ProtocolTree ::ProtocolTree\_EGH(ProtocolTree)

Note: before starting to change the KW tree, save a copy of it for the reachability!!!!!

A Static method of Protocol Tree class:

1. Go over all nodes and For each node: for each third child, copy the subtree, starting from it's grandparent until the depth of the new tree (in the current path)less than or equals to N = n\*((1/3)/epsilon)- **in EGH always 1/3 –epsilon!!**
2. For each leaf: if the leaf's depth is less than N, zero padding.

SubRoutines:

If (father hight + grandparent\_hight) <= N)

If(parent.get\_parent() == NULL) -> add all tree (from the root).

* Set error child (node parent):

Copy the parent.get parent() subtree to parent.child[2], using preorder with the grandparent as the root.

* Zero\_padding:

Go over all leaves: for each leaf: if the depth is less than N- add subtree of zeros with depth = 2n – leaf.depth.(add and set a field which indicates if the node is a zero padding node). This subtree will contain nodes in alternating order (A-B,A-B…)

Return the resilient ProtocolTree.

ProtocolTree::ProtocolTree\_REVERSE\_KW(ProtocoTree(kw),Resillient\_protocol\_tree(EGH),error\_fraction,flags):

Protocolt\_tree Reachability (ProtocoTree(kw),Resillient\_protocol\_tree(EGH)):

1. Create an array that contains all P' leaves.
2. Create a new protocol\_tree (root).
3. For each leaf: if (is\_reachable (leaf)) -> add all path from the root to this leaf (in EGH protocol\_tree) to the new protocol\_tree.

Bool is\_reachable (ref ProtocoTree(kw),ref Resillient\_protocol\_tree(EGH),ref leaf)

1. Given a leaf, generate all legal error vectors which lead to this leaf in P'.

* For each error vector, check if there's exist a path which ends with a leaf in P.

Legal error vector:

* Doesn’t contain more than |num of leaves|\*(error\_fraction) symbols different from '\*'.
* Check if there exists a path in P:

Go over P, using the given error vector. At the end, check if the current node is a leaf (if\_is\_leaf).

If (is\_leaf)

Return true;

Else

False;

Return if\_reachable.