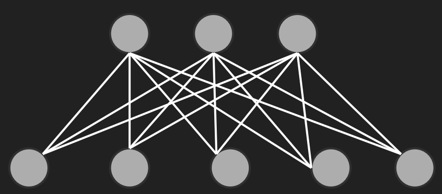
An Algorithm for Computing Dynamic Flooding Topologies

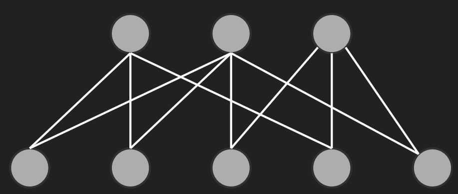
draft-chen-lsr-dynamic-flooding-algorithm-00

Dynamic Flooding

- Decouple flooding topology (FT) from physical topology



Physical topology



Flooding topology

  - Data traffic is forwarded on the physical topology
  - Link-state protocol data units are flooded on the flooding topology

Flooding Topology Requirements

- Include all systems in the area.
- Bi-connected if allowed. Avoid single point of failure.
- Bounded diameter.
  - Sparser FT => Larger diameter => longer propagation delay
- Balanced node degrees.
  - Node degree indicates the flooding burden on a node.

An Algorithm for Computing FT

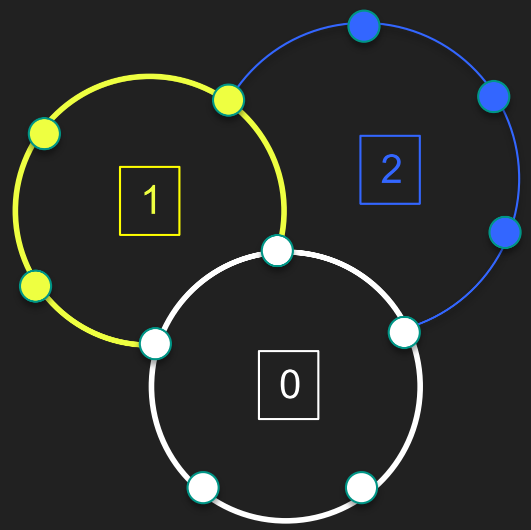
- Not intended to find the optimal solution.
- Not intended to standardize this algorithm.
- A pragmatic approach.
- Welcome future research and refinement to improve it.

Problem Statement

- Model the physical topology as a graph G(V,E).
  - V is the set of all reachable nodes in this area
  - E is the set of all edges, each connecting two nodes that advertise each other as neighbors.
- Find a sub-graph composing of overlapping cycles to cover all nodes.
  - Overlapping cycles to provide bi-connectivity.
  - Diameter of the sub-graph and node degree are considered.

Algorithm Outline

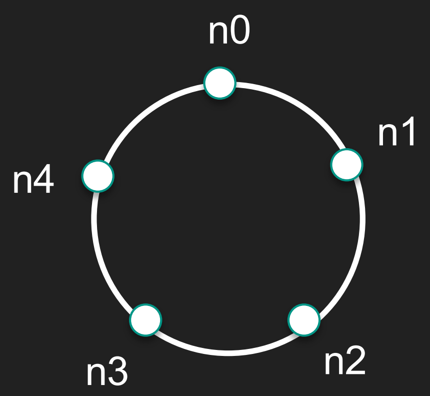
1. Start with an empty sub-graph G'({}, {}).
2. Find V(0) and E(0) to form an initial cycle. G'(V(0), E(0))
3. Find V(i) and E(i) to form an arc path.
    - V(i) and E(i) are in G but not G'
    - Two endpoints of this arc path are in G'
    - Add the arc path to the sub-graph G'(V(0)+V(1)+…V(i), E(0)+E(1)+…+E(i))
4. Repeat Step 3 until all nodes are included in the sub-graph G'.



0: initial cycle V(0) E(0)
1: arc path V(1) E(1)
2: arc path V(2) E(2)

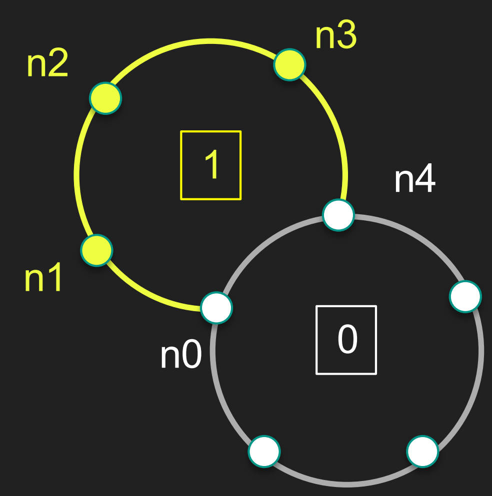
Algorithm Details

- Initial cycle
  - Select a starting node n0.
  - Depth-First search for a limited number of steps or until finding a node whose neighbors are all visited: n0 – n1 – n2 – n3.
  - Breadth-First search for a shortest path back to the starting node n3 – n4 – n0.



