#ASLI ENGINEERING

Synch BFS in Distributed Systems

SWIPE

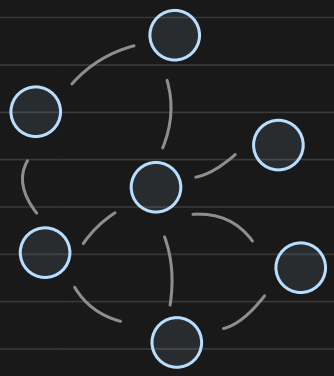
BY

ARPIT BHAYANI

Synchronous BFS in distributed systems

Breadth-First Search is a critical algorithm in Distributed Systems because it powers some key features like

- Broadcast in minimum time
- Building topological understanding
- Topological stat like Diameter and Shortest path across nodes



We discuss a Synchronous BFS algorithm in which all nodes synchronize and move ahead with each round in sync.

Output of BFS

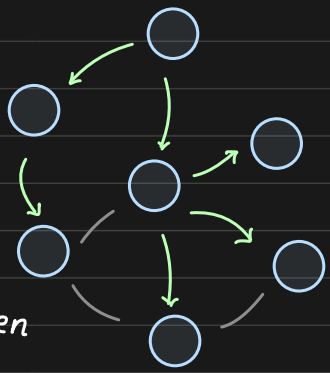
The output of this traversal is a

breadth-first directed spanning tree

↓

all nodes covered with min edges

Each node knows its parent and its children



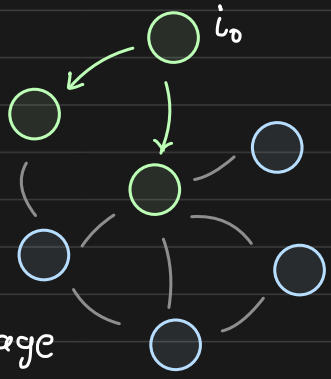
* Node may be reachead via multiple nodes but in BFS spanning tree, only one will be the parent

The algorithm

Because the algorithm is synchronous, nodes proceed in sync

Say, node i_0 initiates the BFS and there are total n nodes.

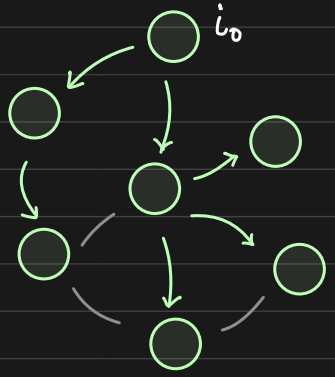
Round 1 : i_0 sends search message to its neighbours



When an unmarked node receives the message

- it marks itself
- updates its parent → to node it received message from

Round 2 : the nodes who received message in round 1 sends search message to its out going neighbours



The receiving nodes takes the action by marking themselves and continuing the process.

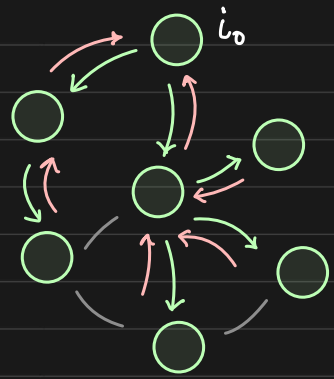
Eventually every single node will be covered and we would be having a directed spanning tree

Complexity Analysis

Given that the nodes proceed every round, the time taken to cover the entire network will be proportional to the farthest node ≈ diameter of the network

The # messages exchanged ie. communication complexity will be at max the number of edges in the network.

Conveying child pointers

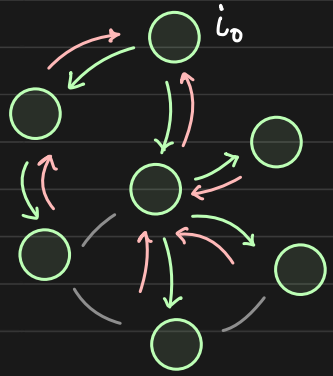


With our BFS implementation, every node upon receiving an incoming search message, knows its parent, but how would a parent know its children?

* Outgoing edge ≠ child. We are building a spanning tree here

An node, upon receiving the search message, would
if unmarked, update the local state with parent
if marked, discard the message
and inform back if the node is chosen as parent or non-parent

How would our algorithm terminate?



The most important part of any distributed algorithm is its termination.

How would the node know that BFS is done?

The approach we use is called Convergecast

Core idea: what if the nodes respond to their parent only after they got response from their descending nodes

Kind of like Post order traversal but in a distributed setup.

Nodes would respond back with parent / non-parent messages

Instead of responding immediately, and we wait until we get response from descendents

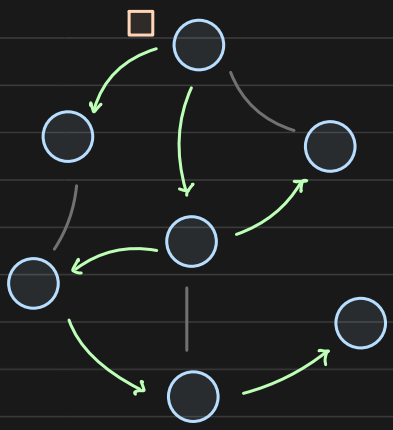
Thus, the root node will get response only after the entire network is covered.

This is called Convergecast

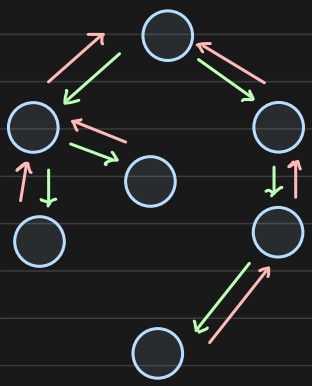
If edges are bi-directional, then above approach works

If they are uni-directional, then each node would trigger another BFS to inform the parent

Applications



Broadcast : If a node in a distributed system wants to broadcast a message m, it can initiate a SynchBFS with itself as the root. The message will be propagated throughout the tree from parent to children.



Distributed computation : In order to use the computation capability of the entire network the computation is sent from root to all the nodes and the results are propagated back to the root - like a post order traversal

Computing diameter of the network :

Each node initiates a BFS and thus knowing the farthest node. The max distances observed by each node is then flooded through another BFS to compute the global max i.e. diameter.