the distance from N to the destination (using the ST) and then add its own distance to N (either 1 or 2 hops).

The forwarding decision works as follows:

```
If destination HLMAC D is_a_prefix_of current bridge HLMAC C
  Then forwarding_port = root_port  # Follow tree upwards
  Else If C is_a_prefix_of D        # Follow tree branch of D downwards
      Then forwarding_port = first_remaining_port (D-C)
      Else
          Set forwarding_port = root_port   # Set default forwarding via tree
          currentDist = tree_distance (C,D)
          For each N in '2-hop neighbor set of C'
              dist = tree_distance (N,D) + distance (C,N)
              If dist is lower than currentDist
                  Then Set forwarding_port = port_leadingTo(N)
                       Set currentDist = dist
```

An example of the forwarding process for TRE+ is presented in Fig. 1. Node S sends a frame to node D. Node S forwards the frame up the ST to 1.7. via its root port. The distance from bridge 1.7. to D using the ST is computed as 6 hops (no common prefix, 2+4 non-zero elements). Next, bridge 1.7. checks the available shortcuts. Bridges 14., 8., and 8.6. are in its 2-hop neighborhood

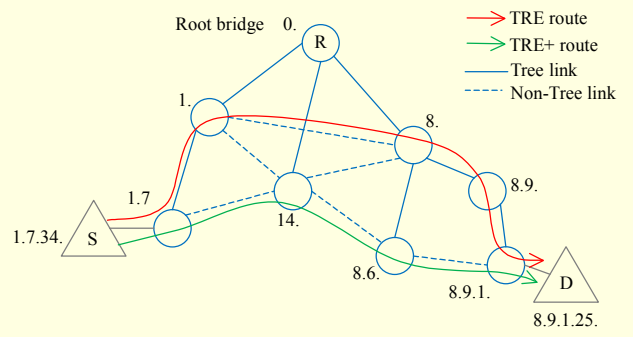

Fig. 1. HLMAC address assignment and forwarding path from S to D using TRE and TRE+.

and do not belong to its tree branch. The minimum distance to D, as computed by bridge 1.7., results from using bridge 8. The distance predicted to D is 5 hops, obtained by adding the distance from 1.7. to 8. (2 hops), plus the distance from 8. to D across the ST (3 hops). The port leading to bridge 14. (and ultimately to 8.) is chosen; however, when the frame arrives at bridge 14., it unveils a new shortcut to D via its 2-hop neighbor bridge 8.9.1., which further reduces the number of hops by 1.

TRE+ is loop-free when addresses are stable. The path defined by a shortcut is loop-free in steady state because it results from a simple distance vector exchange. The combination of ST subpaths and shortcuts is also loop-free because a shortcut is only selected when the distance to the destination is strictly shorter than the one through the ST path. After each hop, either performed through the ST or through a shortcut, the distance to the destination is lower than the one estimated in the previous bridge. Therefore, a frame should arrive at the destination in a finite number of hops.

To analyze the behavior when a topology change occurs, we separately consider failures affecting shortcuts and failures affecting the ST. Regarding shortcuts, in the conditions defined, shortcuts never induce transient loops. Circular loops, which are loops involving 3 or more bridges, can never occur when the maximum hop distance of a bridge to be considered is restricted to 2. Therefore, transient loops could only occur if two bridges incorrectly assume that the other one is directly connected to the destination. Consider bridges B1 and B2, directly connected and also connected to D. This situation occurs if both B1 and B2 send an advertisement for D (distance 1) just before D fails (or both links B1-D and B2-D fail). B1 immediately sends an advertisement to B2 with distance 2 for D to indicate that B2 must withdraw, but then it receives the (now outdated) advertisement from B1 announcing distance 1 to D. Although this advertisement will be followed by a withdraw advertisement, a short period may occur with B1 and B2 exchanging traffic. To prevent this situation, the withdraw advertisement sent by bridge B1 for neighbor D to bridge B2

Table 1. Summary of Barabasi-Albert model networks results.

