## 3.8 KNAPP'S CLASSIFICATION OF DISTRIBUTED DEADLOCK DETECTION ALGORITHMS

The four classes of distributed deadlock detection algorithm are:

1. Path-pushing
2. Edge-chasing
3. Diffusion computation
4. Global state detection

### 3.8.1 Path Pushing algorithms

- In path pushing algorithm, the distributed deadlock detection are detected by maintaining an explicit global wait for graph.
- The basic idea is to build a global WFG (Wait For Graph) for each site of the distributed system.
- At each site whenever deadlock computation is performed, it sends its local WFG to all the neighbouring sites.
- After the local data structure of each site is updated, this updated WFG is then passed along to other sites, and the procedure is repeated until some site has a sufficiently complete picture of the global state to announce deadlock or to establish that no deadlocks are present.
- This feature of sending around the paths of global WFGhas led to the term path-pushing algorithms.
  **Examples:**Menasce-Muntz , Gligor and Shattuck, Ho and Ramamoorthy, Obermarck

### 3.8.2 Edge Chasing Algorithms

- The presence of a cycle in a distributed graph structure is be verified by propagating special messages called probes, along the edges of the graph.
- These probe messages are different than the request and reply messages.
- The formation of cycle can be deleted by a site if it receives the matching probe sent by it previously.
- Whenever a process that is executing receives a probe message, it discards this message and continues.
- Only blocked processes propagate probe messages along their outgoing edges.
- Main advantage of edge-chasing algorithms is that probes are fixed size messages which is normally very short.
  **Examples:**Chandy et al., Choudhary et al., Kshemkalyani–Singhal, Sinha–Natarajan algorithms.

### 3.8.3 Diffusing Computation Based Algorithms

- In diffusion computation based distributed deadlock detection algorithms, deadlock detection computation is diffused through the WFG of the system.
- These algorithms make use of echo algorithms to detect deadlocks.
- This computation is superimposed on the underlying distributed computation.
- If this computation terminates, the initiator declares a deadlock.
- To detect a deadlock, a process sends out query messages along all the outgoing edges in the WFG.
- These queries are successively propagated (i.e., diffused) through the edges of the WFG.
- When a blocked process receives first query message for a particular deadlock detection initiation, it does not send a reply message until it has received a reply message for every query it sent.

- For all subsequent queries for this deadlock detection initiation, it immediately sends back a reply message.
- The initiator of a deadlock detection detects a deadlock when it receives reply for every query it had sent out.

**Examples:**Chandy–Misra–Haas algorithm for one OR model, Chandy–Herman algorithm

### 3.8.4 Global state detection-based algorithms

Global state detection based deadlock detection algorithms exploit the following facts:

1. A consistent snapshot of a distributed system can be obtained without freezing the underlying computation.
2. If a stable property holds in the system before the snapshot collection is initiated, this property will still hold in the snapshot.

Therefore, distributed deadlocks can be detected by taking a snapshot of the system and examining it for the condition of a deadlock

## 3.9 MITCHELL AND MERRITT'S ALGORITHM FOR THE SINGLE-RESOURCE MODEL

- This deadlock detection algorithm assumes a single resource model.
- This detects the local and global deadlocks each process has assumed two different labels namely private and public each label is accountant the process id guarantees only one process will detect a deadlock.
- Probes are sent in the opposite direction to the edges of the WFG.
- When a probe initiated by a process comes back to it, the process declares deadlock.

**Features:**

1. Only one process in a cycle detects the deadlock. This simplifies the deadlock resolution – this process can abort itself to resolve the deadlock. This algorithm can be improvised by including priorities, and the lowest priority process in a cycle detects deadlock and aborts.
2. In this algorithm, a process that is detected in deadlock is aborted spontaneously, even though under this assumption phantom deadlocks cannot be excluded. It can be shown, however, that only genuine deadlocks will be detected in the absence of spontaneous aborts.

Each node of the WFG has two local variables, called labels:

1. a private label, which is unique to the node at all times, though it is not constant.
2. a public label, which can be read by other processes and which may not be unique.

Each process is represented as u/v where u and u are the public and private labels, respectively. Initially, private and public labels are equal for each process. A global WFG is maintained and it defines the entire state sof the system.

- The algorithm is defined by the four state transitions as shown in Fig.3.10, where z = inc(u, v), and inc(u, v) yields aunique label greater than both u and v labels that are notshown do not change.
- The transitions in the defined by the algorithm are block, activate , transmit and detect.
- **Block** creates an edge in the WFG.
- Two messages are needed, one resource request and onemessage back to the blocked process to inform it of thepublic label of the process it is waiting for.
- **Activate** denotes that a process has acquired the resourcefrom the process it was

waiting for.

- **Transmit** propagates larger labels in the opposite directionof the edges by sending a probe message.

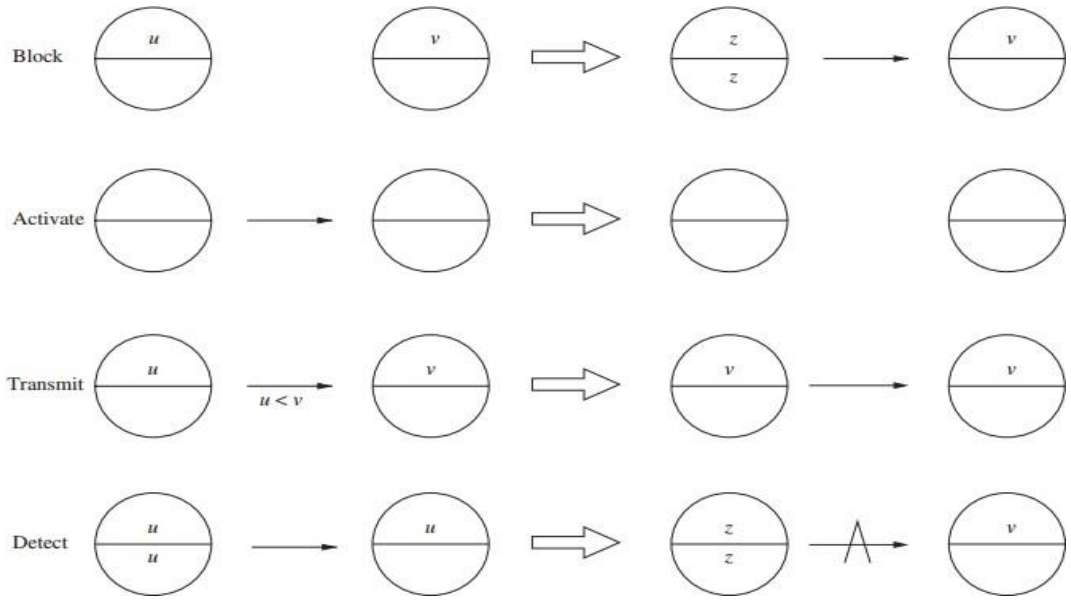


**Fig 3.10: Four possible state transitions**

- **Detect** means that the probe with the private label of some process has returned to it, indicating a deadlock.
- This algorithm can easily be extended to include priorities, so that whenever a deadlock occurs, the lowest priority process gets aborted.
- This priority based algorithm has two phases.
    1. The first phase is almost identical to the algorithm.
    2. The second phase the smallest priority is propagated around the circle. The propagation stops when one process recognizes the propagated priority as its own.

**Message Complexity:**

If we assume that a deadlock persists long enough to be detected, the worst-case complexity of the algorithm is s(s - 1)/2 Transmit steps, where s is the number of processes in the cycle.