RAMAIAH
Institute of Technology

M.S. Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of Computer Science and Engineering

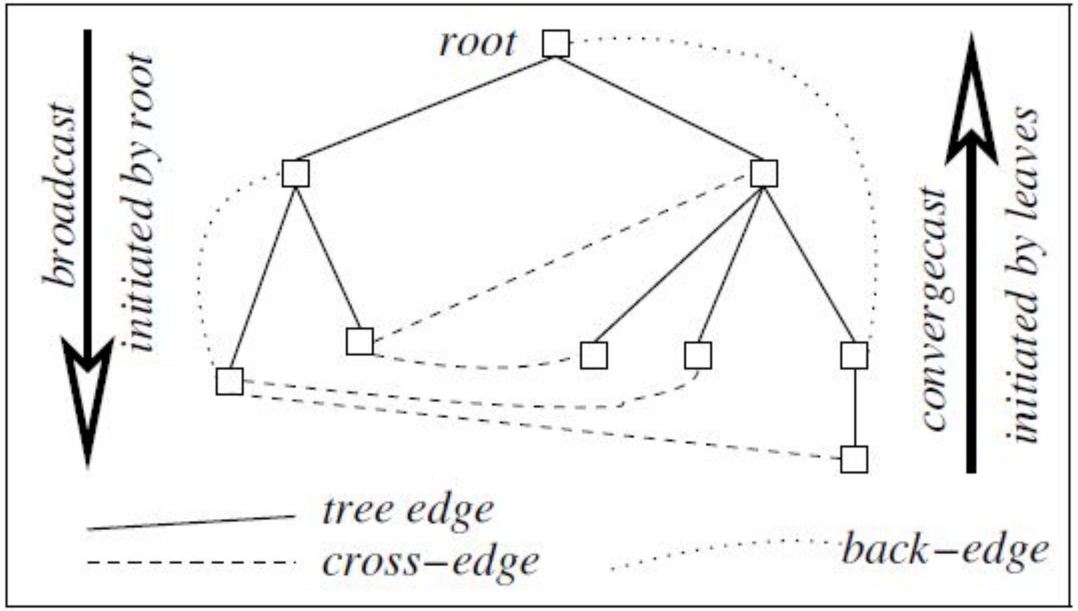
Course Name: Distributed Systems
Course Code: CSE20
Credits: 3:0:0:1

