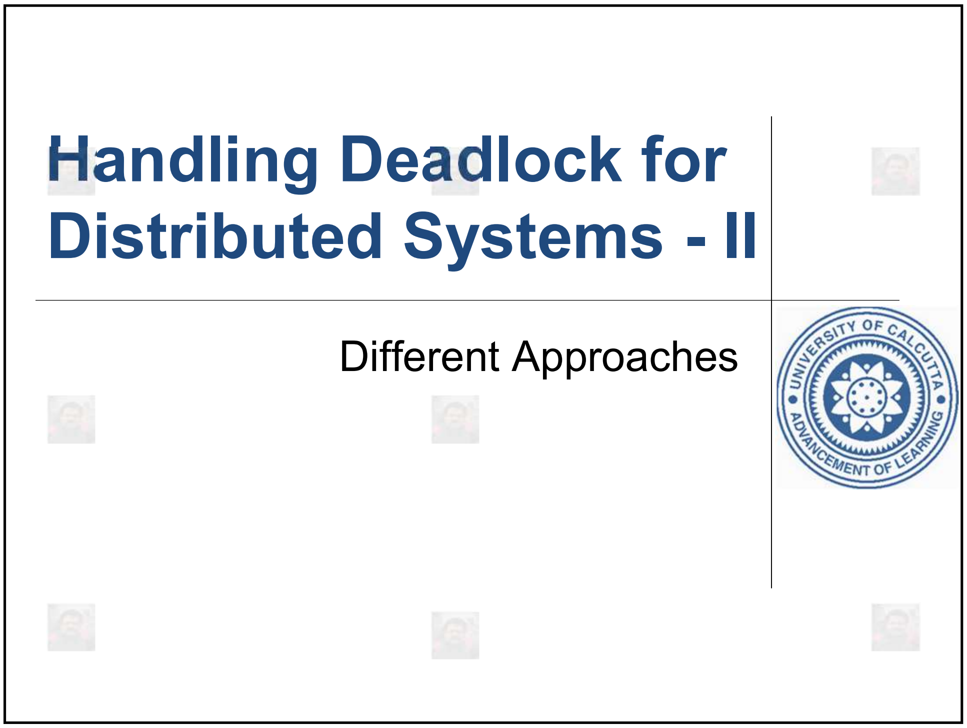
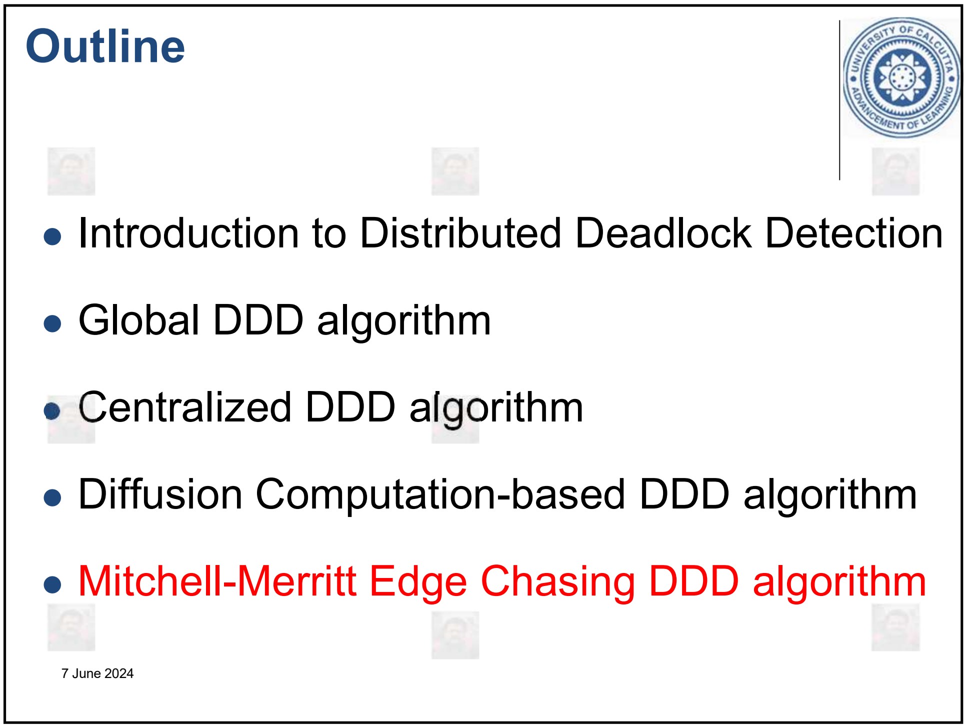


Handling Deadlock for Distributed Systems - II

Different Approaches

1



Outline

- Introduction to Distributed Deadlock Detection
- Global DDD algorithm
- Centralized DDD algorithm
- Diffusion Computation-based DDD algorithm
- Mitchell-Merritt Edge Chasing DDD algorithm

7 June 2024

2

Mitchell – Merritt DDD Algorithm

- It is an *edge chasing* algorithm where Control messages are sent over WFG edges to detect cycles
- Each process is represented as u/v where u and v are the public and private labels, respectively.
- Labels are initially identical and unique for each node