	Degree=4			Degree=6			Degree=8		
Nodes	64	128	256	64	128	256	64	128	256
Average path length									
ST	3.8	4.6	5.2	3.6	4.1	4.5	3.3	3.8	4.3
TRE	3.1	3.5	4.2	2.8	3.2	3.5	2.5	2.9	3.3
TRE+	2.8	3.2	3.7	2.4	2.8	3.1	2.2	2.5	2.8
SP	2.8	3.1	3.5	2.4	2.7	3.0	2.2	2.4	2.7
Throughput relative to shortest path (%)									
ST	255.0	19.2	15.4	13.6	10.1	9.2	10.2	7.4	5.6
TRE	58.4	46.9	36.5	41.0	32.3	29.8	36.2	28.7	22.5
TRE+	91.9	89.5	75.5	96.2	84.6	50.9	97.2	91.1	82.7
Throughput ratio									
TRE+/ST	91.9	89.5	75.5	96.2	84.6	80.9	97.2	91.1	82.7
TRE+/TRE	1.6	1.9	2.1	2.3	2.6	2.7	2.7	3.2	3.7

Table 2. Summary of Waxman model networks results

	Degree=4			Degree=6			Degree=8		
Nodes	64	128	256	64	128	256	64	128	256
Average path length									
ST	4.4	5.4	6.3	3.9	4.6	5.4	3.6	4.2	4.9
TRE	3.6	4.4	5.2	3.0	3.7	4.4	2.7	3.3	3.9
TRE+	3.2	3.9	4.6	2.6	3.1	3.7	2.3	2.7	3.2
SP	3.0	3.5	4.0	2.5	2.9	3.3	2.2	2.6	2.9
Throughput relative to shortest path (%)									
ST	18.6	13.2	9.0	12.2	8.1	5.1	10.0	6.3	4.4
TRE	38.4	27.1	21.4	31.4	20.8	12.2	27.1	18.1	11.6
TRE+	76.2	60.8	43.8	86.0	65.5	44.8	92.0	70.0	55.9
Throughput ratio									
TRE+/ST	4.1	4.6	4.9	7.0	8.1	8.7	9.2	11.1	12.8
TRE+/TRE	2.0	2.2	2.1	2.7	3.1	3.7	3.4	3.9	4.8

includes a nonce (unique random number) and forces bridge B2 to send a new advertisement including the nonce if B2 still has a direct connection to D. B1 does not accept reachability advertisements for D from B2 until the nonce is returned; thus, B2 is not considered a valid shortcut to D.

When a failure affects the ST, TRE+ relies on RSTP messages to reconfigure the ST and reassign HLMAC addresses. Each bridge receiving such a message immediately stops forwarding and clears all acquired neighbor HLMACs until RSTP reconfigures the tree and assigns new HLMAC addresses. The 2-hop neighborhood information is then rebuilt.

TRE+ has low computational complexity compared to SP since the number of address comparisons needed to forward a frame is in the order of d^2, d being the average node degree. Note that ST bridges require a lookup in a table of A elements, A being the number of active stations, and SP requires a lookup in a table of N elements, N being the number of nodes.

III. Performance Evaluation

We show now the throughput and path length results obtained via flow level simulations across different network types, sizes, and topologies. We assume a fluid model where flows are transmitted at a fixed rate. Each bridge establishes a session (flow) with every other bridge. Each flow is routed along the shortest path allowed by the protocol under study (RSTP, TRE, TRE+, and an ideal SP). Next we determine the bottleneck link, which is the link shared by the higher number of flows. When the traffic per flow is increased, the bottleneck link is the first to reach its link capacity. Assuming a constant rate per flow, the relative throughput is computed by dividing the number of flows at the bottleneck link by the number of flows obtained for SP.

Scale-free (Barabasi-Albert model following power-law distribution) and random (Waxman model) topologies were generated using BRITE [5], varying both the network size (64, 128, and 256 bridges) and the average node degree (4, 6, and 8). Forty different topologies were evaluated for each combination. To remove the dependency on the particular root bridge elected, M iterations per topology (M being the number of bridges with degree greater than or equal to the average node degree) were performed, electing a different root bridge each time. Results are shown in Tables 1 and 2.

TRE+ increases the throughput of TRE 2.0 to 4.8 times for Waxman topologies and 1.6 to 3.7 times for Barabasi topologies. The benefit increases with the number of bridges and with the average node degree of the network. Average path lengths are shortened by 10% to 15%, approaching the shortest path length.

References

[1] R. Perlman, "Rbridges: Transparent Routing," *Proc. IEEE Infocom*, Mar. 2004.

[2] H. Chi and C. Tang: "A Deadlock-Free Routing Scheme for Interconnection Networks with Irregular Topologies," *Proc. ICPADS*, 1997, pp. 88-95.

[3] G. Ibáñez et al. "Evaluation of Tree-Based Routing Ethernet," *IEEE Commun. Lett.*, vol. 13, no. 6, June 2009, pp. 444-446.

[4] G. Ibáñez et al., HURP/HURBA: Hierarchical Up/Down Routing Architecture for Ethernet Backbones and Campus Networks, Computer Networks, *Computer Networks*, vol. 54, no. 1, Jan. 2010, pp. 41-56.

[5] Boston University Representative Topology Generator - BRITE, available at http://www.cs.bu.edu/brite/.