Laboratory session: Discovery

Distributed Systems

Universiteit Antwerpen

Project

- Goal of the session:
  - to enable 'node life-cycle' in system Y
- Team work:
  - work in your groups (available on Blackboard)
- Node life-cycle:
  1. Discovery
  2. Bootstrap
  3. Shutdown
  4. Failure

Node life-cycle operations

- Discovery: Auto-discover the Naming server and existing nodes in the network.
- Bootstrap: Starting each node, initializing local parameters (previous node, next node), and updating parameters of existing nodes
- Shutdown: A node leaves the ring network of system Y, and updates parameters of neighbor nodes and the Naming server
- Failure: Node's failure is detected. Naming server updates parameters of remaining nodes.

Discovery and Bootstrap (1/2)

1. Develop a method that sends a multicast message to existing nodes and the Naming server (i.e., all nodes in local network). This method will be needed in step 3.
2. Develop a method that calculates a hash based on the node name. Try to reuse this functionality from the previous session.
3. During bootstrap the node will send its name and its IP address to all nodes and the Naming server in the network using multicast.
4. The Naming server receives the multicast message and executes the following steps:
   a) Calculates the hash of a node name.
   b) Adds the hash and IP address in its map data structure (see previous session).
   c) Responds to the new node with the number of existing nodes that are currently in the network.

Discovery and Bootstrap (2/2)

5. Other nodes in the network also receive this multicast message, and they perform the following:
   a) Calculate the hash of the node that sent multicast message.
   b) Each node calculates its own hash, and stores it as currentID.
   c) If currentID < hash < nextID, nextID = hash, current node updates its own parameter nextID, and sends response to node giving the information on currentID and nextID
   d) if previousID < hash < currentID, previousID = hash, current node updates its own parameter previousID, and sends response to node giving the information on currentID and previousID
6. A node that sent multicast message receives response from Naming server:
   a) if the number of existing nodes in the network is <1, it means that this is the only node in network (this node is its previous and next node, previousID = currentID, nextID = currentID).
   b) If the number of existing nodes in the network is >1, this node receives parameters for its previous and next node of corresponding nodes in network.

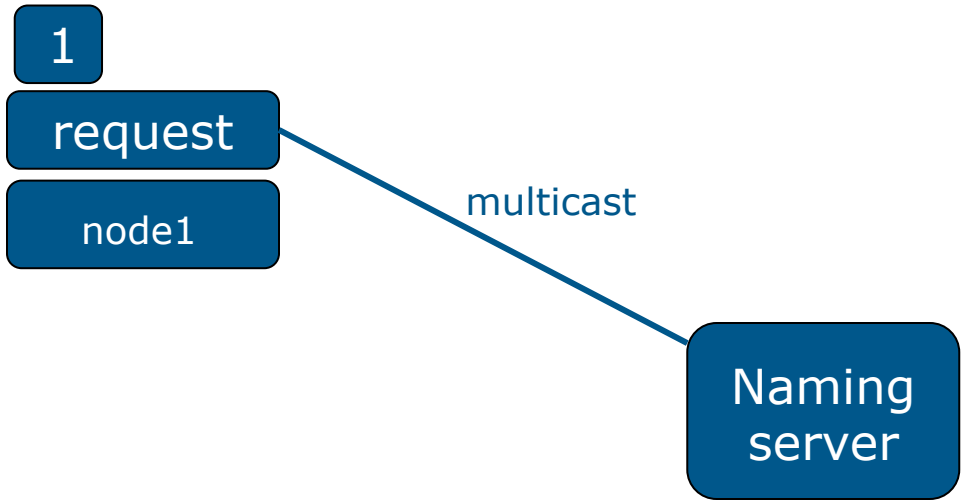
Shutdown

- Send the ID of the next node to the previous node. In the previous node, the next node parameter will be updated according to this information.
- Send the ID of the previous node to the next node. In the next node, the previous node parameter will be updated according tot his information.
- Remove the node from the Naming server’s Map.