3

Mitchell – Merritt DDD Algorithm

- Labels of a process change when it gets blocked on a resource request
- Labels also change when it waits for a process having a larger public label
- A wait-for edge with a specific relation between public and private labels of its source and destination processes indicates presence of a deadlock

4

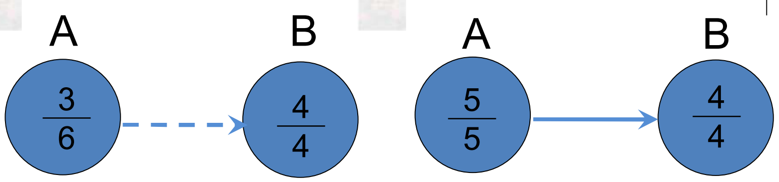
State Transitions

- Initiate: Set same random value for u, v. Value for each node need to be unique.
- Block: This will be in effect every time a process is blocked.
  - Add a new block edge in the WFG
  - Set both u, v of the blocked process with k:
    - k = f(u1, u2) yields a unique label greater than both u1 and u2 – the two public labels for the blocking and blocked processes

State Transitions

- Transmit: If public label of a blocking process is greater than that of the blocked process in the WFG, then this higher public label propagates in the opposite direction of the edges.
- Detect: If the public and private labels of a blocked process are same, and the value is again same as the public label of the blocking process, a deadlock is detected

