Distributed implementation of Baruva's Algo

1. Identify components.

Leader election: Selecting one leader out of several peers

Using leader election one can name components with label $i$ or lower component id.
2. Selection of minimum cross-edge of a component

- can also be done using leader election (with the component) or based on weights of cross edges.
Multicast Tree Construction Algo for Internet:

1. Reverse Path Forwarding
   1. Use flooding to build Spanning Tree
   2. Pruning to remove unnecessary parts of the tree.

2. Core-Based Tree

   $C_g$: Core node for a multicast group $G$.
   - Each member node sends a JOIN message to $C_g$.
   - The JOIN message connects the member node to the existing tree via shortest path.
Approximation Algorithm for Steiner Tree

S: set of nodes to be covered by the Steiner Tree

|S| = K

T: Steiner tree

|T| ← cost of steiner tree

\[
|T_2| \leq 2|T| \left( \frac{(K-1)}{K} \right)
\]

Distance network

Distance network of S is a graph G' in which each pair of nodes in S is connected by a link whose weight is the length of shortest path between u and v in the original network graph G.
Construction of $T_2$:

1. $T_1: \text{MST}(G')$

2. $T_1 \rightarrow T_2$ in $G$ by substituting each edge $(u,v)$ in $T_1$ by shortest path between $u$ & $v$ in $G$

$|T_2| \leq |T_1|$

$|W| = 2|T|$

from $W$ after chopping it in subpath we get $P$
P is a set of k subpaths of W (after the chopping is done).

|P| = |W|

Remove the highest weight subpath from P to get P'

|P'| ≤ (k-1)|P| / k

Replace each subpath in P' by shortest path between the corresponding nodes.

Replace these shortest path by correspondingly link in the distance network to obtain T e is a spanning tree in the distance network G'

|T_1| ≤ |T_0| ≤ |T'| = |P'| ≤ \frac{k-1}{k} |P| = \frac{(k-1)^2}{K} |T_1|
Summary & Construction

1. \( G' \): distance network (graph) of \( S \)
2. Constructed MST in \( G' \) (\( T_i \))
3. We used \( T_i \) to construct a multicast tree in \( G \), called \( T_2 \), and showed that the cost of \( T_2 \) is bounded by cost of \( T_i \)
4. Obtained another spanning tree in \( G' \), called \( T' \). Since \( T_1 \) is a minimum spanning tree in \( G' \), we have \( \text{cost}(T_i) \leq \text{cost}(T') \)

\( T' \) was obtained from \( T \) in such a way that...
\[ |T'| \leq 2|T| \]