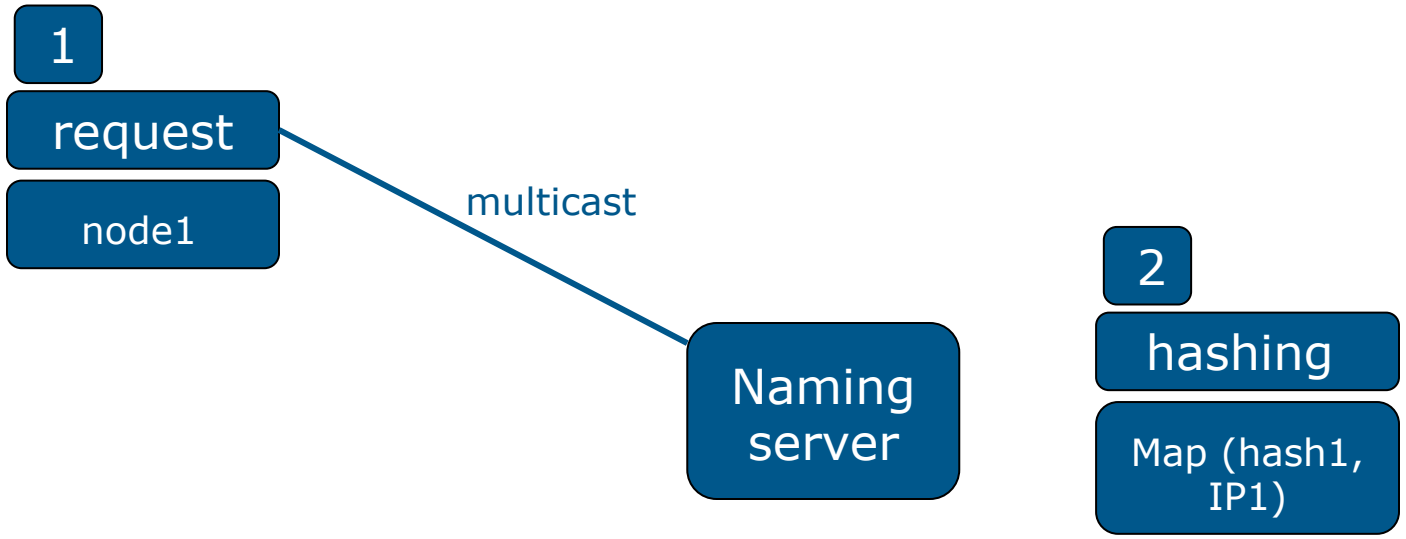
Failure

- This algorithm is activated in every exception thrown during communication with other nodes. This allows distributed detection of node failure.
- Request the previous node and next node parameters from the nameserver.
- Update the `next node` parameter of the previous node with the information received from the nameserver.
- Update the `previous node` parameter of the next node with the information received from the nameserver
- Remove the node form the Naming server.
- Test this algorithm by manually terminating a node (CTRL – C) and use a ping method as part of each node, that throws an exception when connection fails to a given node.