Mitchell – Merritt DDD Algorithm

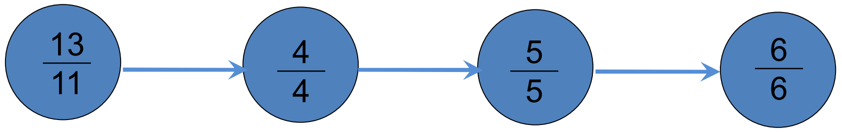


- Block rule
- Assume, f(u1, u2)=Maximum(u1, u2)+1

7 June 2024

7

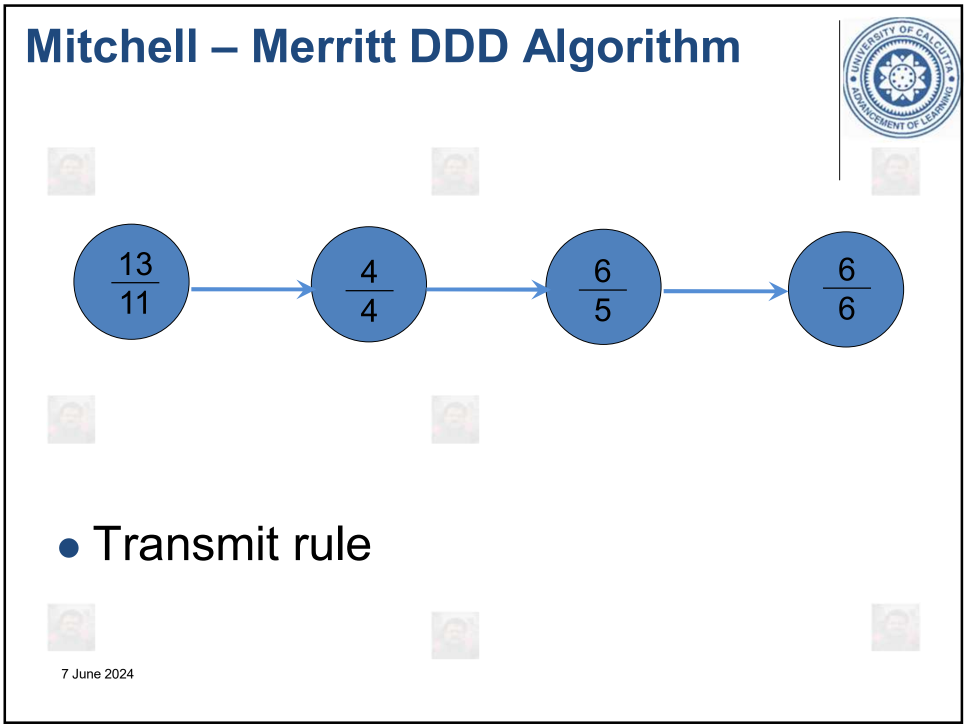
Mitchell – Merritt DDD Algorithm



- Transmit rule

7 June 2024

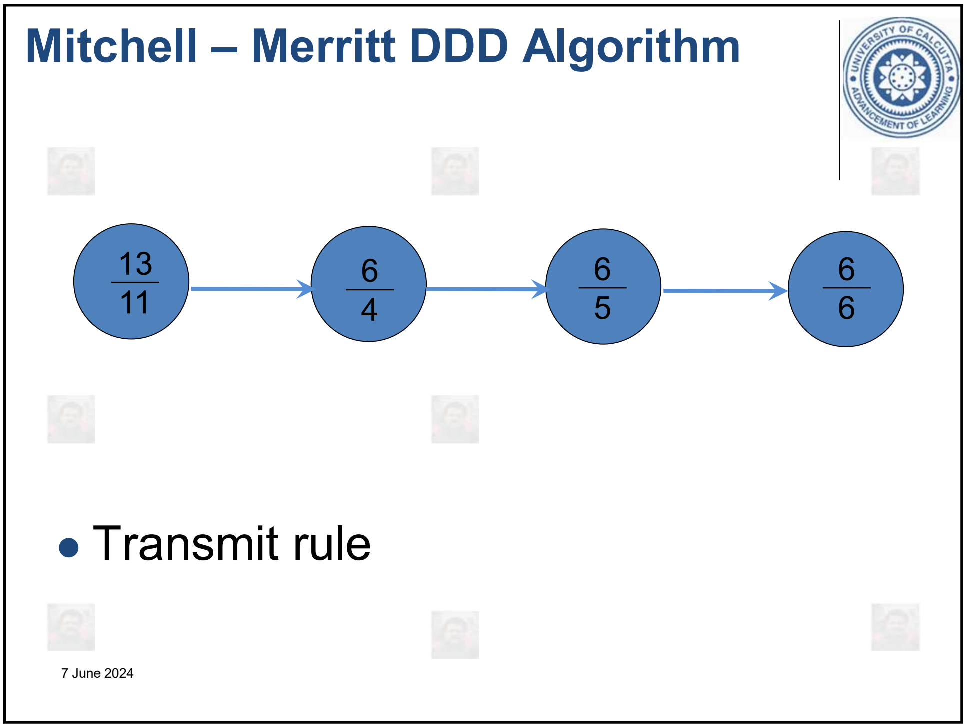
8



Mitchell – Merritt DDD Algorithm

- Transmit rule

9

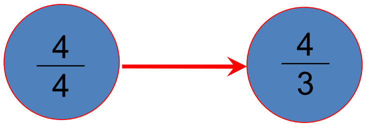


