Flooding-Based Asynchronous Spanning Tree Construction

The messages used in this algorithm, called algorithm *Flood_ST*, are *probe*, *ack*, and *reject*. Any node that wants to build a spanning tree starts the algorithm by sending the *probe* message to its neighbors, which is transferred to other nodes. Since a node may receive more than one *probe* message, acknowledgement (*ack*) and negative acknowledgement (*reject*) messages are needed to check whether a node has received *probe* before as shown in Algorithm 1.

Algorithm 1. *Flood_ST*

```
1:  int parent ← ⊥
2:  set of int childs ← ∅, others ← ∅
3:  message types probe, ack, reject
4:
5:  if i = root then                                   ▷ root initiates tree construction
6:      send probe to Γ(i)
7:      parent ← i
8:  end if
9:
10: while (childs ∪ others) ≠ (Γ(i)\{parent}) do
11:     receive msg(j)
12:     case msg(j).type of
13:             probe:  if parent = ⊥ then             ▷ probe received first time
14:                         parent ← j
15:                         send ack to j
16:                         send probe to Γ(i)\{j}
17:                     else                           ▷ probe received before
18:                         send reject to j
19:             ack:    childs ← childs ∪ {j}          ▷ include j in children
20:             reject: others ← others ∪ {j}          ▷ include j in unrelated neighbors
21: end while
```

The *root* starts the algorithm, and whenever node *i* receives a *probe* message, it marks the sender *j* as its parent and sends an *ack* message to *j*. The parent *j*, in receipt of an *ack*, marks the child as one of its children. Then, node *i* sends *probe* message to all of its neighbors except the parent *j* consequently. If a node already has a parent when it receives a *probe* message from a neighbor node, it sends a *reject* message to the neighbor. The termination condition is when the union of the children (*childs*) and unrelated neighbors (*others*) of a node *i* equals its neighbors except the parent, as checked in line 10 of the algorithm. It should be noted that the main body of the algorithm between lines 10–21 is also executed by the *root*.

Figure 1 shows an example spanning tree formed by the *Flood_ST* algorithm over a network consisting of six nodes *a, b, c, d, e*, and *f*. Node *a* is the root and starts construction of the tree by sending *probe* messages to its neighbors *b, e*, and *f*. Each message is labeled with the time frame number such that messages with the same label occur concurrently within the same time frame. It can be seen that due to the delay in the link between *a* and *e*, the probe message from *b* reaches *e* before, and *b* becomes the parent of the node *e* although it is now two hops away from the root. The final constructed spanning tree is shown by the bold lines.

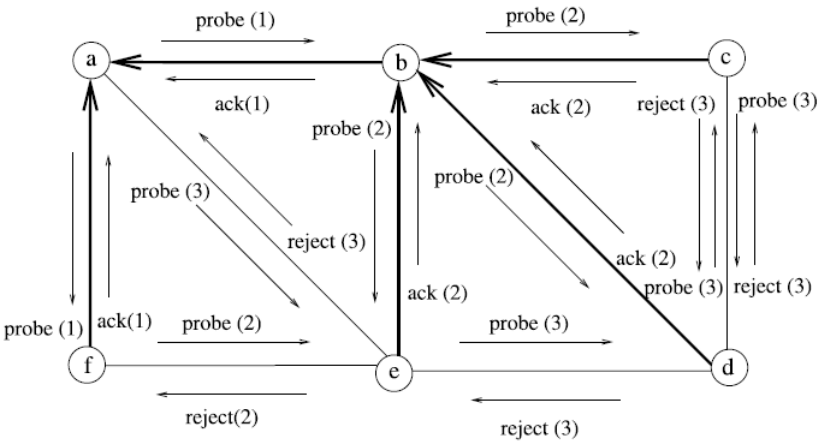

Figure 1 An example spanning tree formed by *Flood_ST* algorithm