Term: September – December 2020

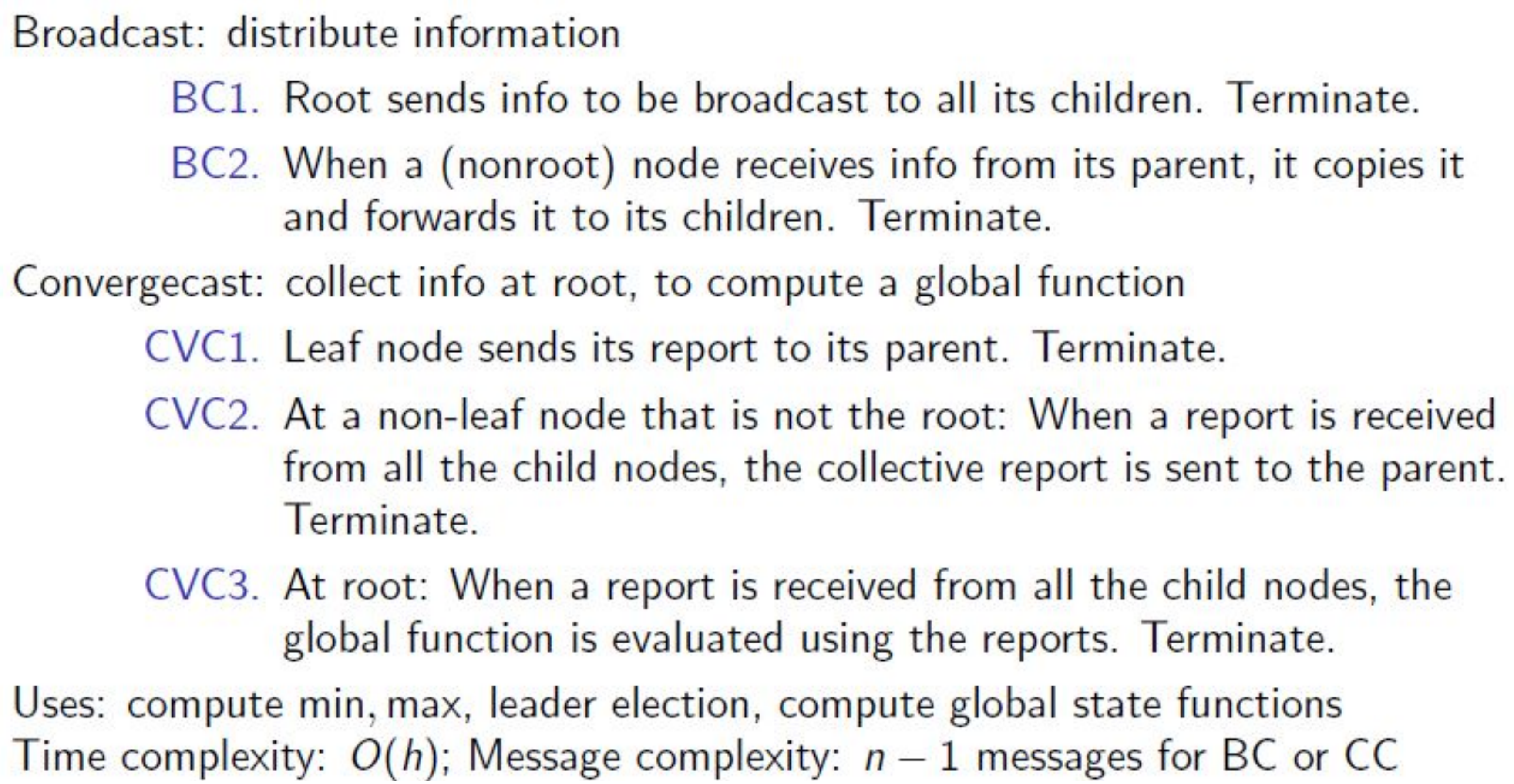
Faculty:
Sini Anna Alex

Broadcast and Converge cast

Broadcast and Convergecast on a Tree



Broadcast and Converge cast



Broadcast: distribute information

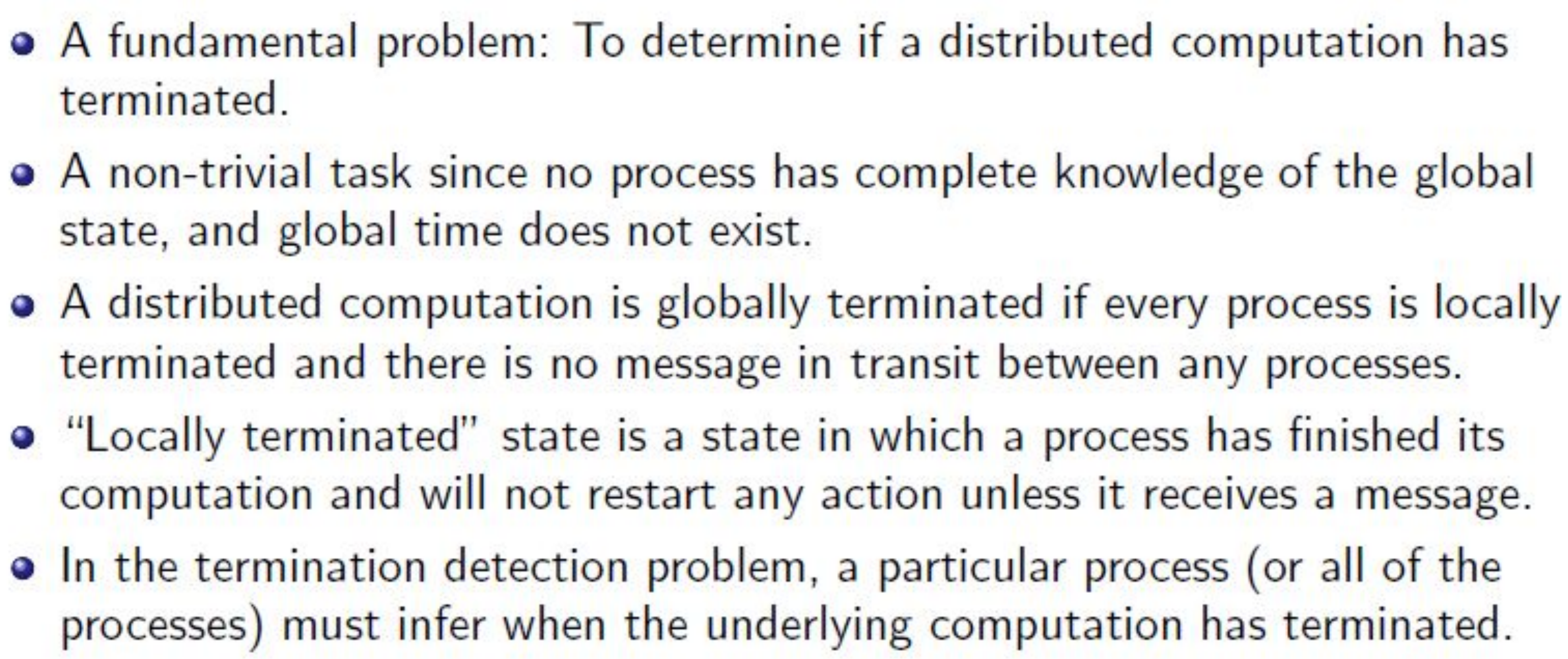
- BC1. Root sends info to be broadcast to all its children. Terminate.
- BC2. When a (nonroot) node receives info from its parent, it copies it and forwards it to its children. Terminate.