Mitchell – Merritt DDD Algorithm

- Transmit rule

10

Mitchell – Merritt DDD Algorithm

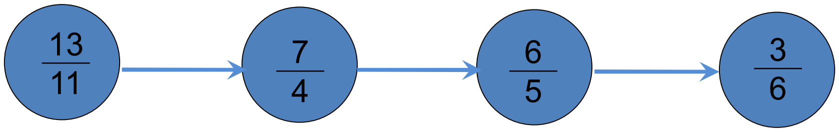


- Detection rule

7 June 2024

11

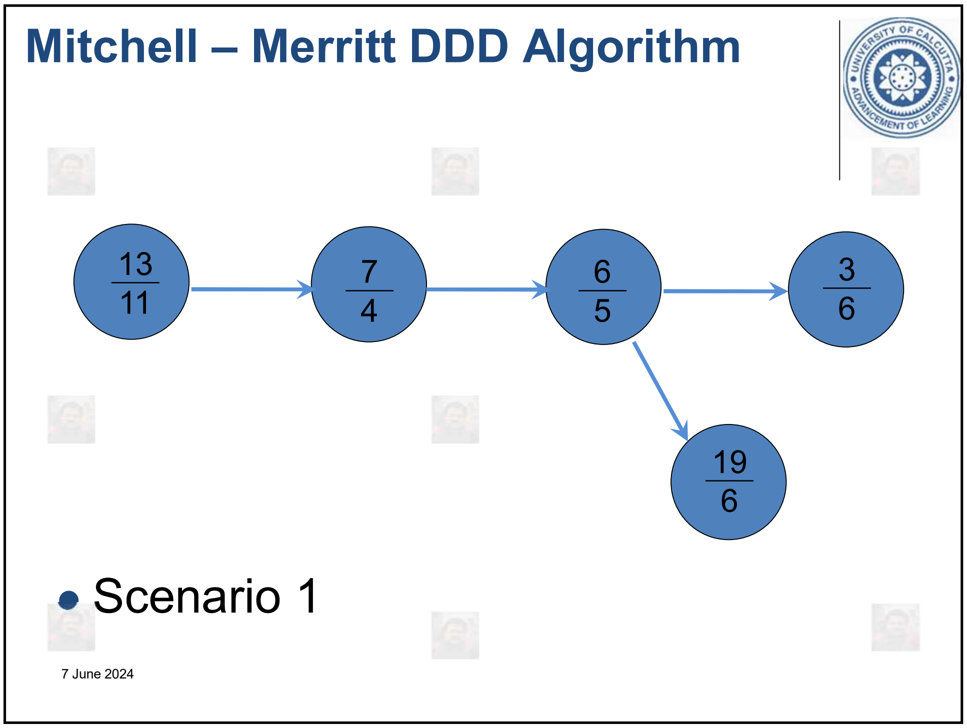
Mitchell – Merritt DDD Algorithm



- Scenario 1

7 June 2024

12

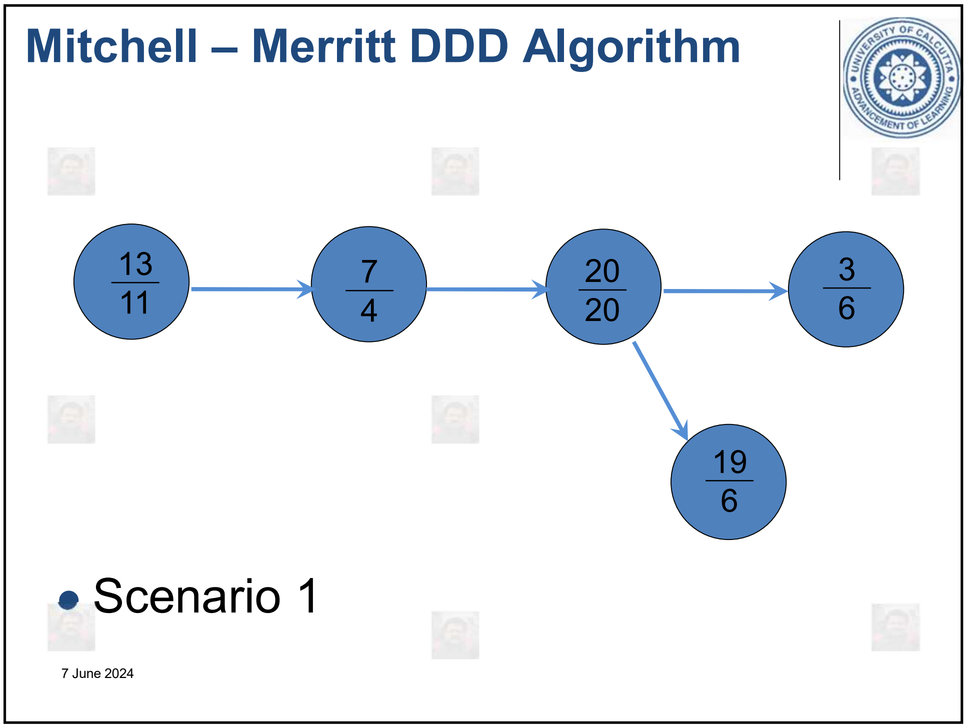


Mitchell – Merritt DDD Algorithm

- Scenario 1

7 June 2024

13

