CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

LEADER ELECTION

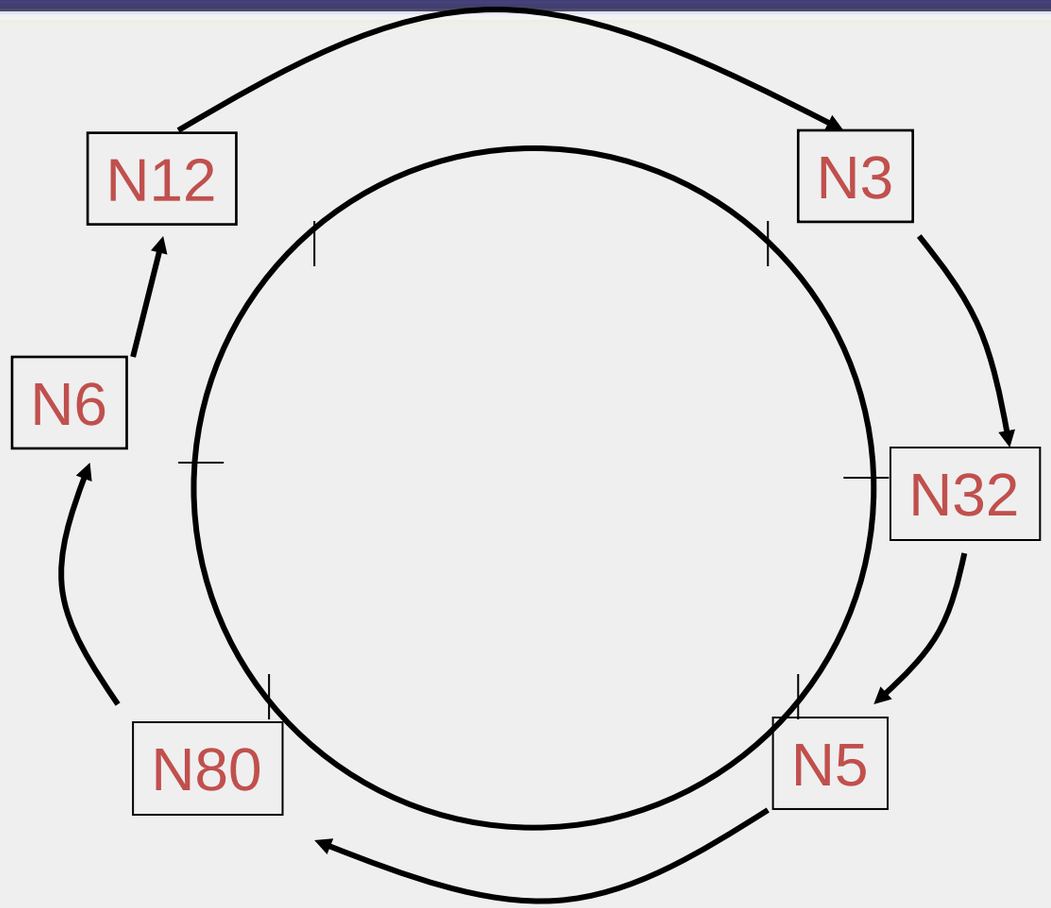
Lecture B

RING LEADER ELECTION

The Ring

- *N* processes are organized in a logical ring
  - Similar to ring in Chord p2p system
  - *i*-th process p_i has a communication channel to $p_{(i+1) \bmod N}$
  - All messages are sent clockwise around the ring.

The Ring



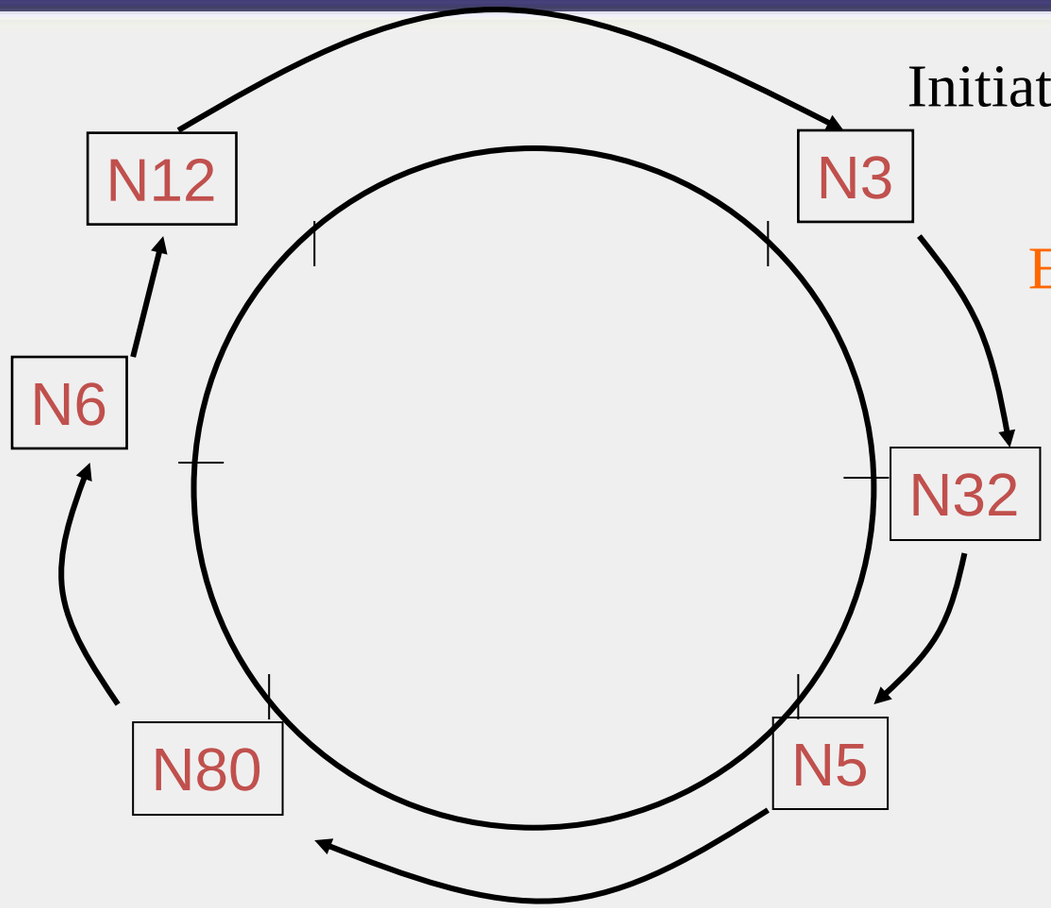
The Ring Election Protocol

- Any process p_i that discovers the old coordinator has failed initiates an "Election" message that contains p_i's own id:attr. This is the *initiator* of the election.
- When a process p_i receives an "Election" message, it compares the attr in the message with its own attr.
  - If the arrived attr is greater, p_i forwards the message.
  - If the arrived attr is smaller and p_i has not forwarded an election message earlier, it overwrites the message with its own id:attr, and forwards it.
  - If the arrived id:attr matches that of p_i, then p_i's attr must be the greatest (why?), and it becomes the new coordinator. This process then sends an "Elected" message to its neighbor with its id, announcing the election result.

