Load balancing based on star decomposition

Real values case

Variables at node u

- Load
- State = Out, Leaf, Center
- AvgGap // computed by AvgGap()
  = \frac{\sum_{$v \in N[u]$} $v.Load$ }{|N[u]|}
- P

Events at node u (ordered set of rules)

- Incoherent local gap
  - If u.AvgGap != AvgGap() then u.AvgGap := AvgGap();
- Become center
  - If u.State = Out and u.AvgGap > ε and \forall v \in N(u): u.AvgGap ≥ v.AvgGap then u.State = Center
- Incoherent center
  - If u.State = Center and u.AvgGap $\leq \varepsilon$ or \exists v \in N(u) : v.AvgGap > u.AvgGap then u.State = Out
- Become leaf
  - If u.State = Out and \exists v \in N(u) : v.State = Center then u.State = Leaf; u.P = random({v \in N(u) : v.State = Center});
- Incoherent leaf
  - If u.State = Leaf and u.P.State != Center then u.P = null; u.State = Out;
- Perform local equilibrium
  - If u.State = Center and \exists v \in N(u) : v.State = Leaf then Wait(t (ms)) // collect new leaves

    In transaction mode and compute new load value for u and its leave

    u.State = Out