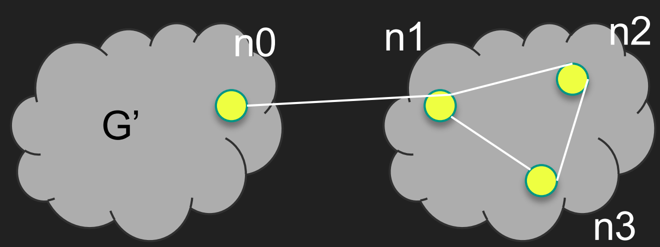
Algorithm Details

- Arc path selection
  - Select a starting node n0 on the current sub-graph G'.
  - DFS for a limited number of steps: n0 – n1 – n2.
  - BFS for a shortest path from n2 to any node on G' | n0: n2 – n3 – n4.
  - Add path n0 – n1 – n2 – n3 – n4 to sub-graph G'.
- Endpoint selection:
  - Consider node degree for balanced node degree.
  - Consider node distance to the initial cycle for limited diameter.
  - A tradeoff is needed.
- Repeat arc path selection until G' includes all nodes.



Exceptions

- In BFS stage, unable to find a return path to the current sub-graph.



1. Given current subgraph G', we select n0 in G' as the starting node in this iteration.
2. DFS finds n0 - n1 - n2 - n3, but no return path from n3 to G' | n0.
3. Discard last node in DFS path. Perform BFS.
4. Repeat 3 until BFS finds a return path or DFS path contains only two nodes n0 - n1.

- In BFS stage, unable to find a return path to the starting node of the initial cycle.

Properties

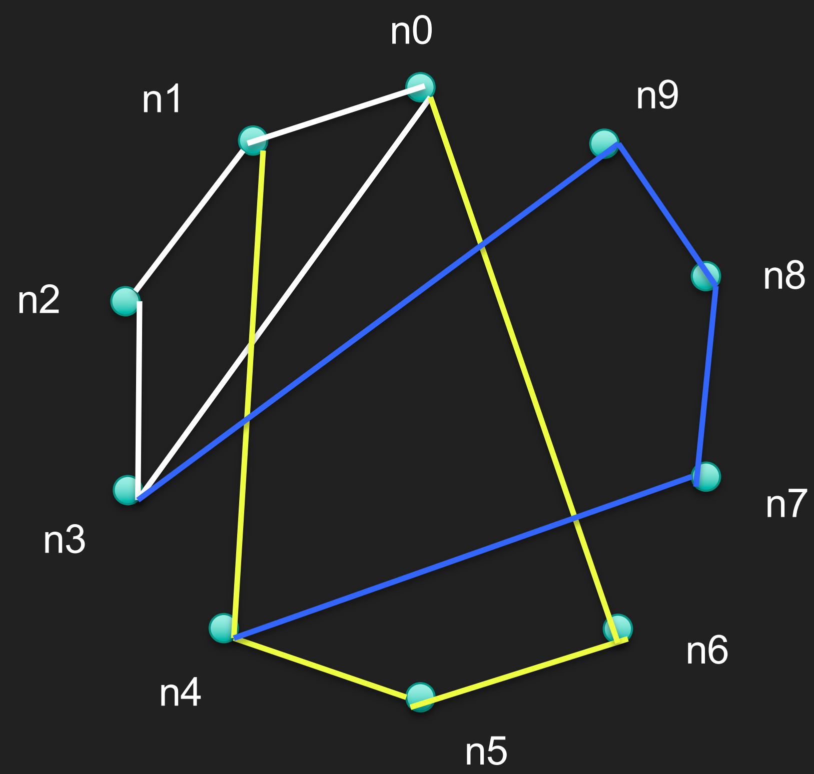
- All nodes are included
- Bi-connected
  - A cycle is a bi-connected graph.
  - Overlapping cycles that share at least two nodes are also bi-connected.
- Bounded diameter
  - Enforce a limited number of steps in DFS.
  - Use BFS to find the shortest returning path.
  - Select starting node of an arc path closer to the initial cycle.
- Balanced node degree
  - Select nodes with lower degree to be endpoints of an arc path.

Examples K10

- Maximum steps in DFS: 3
- Initial cycle: n0 – n1 – n2 – n3 – n0
- 1st arc path: n1 – n4 – n5 – n6 – n0
- 2nd arc path: n3 – n9 – n8 – n7 – n4

Comparison: sub-graph vs. base graph

- Number of edges: 12 vs. 45
- Diameter: 4 vs 1
- Node degree: { 2 or 3 } vs. 9



LANs

- Represent a pseudo-node as a node in the base graph.
- Possible optimizations:
  - Pseudo-nodes do not need to be included in the resulting sub-graph.
  - If a pseudo-node is included in an intermediate sub-graph, it provides uni-connectivity to all nodes on the LAN that are not yet included in the sub-graph.

Summary

- The algorithm applies to any connected base graph.
- The resulting sub-graph has desirable properties of a flooding topology.
- In centralized mode of dynamic flooding, the area leader can use this algorithm to compute the flooding topology.