The Ring Election Protocol (2)

- When a process p_i receives an “Elected” message, it
  - sets its variable $elected_i$ ← id of the message.
  - forwards the message unless it is the new coordinator.

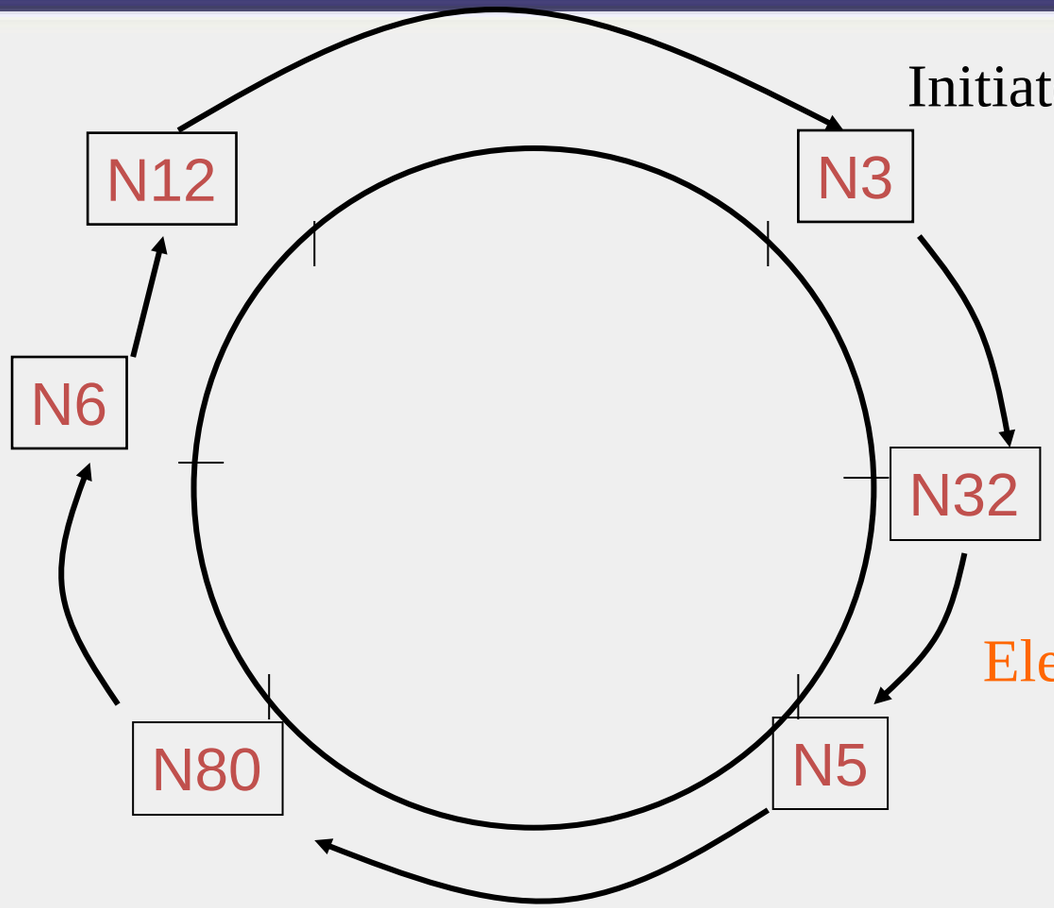
Ring Election: Example



Initiates the election

Election: 3

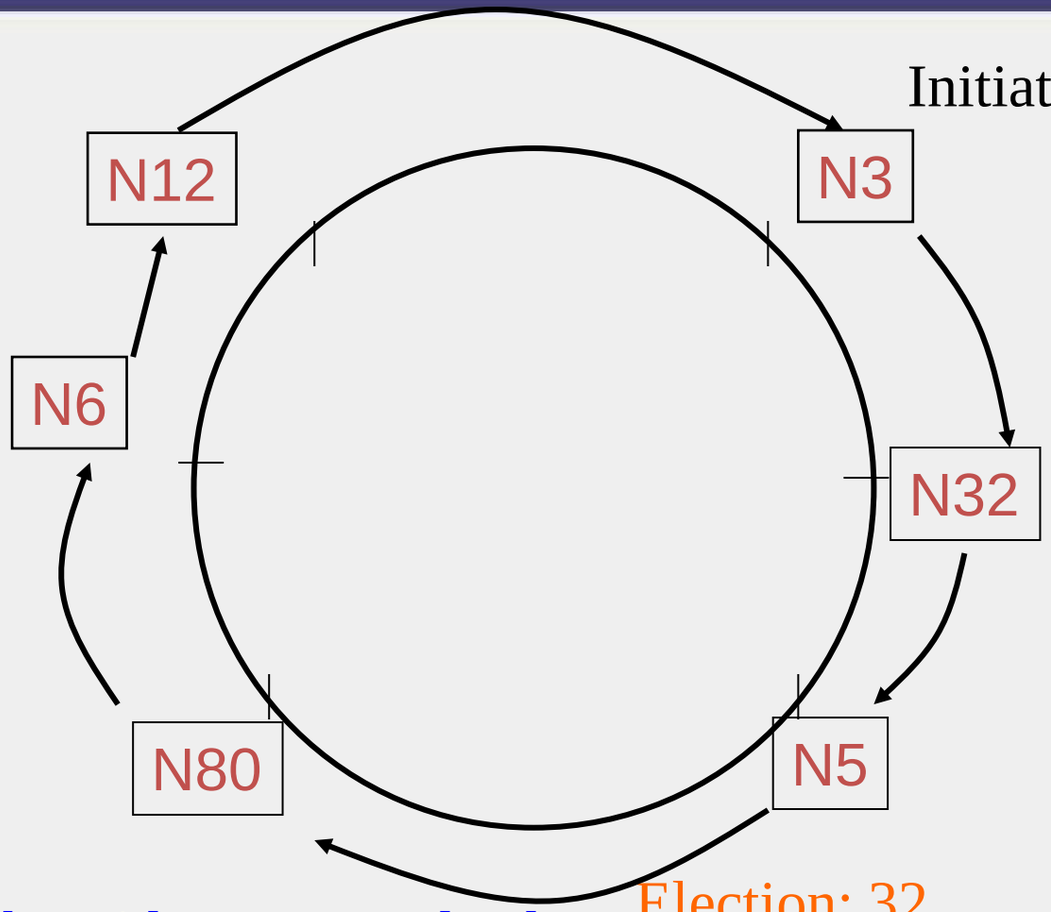
Goal: Elect highest id process as leader