Convergecast: collect info at root, to compute a global function

- CVC1. Leaf node sends its report to its parent. Terminate.
- CVC2. At a non-leaf node that is not the root: When a report is received from all the child nodes, the collective report is sent to the parent. Terminate.
- CVC3. At root: When a report is received from all the child nodes, the global function is evaluated using the reports. Terminate.

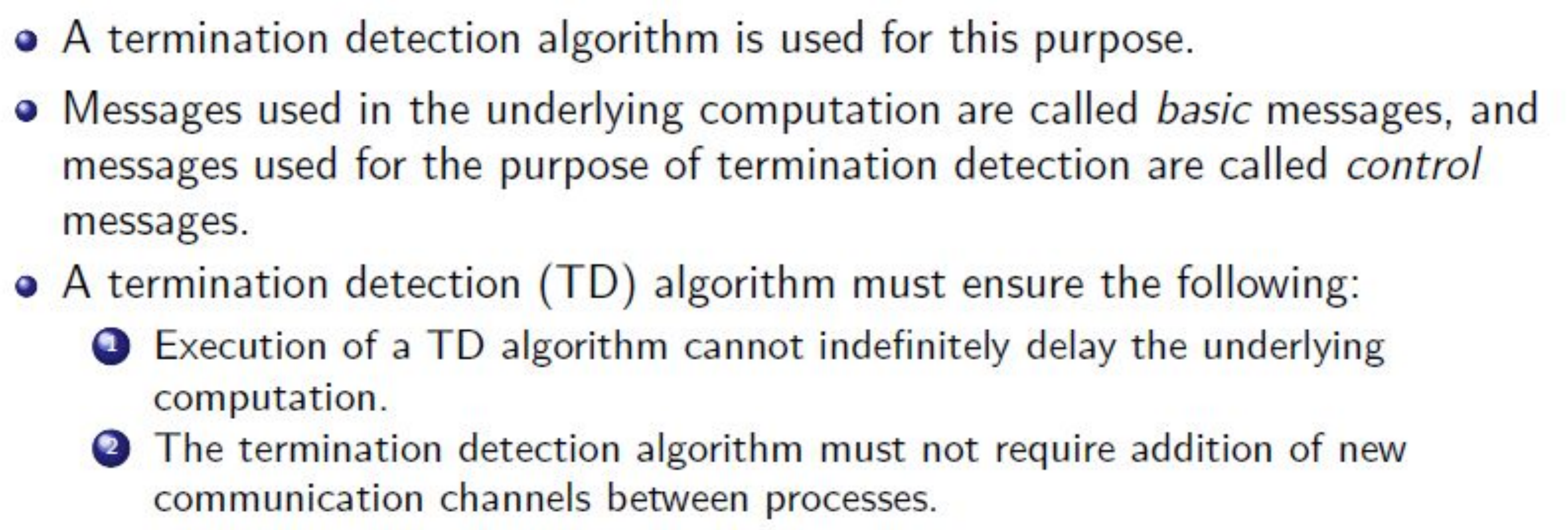
Uses: compute min, max, leader election, compute global state functions
Time complexity: $O(h)$; Message complexity: $n-1$ messages for BC or CC

