An example of distributed deadlock detection algorithm (IBM DB2).

Distributed transaction: a transaction that has been split into into subtransactions executed at different nodes.
When a subtransaction T1 of T activates a subtransaction T2 of T at a different node, T1 waits for the termination of T2.

A transaction Ti waits for a transaction Tj in two different cases:
- Ti waits for a lock to be released by Tj.
- Ti waits for the termination of the subtransaction Tj executed at a different node.

We use wait-for sequences local at nodes.
Assume we have the following wait-for sequence at node 1:

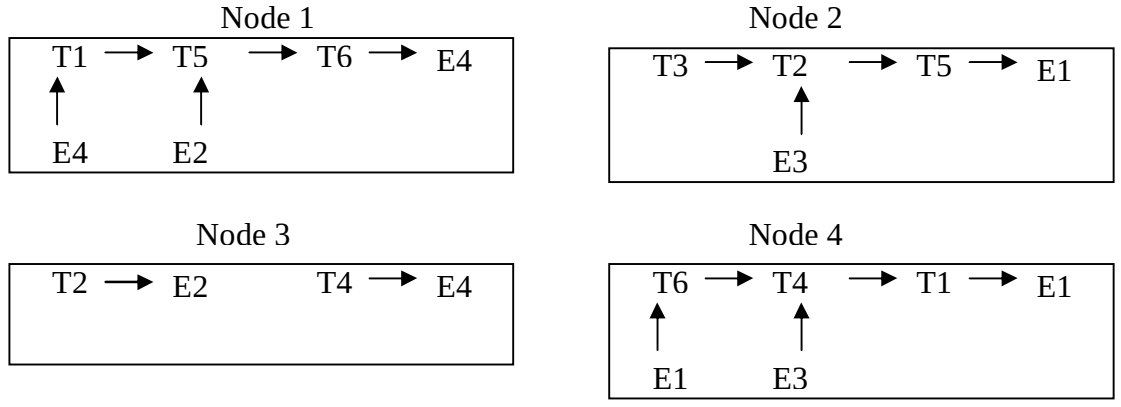
E_{in} -> Ti -> Tj-> E_{out}

We have that:
- a subtransaction executed at node n_{in} waits for Ti
- Ti waits for a lock to be released by Tj at node 1
- Tj waits for the completion of a subtransaction executed at node n_{out}.

Steps of the algorithm:
1) each node must identify the wait-for sequences in its wait-for-graph
2) to guarantee the same deadlock be detected only at one node, each node sends wait-for sequences according to the following rules:
   - only sequences E_{in} -> Ti -> Tj-> E_{out} such that i>j are sent
   - sequences are sent only to the node n_{out}
3) upon receiving the wait-for-sequences, each node updates its local wait-for graph; if a deadlock is detected, one transaction is selected and aborted. The decision is sent to all the other nodes.

Exercise (Distributed Deadlock detection algorithm)

Assume we have the following local wait-for graphs:



Show the application of the distributed deadlock detection algorithm.

Step 1)

Node 1:
E_4 -> T1 -> T6-> E_4
E_2 -> T5 -> T6-> E_4

Node 2:
E_3 -> T2 -> T5-> E_1

Node 3:
-

Node 4:
E_1 -> T6 -> T1-> E_1
E_3 -> T4-> T1-> E_1

Step 2)

Node 1:
sequences are not sent (i<j)

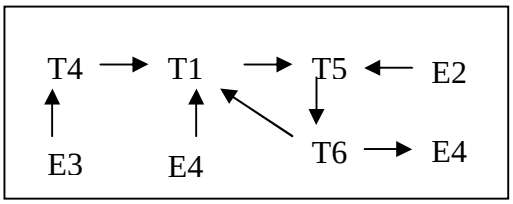
Node 2:
sequences are not sent (i<j)

Node 3:
There are no sequences

Node 4:
E_1 -> T6 -> T1-> E_1 is sent to node 1
E_3 -> T4-> T1-> E_1 is sent to node 1

Step 3)

Node 1:
Updates its local wait-for graph:



A deadlock is detected (cycle T1, T5, T6). One transaction is rolled back.
The decision is sent to the other nodes.

Node 2:
Local-wait-for graph unchanged

Node 3:
Local-wait-for graph unchanged

Node 4:
Local-wait-for graph unchanged