Initiates the election

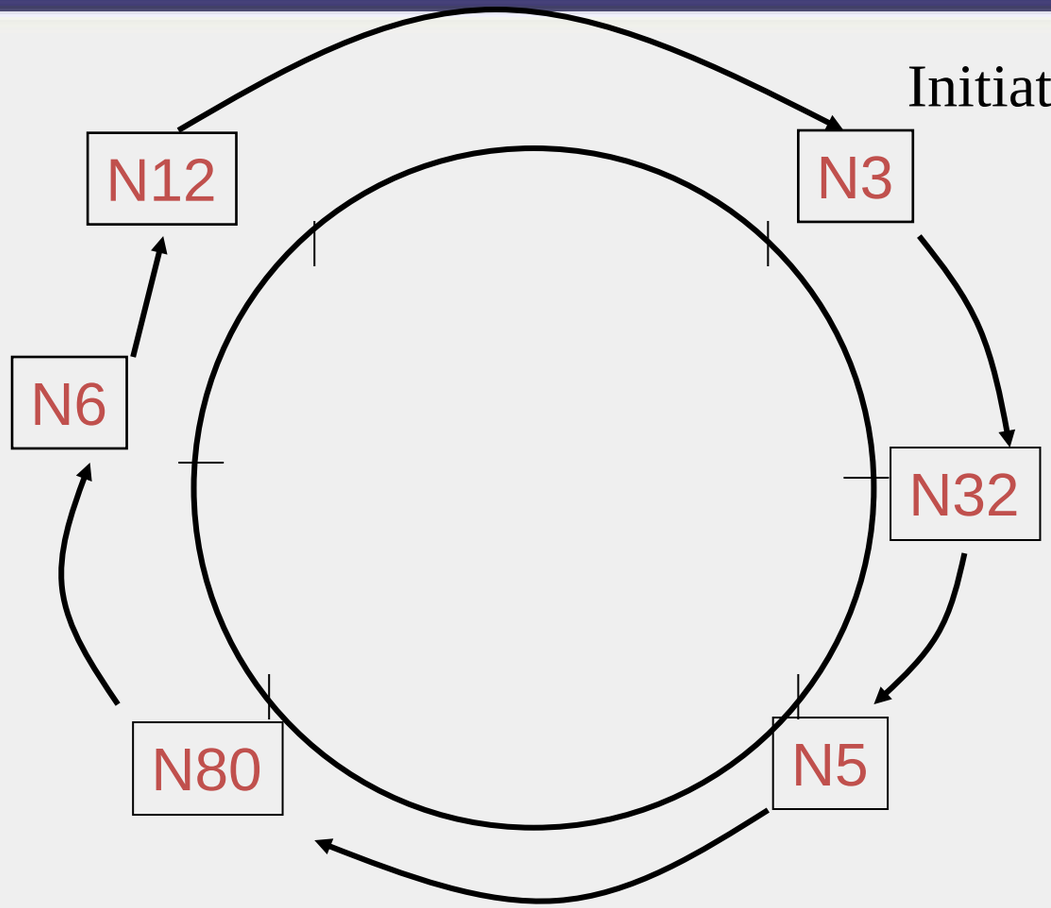
Election: 32

Goal: Elect highest id process as leader



Goal: Elect highest id process as leader

Initiates the election

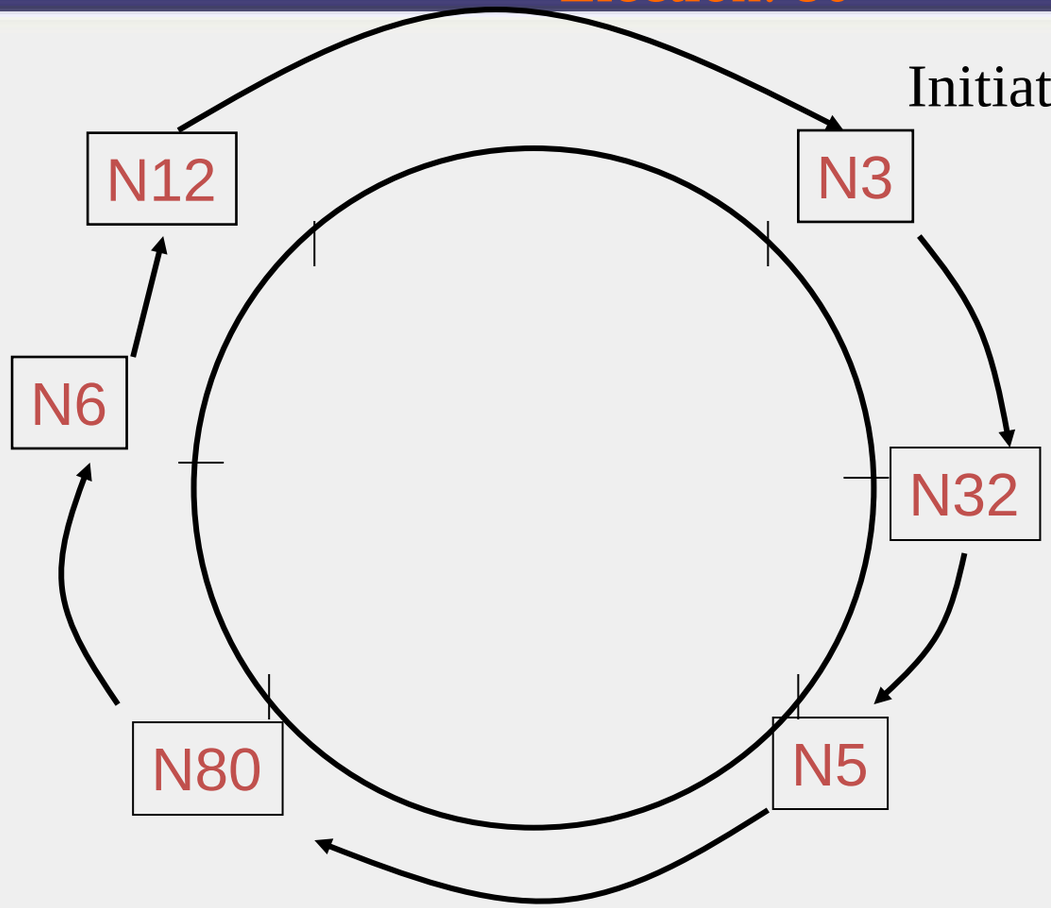