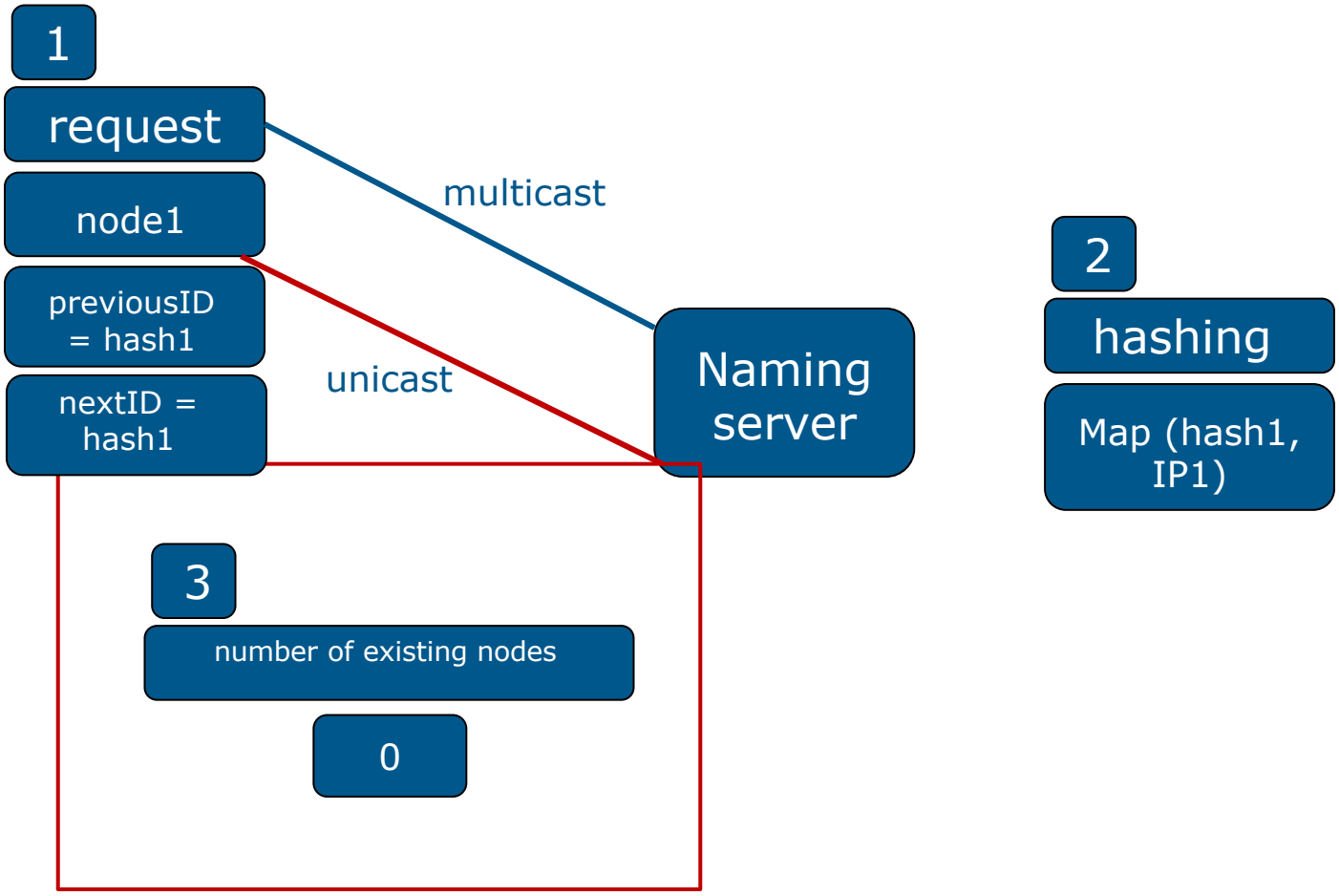
Bootstrap and Discovery: example 1



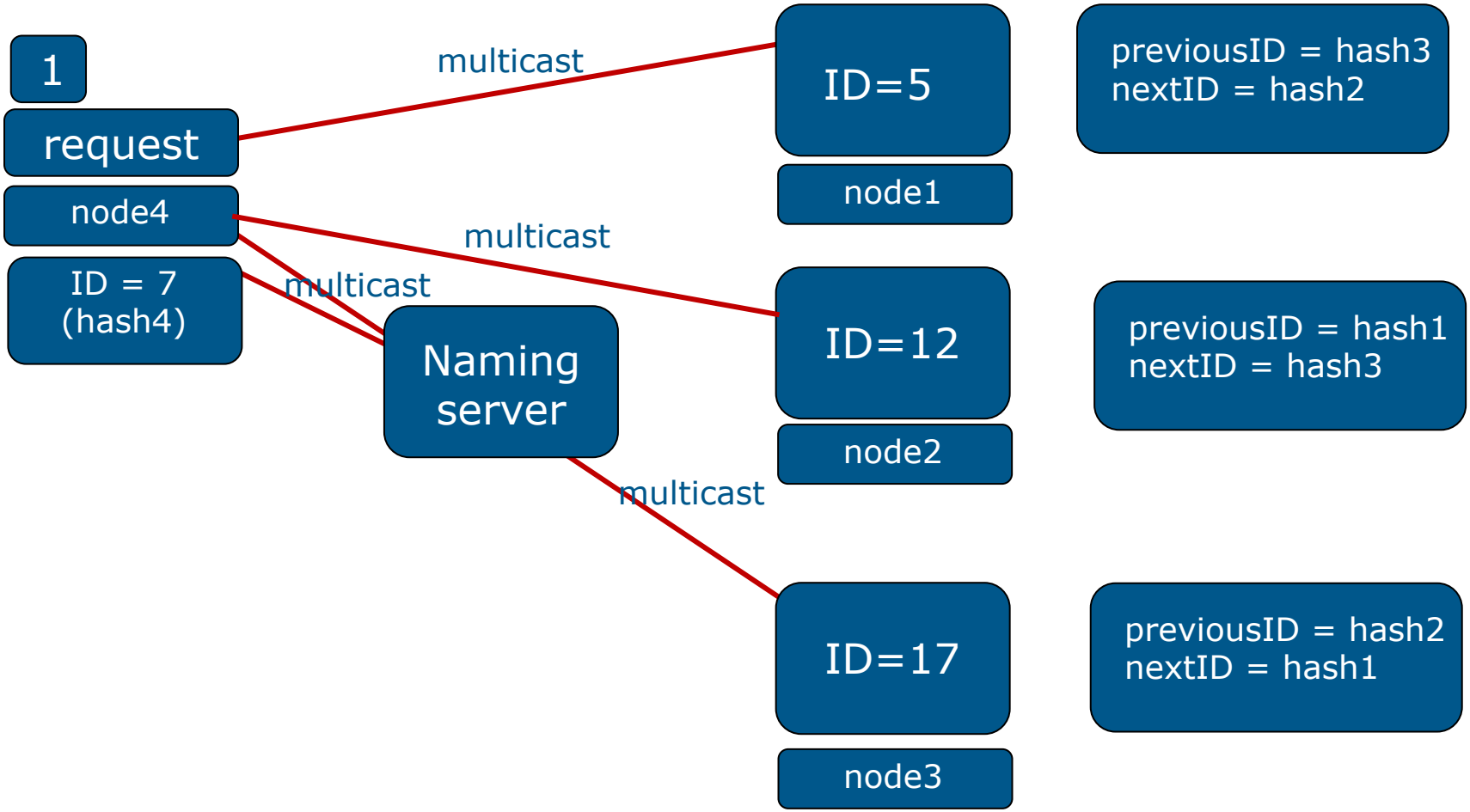
Bootstrap and Discovery: example 1