Termination Detection



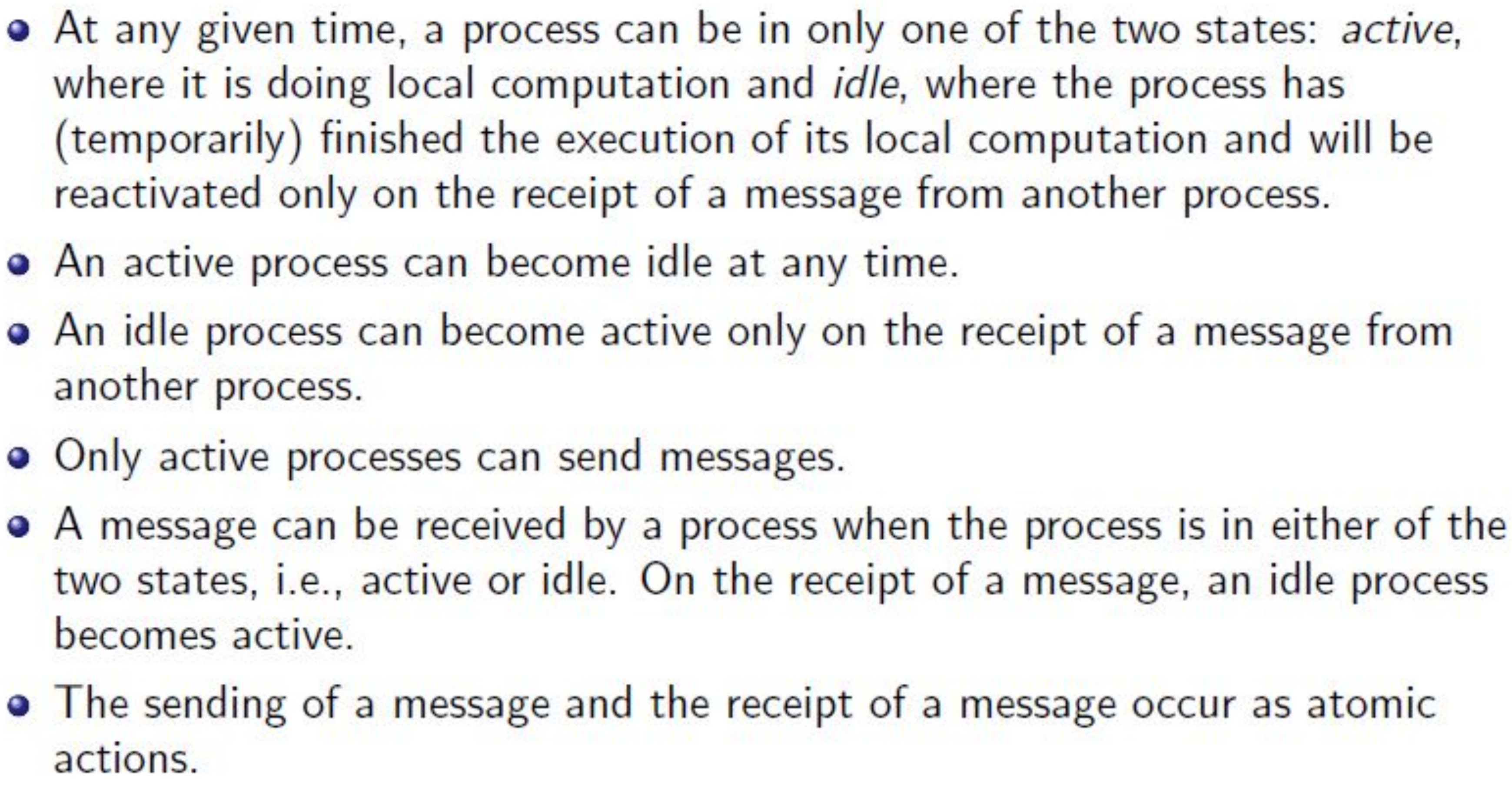
- A fundamental problem: To determine if a distributed computation has terminated.
- A non-trivial task since no process has complete knowledge of the global state, and global time does not exist.
- A distributed computation is globally terminated if every process is locally terminated and there is no message in transit between any processes.
- "Locally terminated" state is a state in which a process has finished its computation and will not restart any action unless it receives a message.
- In the termination detection problem, a particular process (or all of the processes) must infer when the underlying computation has terminated.

Termination Detection



- A termination detection algorithm is used for this purpose.
- Messages used in the underlying computation are called *basic* messages, and messages used for the purpose of termination detection are called *control* messages.
- A termination detection (TD) algorithm must ensure the following:
    1. Execution of a TD algorithm cannot indefinitely delay the underlying computation.
    2. The termination detection algorithm must not require addition of new communication channels between processes.

System Model



- At any given time, a process can be in only one of the two states: *active*, where it is doing local computation and *idle*, where the process has (temporarily) finished the execution of its local computation and will be reactivated only on the receipt of a message from another process.
- An active process can become idle at any time.
- An idle process can become active only on the receipt of a message from another process.
- Only active processes can send messages.
- A message can be received by a process when the process is in either of the two states, i.e., active or idle. On the receipt of a message, an idle process becomes active.
- The sending of a message and the receipt of a message occur as atomic actions.

Thank you