Election: 80

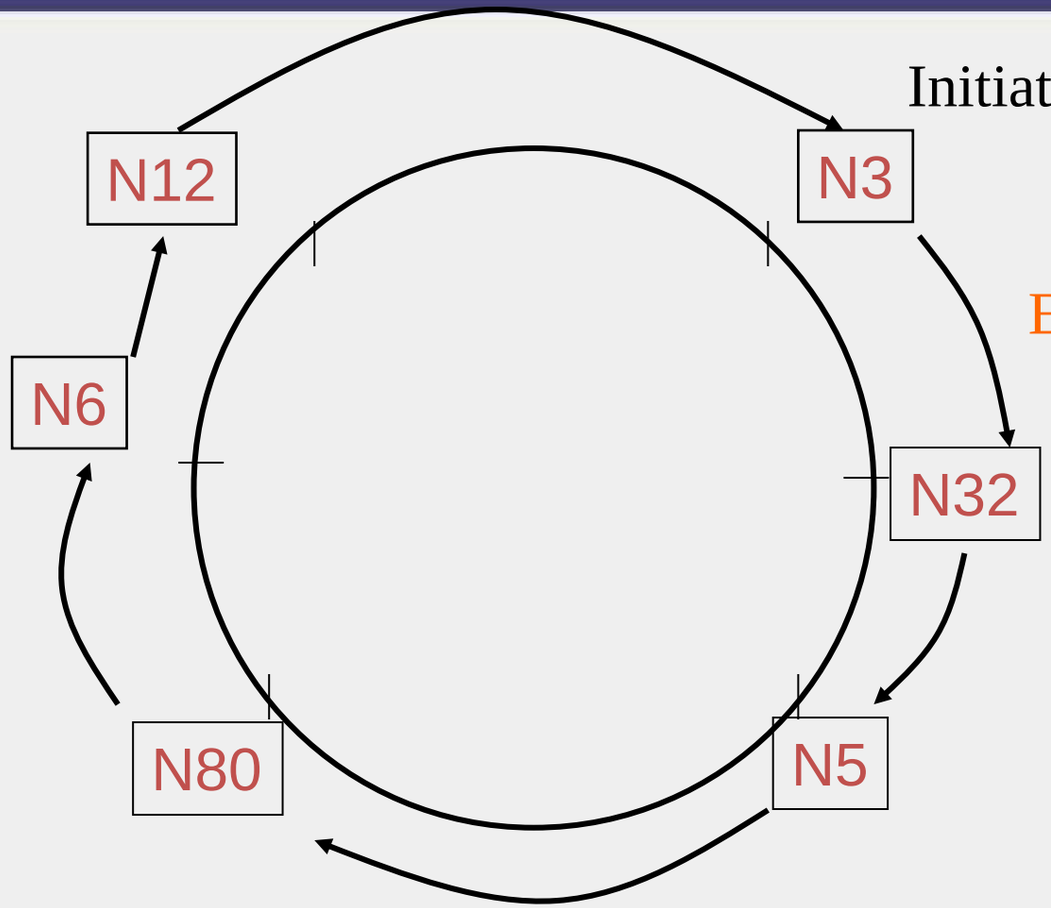
Goal: Elect highest id process as leader

Election: 80

Initiates the election



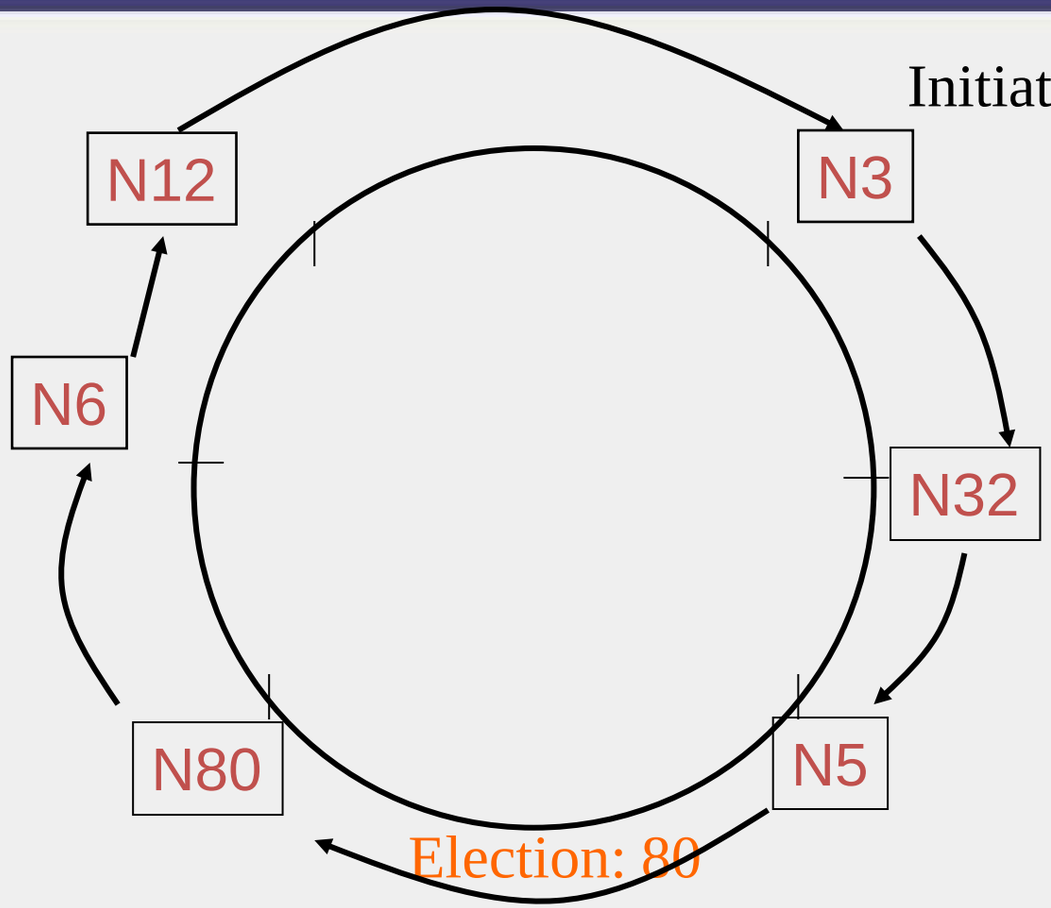
Goal: Elect highest id process as leader