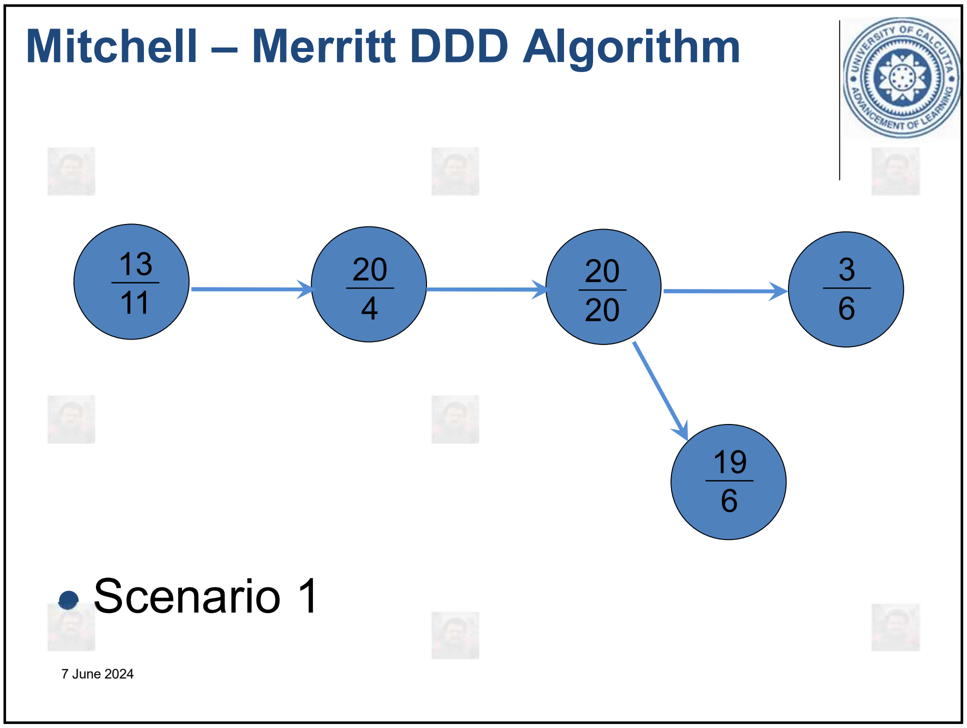


Mitchell – Merritt DDD Algorithm

- Scenario 1

7 June 2024

14

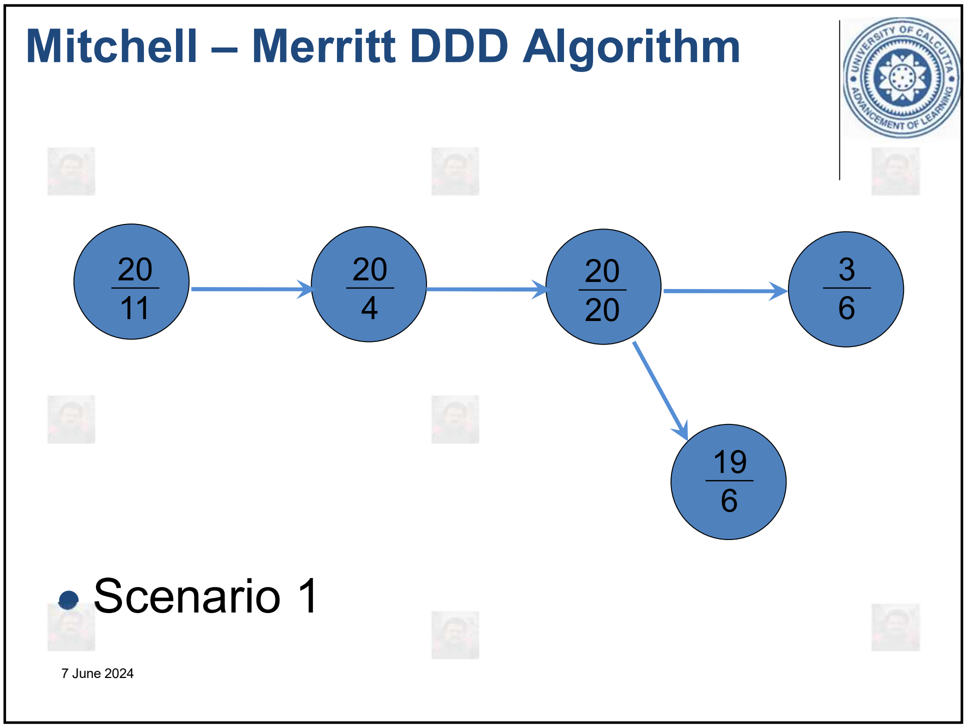


Mitchell – Merritt DDD Algorithm

- Scenario 1

7 June 2024

15



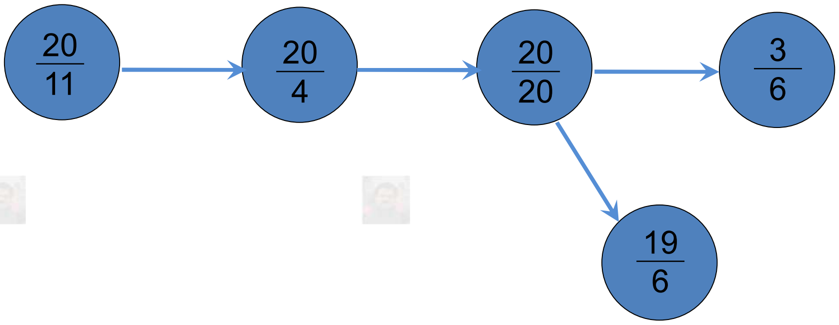
Mitchell – Merritt DDD Algorithm

- Scenario 1

7 June 2024

16

Mitchell – Merritt DDD Algorithm

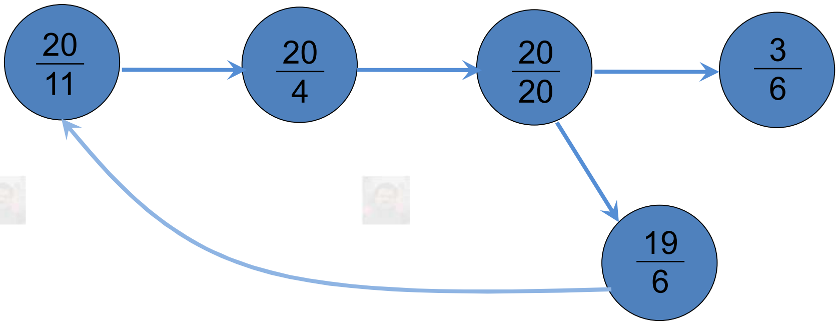


