Termination Detection

Diffusion Algorithms

Description of the problem

- In a diffusing algorithm A the activity begins by some pre-specified source node p_i and diffuses through some portion of the network via messages.
- A global state of A is said to be quiescent provided that no process is enabled to perform any step and there are no messages in the channels.
- **Termination-detection problem for A**
  - If, sometime after the initiation of A by process p_i, A ever reaches a quiescent global state, then eventually p_i must determine termination by setting a local variable $done_i$ to have the value TRUE.

The DijkstraScholten Algorithm Main Ideas

- Augment the underlying algorithm A with the construction and maintenance of a spanning tree of the graph nodes currently involved in A.
  - Use ideas similar to those in F-SpanningTree
- This tree is rooted at the source node p_i.
- The tree can shrink and grow repeatedly, incorporating the same node many times.

The DijkstraScholten Algorithm Main Ideas

- Each process other than the source designates the neighbor from which it first receives an A message as its parent in the spanning tree.
- Any subsequent A message is immediately acknowledged; only the first remains unacknowledged.
- The source process immediately acknowledges any A message it receives.

The DijkstraScholten Algorithm Main Ideas

- Shrinking of the spanning tree
  - Use a convergecast algorithm to report termination to source process.
  - If a process p enters in a state in which both of the following conditions hold:
    - the state of p_i is quiescent.
    - all its outgoing A messages have been acknowledged

    does the following:
    - If $p \neq p_i$, p sends an acknowledgment to its parent and deletes all information regarding the execution of the DijkstraScholten Algorithm
    - If $p = p_i$, it sets $done_i$ to TRUE.

The DijkstraScholten Algorithm Main Ideas

- The spanning tree in DijkstraScholten can grow and shrink repeatedly in a different way each time:
- If a process that has clean up, receives an A message, it participates again in the spanning tree construction.
  - This may happen any number of times.

The DijkstraScholten Algorithm Example