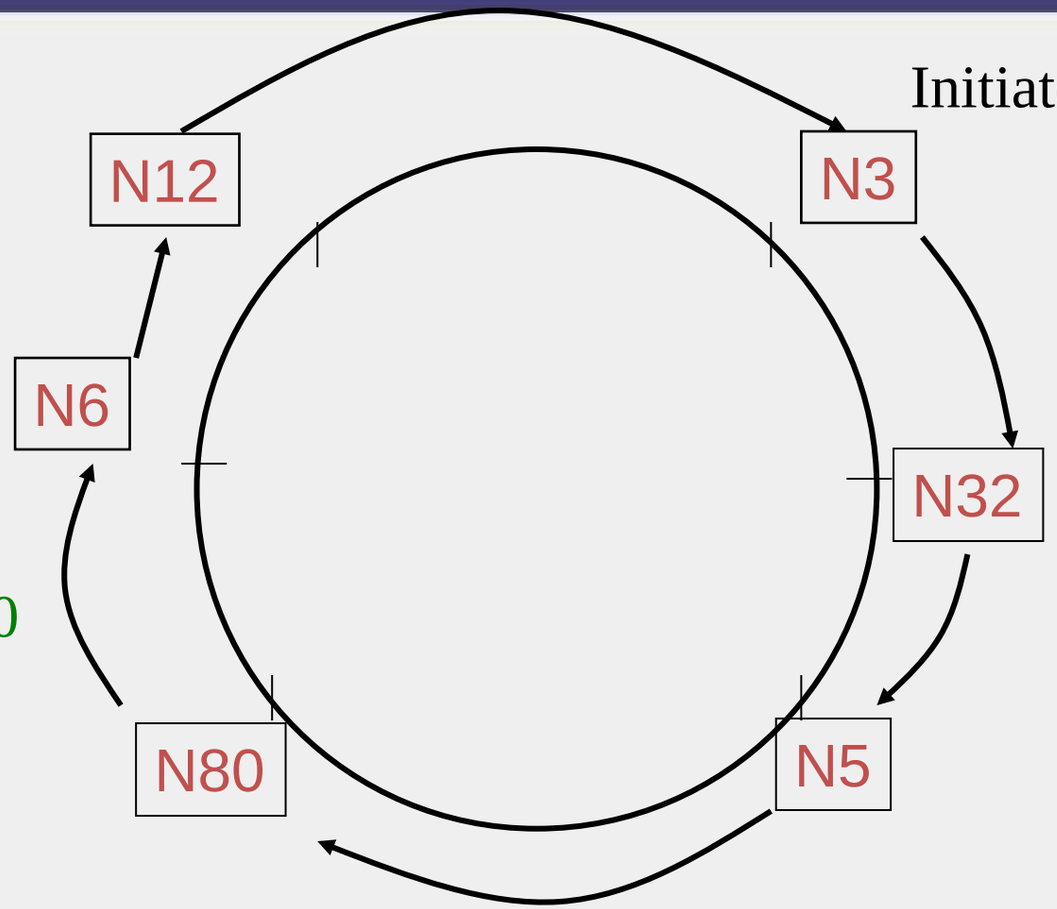
Initiates the election

Election: 80

Goal: Elect highest id process as leader

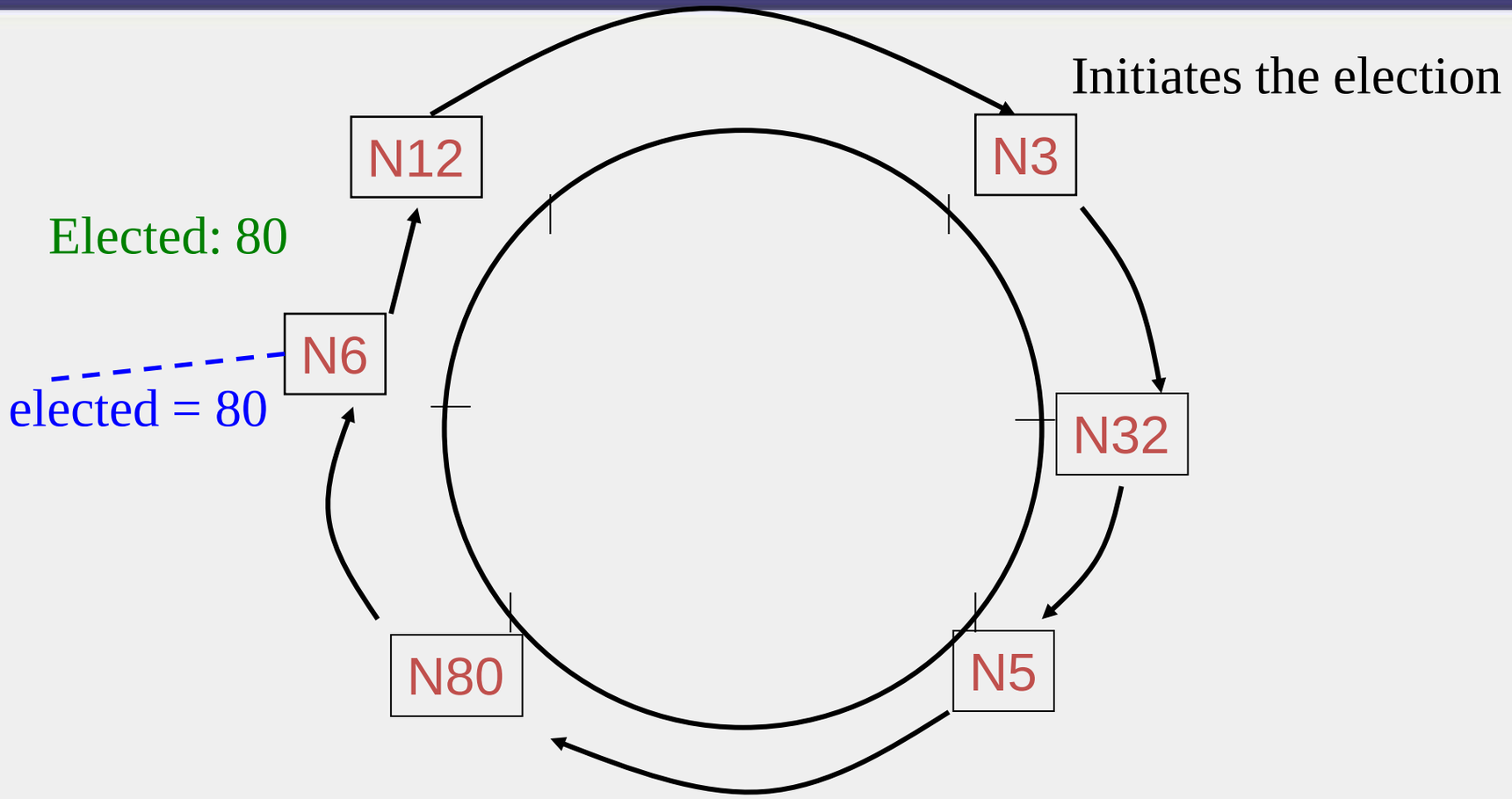