- Scenario 2

7 June 2024

17

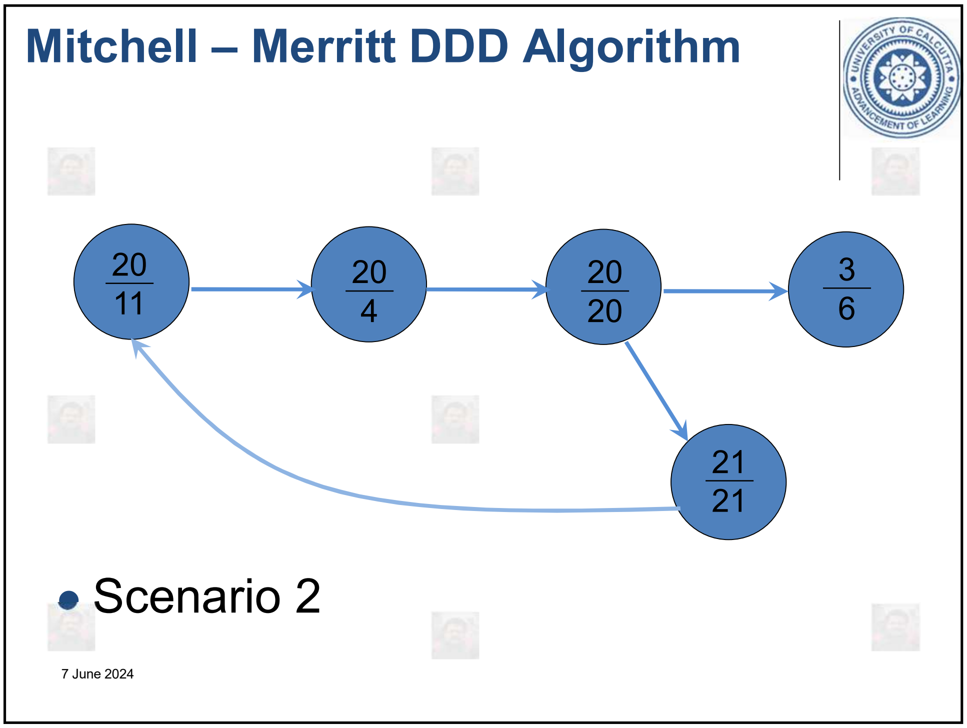
Mitchell – Merritt DDD Algorithm



- Scenario 2

7 June 2024

18

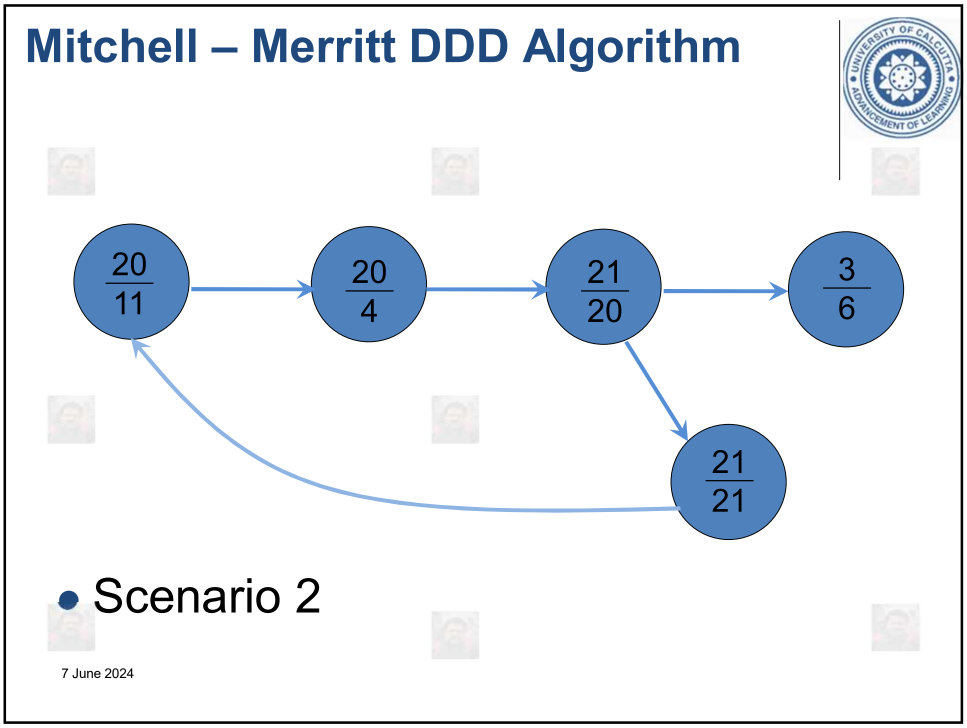


Mitchell – Merritt DDD Algorithm

- Scenario 2

7 June 2024

19

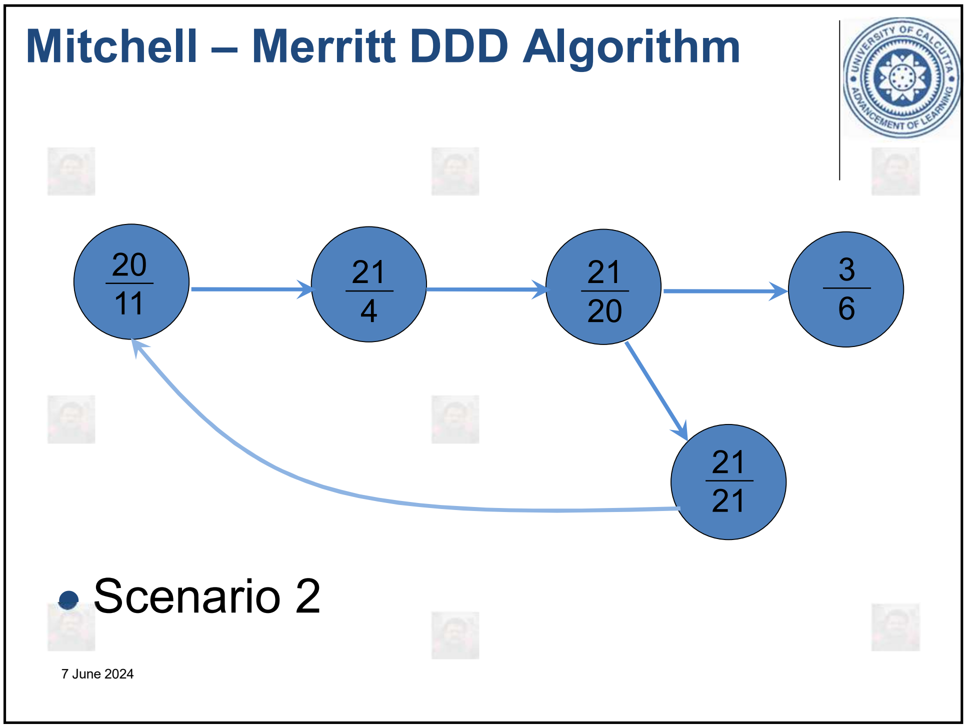


Mitchell – Merritt DDD Algorithm

- Scenario 2

7 June 2024

20

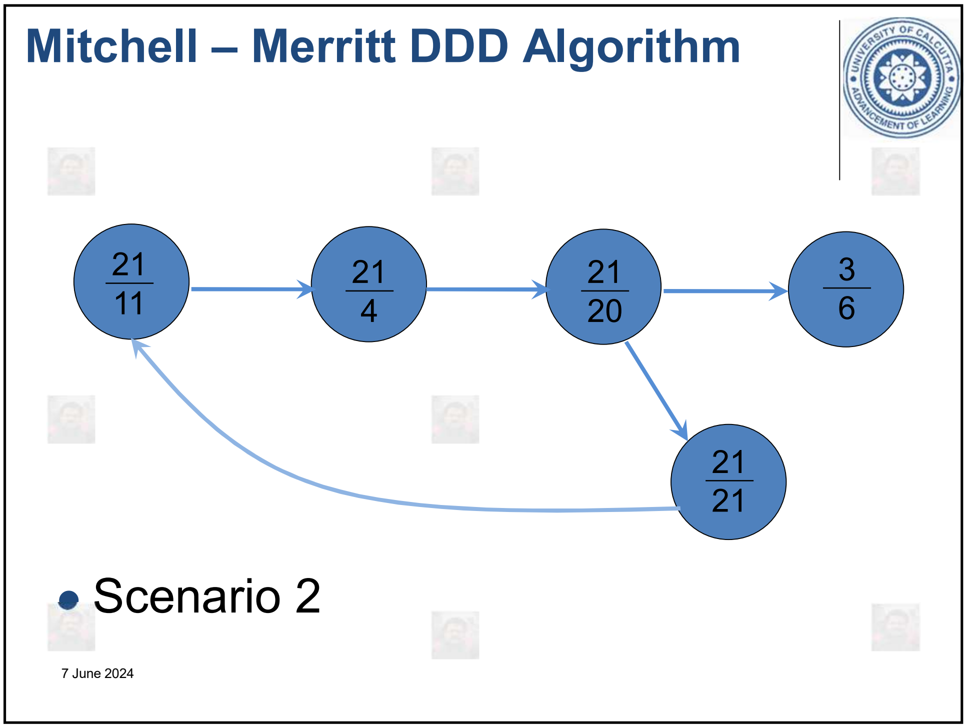


