How can we do this simulation? Diminished Chord [KM04]

**Diminished Chord**

- Assume a base Chord network C
  - Join of nodes in C
  - Leave of nodes in C
  - Routing of keyed messages in C
- For each sub-network (subgroup X), emulate DHT operations efficiently
  - Join of nodes in X
  - Leave of nodes from X
  - Routing of keyed messages in X

**Diminished Chord: Join**

**Diminished Chord: Lookup**

- K=17
- Lookup: Find first green node after itself

**Tree-based solution**

- Lookup: Find first green node after itself
Tree-based solution: Lookup

- Look up: Find first green node after itself

Tree-based solution: State

- State: O(log n) per green node
- Reduce state: One entry per green node

Embedding the tree in Chord

- Place the interior nodes such that edges already exist in Chord network

Sparger rings with pre-fingers

- Tree-node hosted by preceding host
- Right child edges trivial to follow
- Left child edges use pre-fingers + successor; O(log n)

Finger(b)=succ( a + 2)^
Pre-finger(b)=pred( a + \frac{1}{2})

More details in paper

Lookups without pre-fingers

Routing load balance
- If there are many groups, and each group generates similar lookup load, then all nodes have similar load
- What are the load-balancing properties when groups have dissimilar sizes, lookup loads, ...?

Drawbacks – Loss of isolation

A non-member works (routes, lookups) for a group

- Implication: a non-member node with high load may affect group with unloadding members
- Implication: owner preferences violated as a non-member assists in group services
- Lookups, Successor, Insert, Leave now cost O(log n) even if groups are small in size O(k) [web caching?]