Goal: Elect highest id process as leader

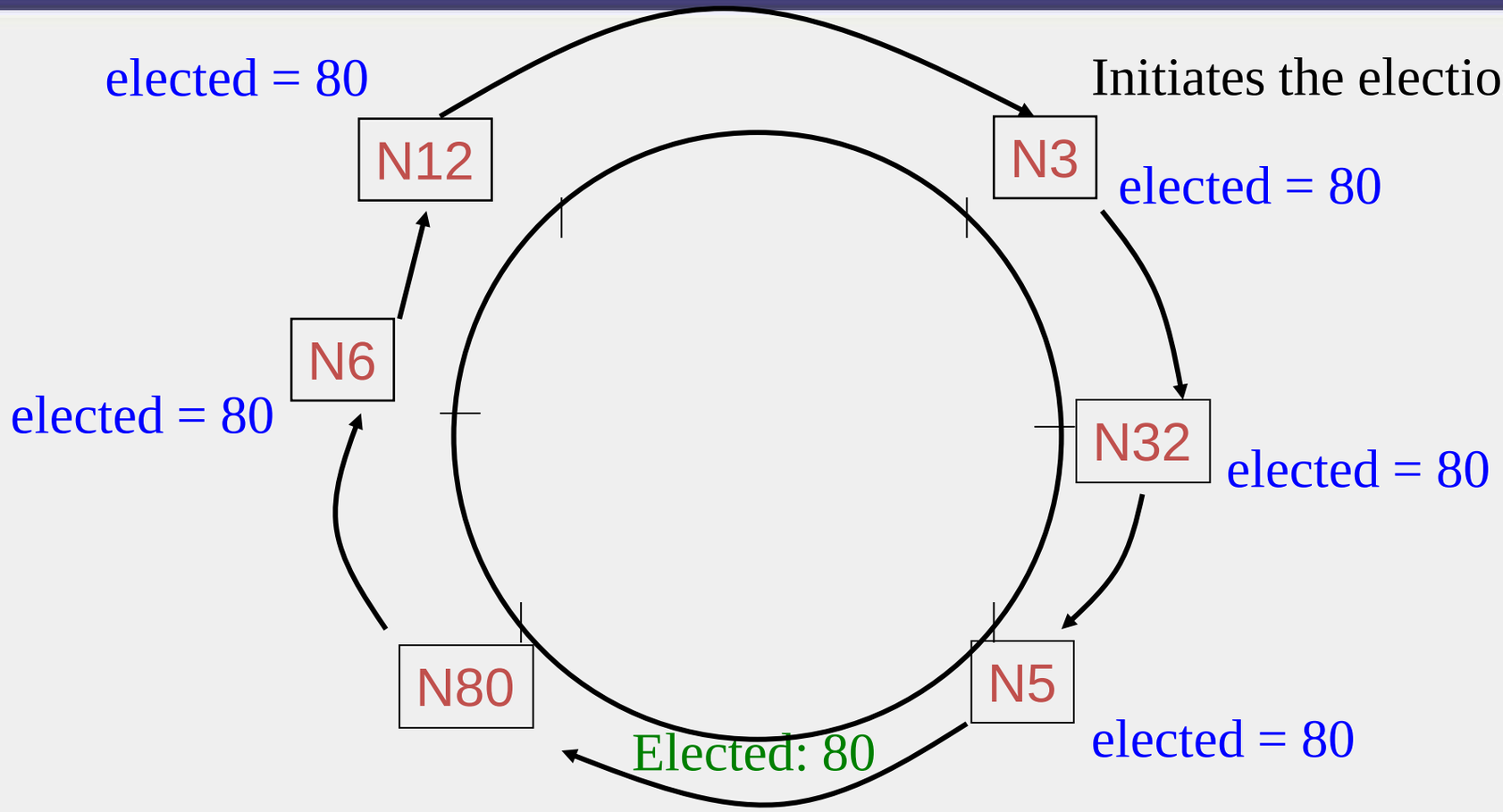


Elected: 80

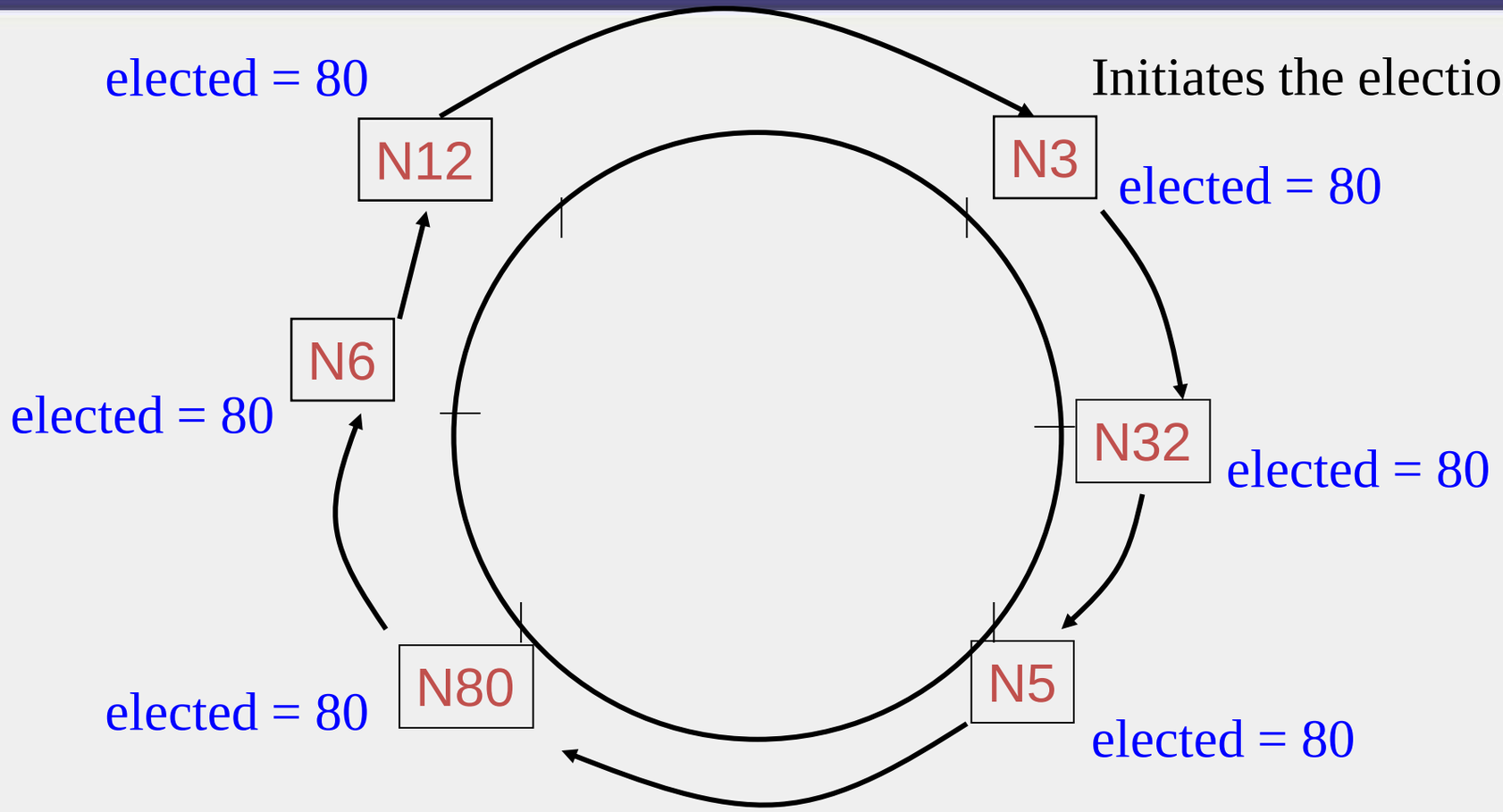
Goal: Elect highest id process as leader