Mitchell – Merritt DDD Algorithm

- Scenario 2

7 June 2024

21



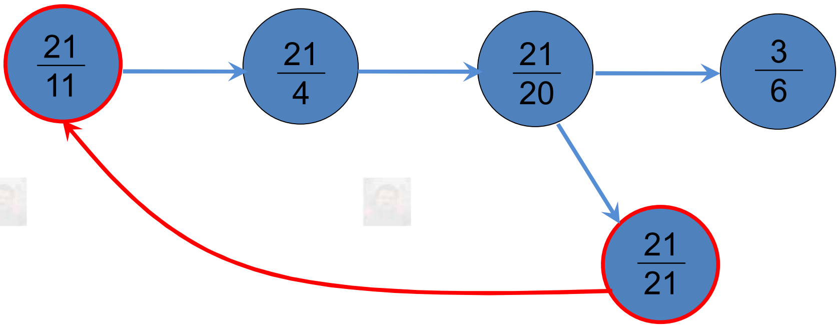
Mitchell – Merritt DDD Algorithm

- Scenario 2

7 June 2024

22

Mitchell – Merritt DDD Algorithm

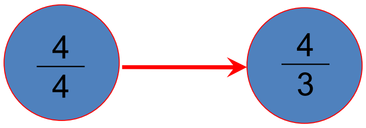


- Scenario 2

7 June 2024

23

Mitchell – Merritt DDD Algorithm



- Detection rule

7 June 2024

24

Few more thoughts….