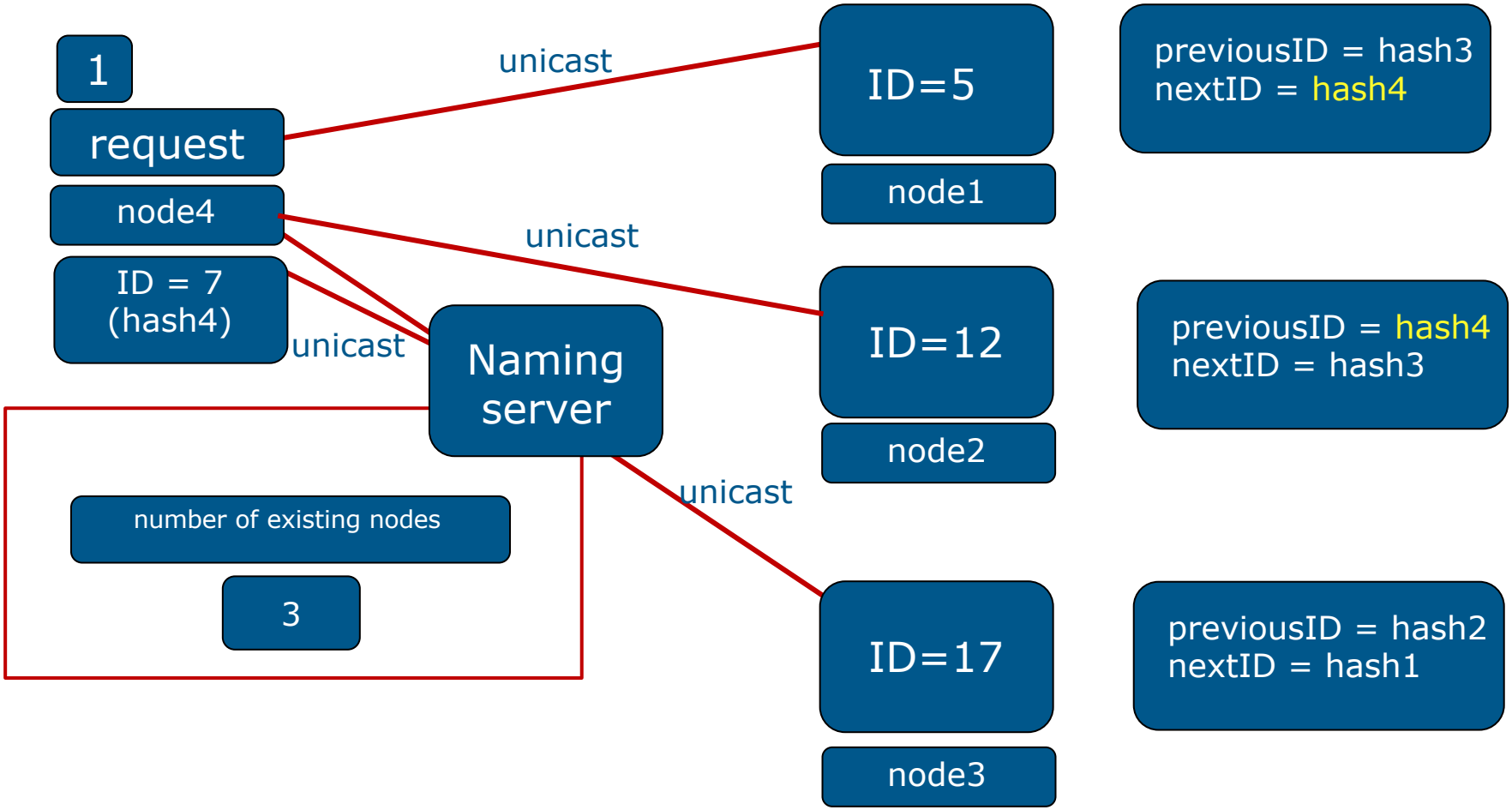
Bootstrap and Discovery: example 1



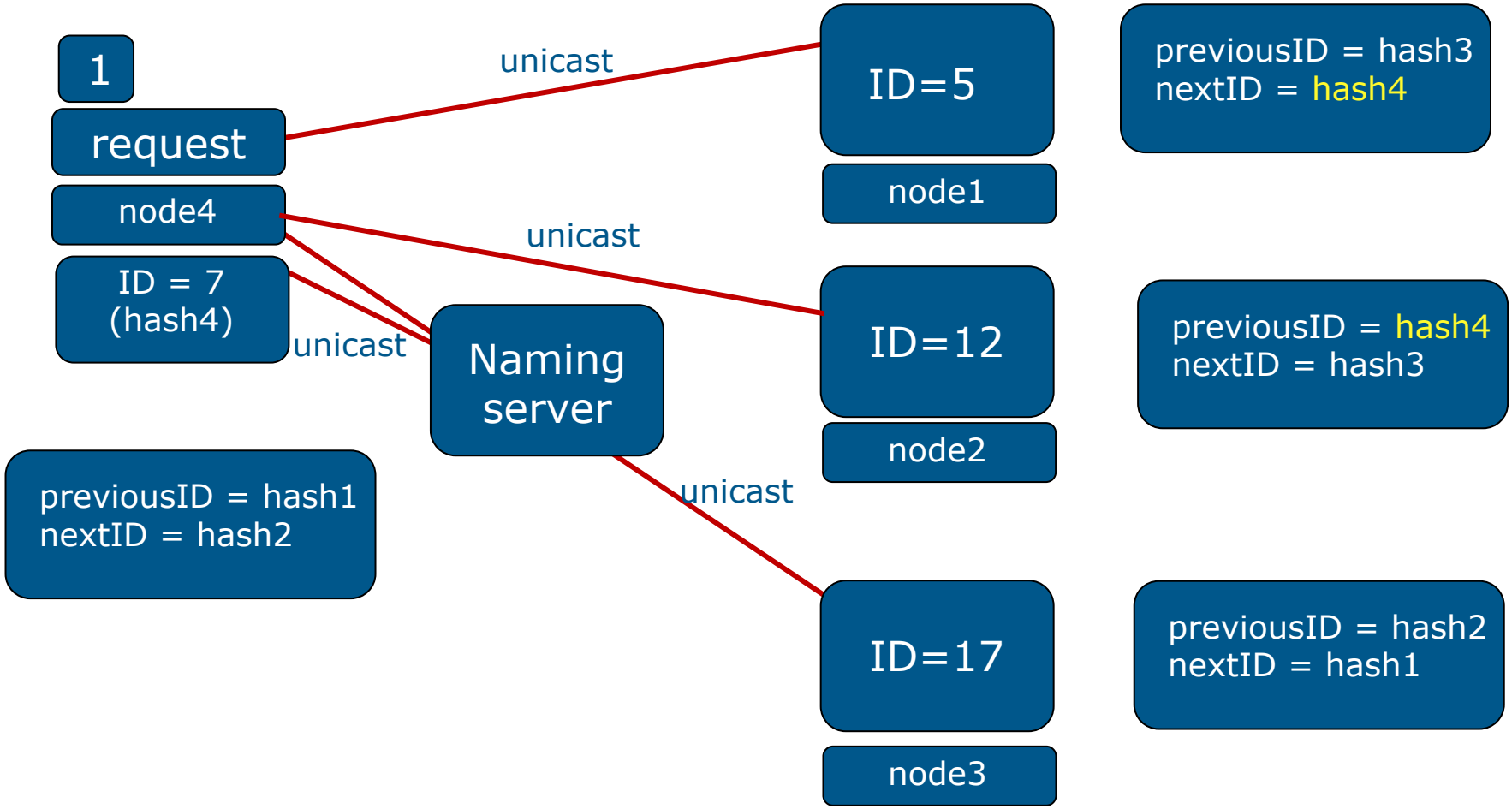
Bootstrap and Discovery: example 2



Bootstrap and Discovery: example 2



Bootstrap and Discovery: example 2



Bootstrap and Discovery: example 3