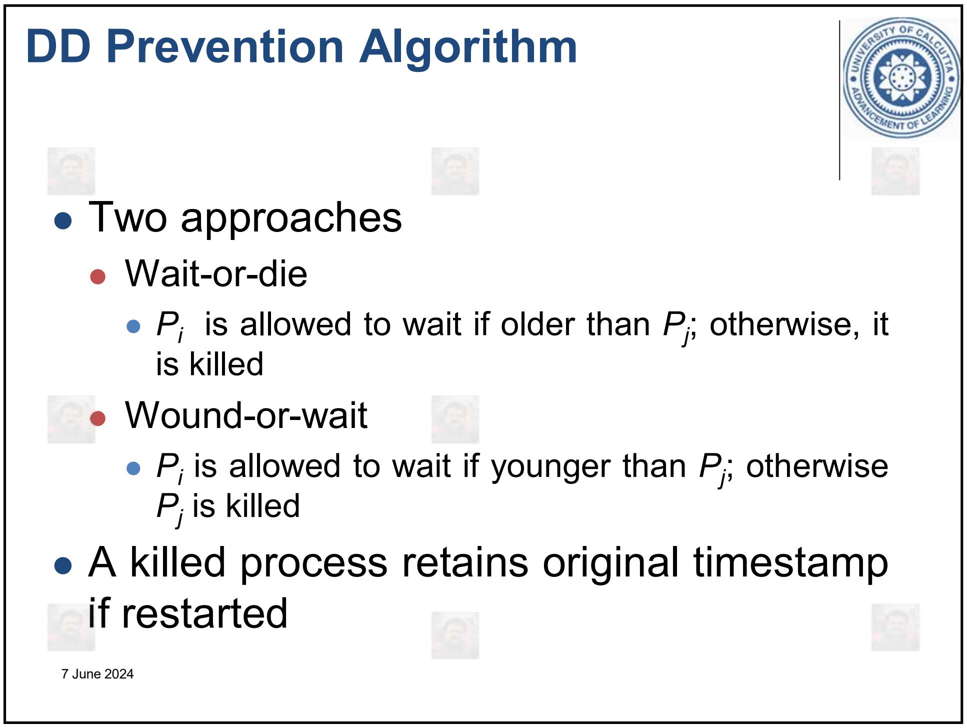
25

DD Prevention Algorithm

- The basic idea is to ensure that Circular wait does not occur
- Time-stamp creation of a process
  - When process P_k requests a resource allocated to P_m, time-stamps of P_k and P_m are used to decide whether P_k can wait for P_m

7 June 2024

26

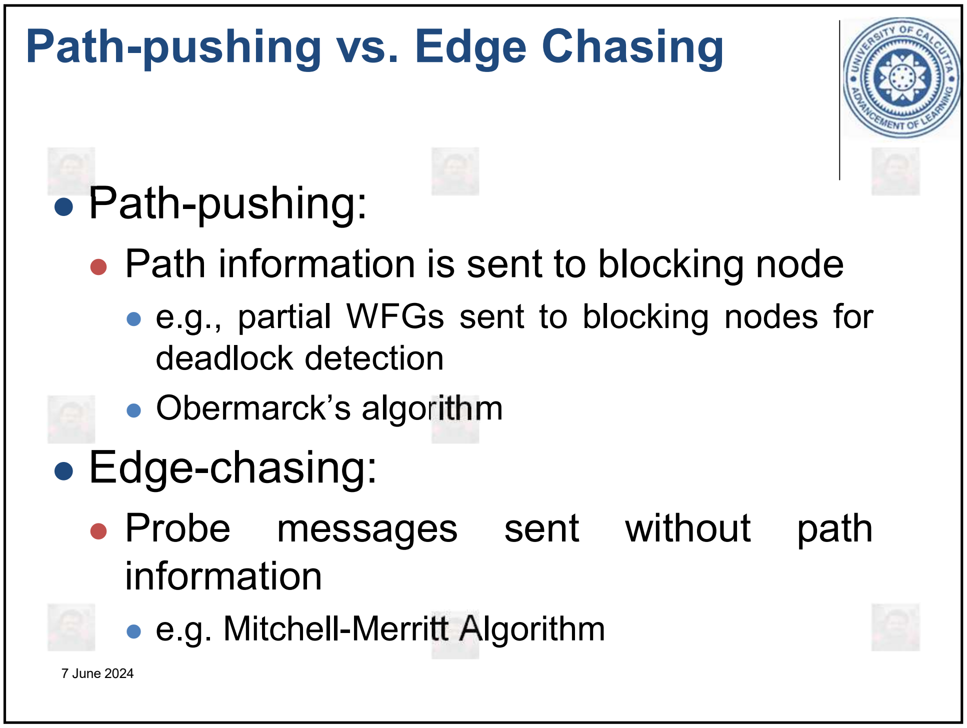


DD Prevention Algorithm

- Two approaches
  - Wait-or-die
    - P_i is allowed to wait if older than P_j; otherwise, it is killed
  - Wound-or-wait
    - P_i is allowed to wait if younger than P_j; otherwise P_j is killed
- A killed process retains original timestamp if restarted

7 June 2024

27



Path-pushing vs. Edge Chasing

- Path-pushing:
  - Path information is sent to blocking node
    - e.g., partial WFGs sent to blocking nodes for deadlock detection
    - Obermarck's algorithm
- Edge-chasing:
  - Probe messages sent without path information
    - e.g. Mitchell-Merritt Algorithm

7 June 2024

28

Thanks for your kind attention

Questions??