Goal: Elect highest id process as leader



Goal: Elect highest id process as leader

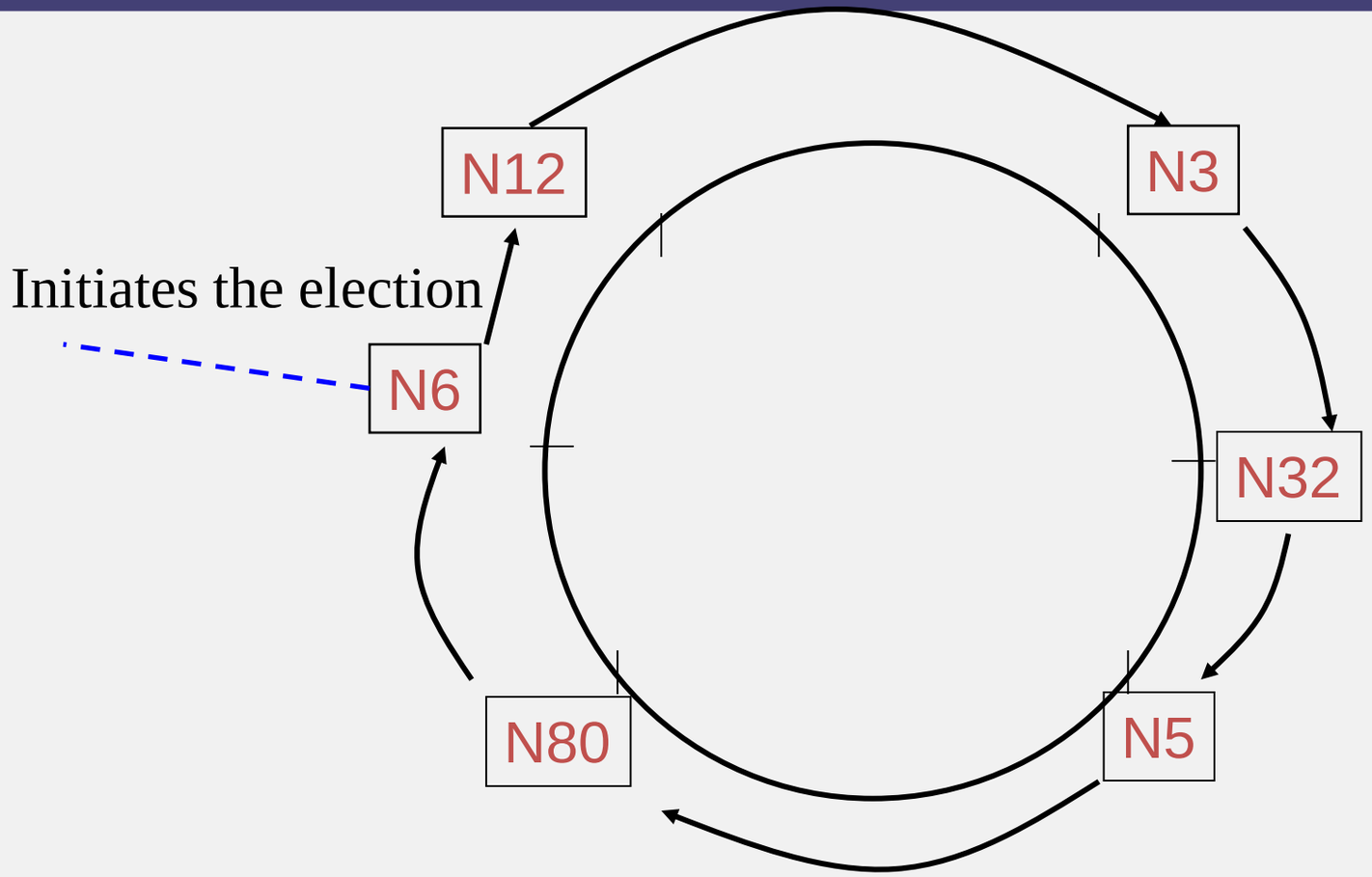


Goal: Elect highest id process as leader

Analysis

- Let’s assume no failures occur during the election protocol itself, and there are N processes
- How many messages?
- Worst case occurs when the initiator is the ring successor of the would-be leader

Worst-case



Goal: Elect highest id process as leader

Worst-case Analysis

- (*N-1*) messages for Election message to get from Initiator (N6) to would-be coordinator (N80)
- *N* messages for Election message to circulate around ring without message being changed
- *N* messages for Elected message to circulate around the ring
- Message complexity: (*3N-1*) messages
- Completion time: (*3N-1*) message transmission times
- Thus, if there are no failures, election terminates (liveness) and everyone knows about highest-attribute process as leader (safety)

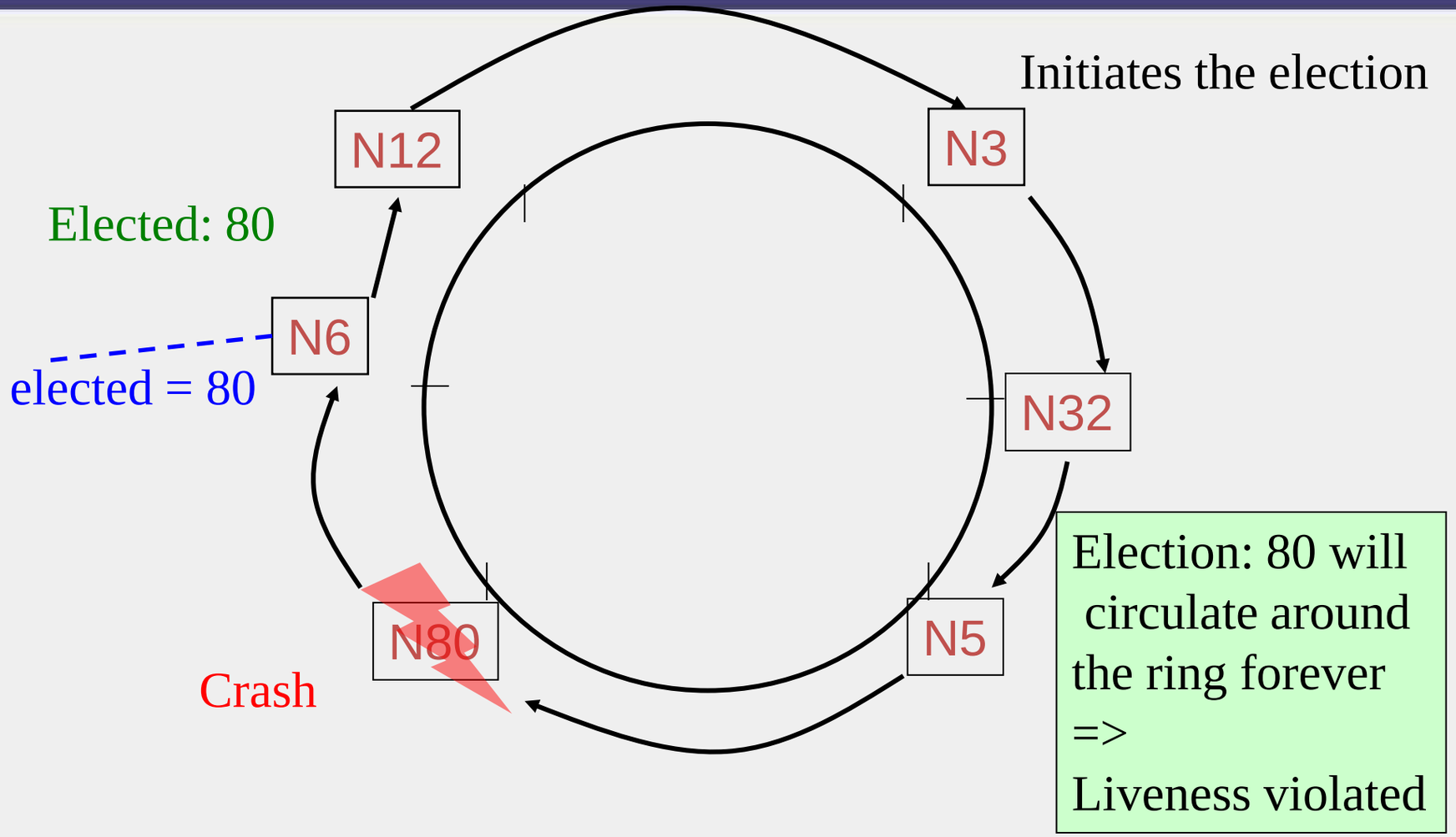
Best Case?

- Initiator is the would-be leader, i.e., N80 is the initiator
- Message complexity: *2N* messages
- Completion time: *2N* message transmission times

Multiple Initiators?

- Each process remembers in cache the initiator of each Election/Elected message it receives
- (All the time) Each process suppresses Election/Elected messages of any lower-id initiators
- Updates cache if receives higher-id initiator’s Election/Elected message
- Result is that only the highest-id initiator’s election run completes

Effect of Failures



Fixing for failures

- One option: have predecessor (or successor) of would-be leader N80 detect failure and start a new election run
  - May re-initiate election if
    - Receives an Election message but times out waiting for an Elected message
    - Or after receiving the Elected:80 message
  - But what if predecessor also fails?
  - And its predecessor also fails? (and so on)

Fixing for failures (2)

- Second option: use the failure detector
- Any process, after receiving Election:80 message, can detect failure of N80 via its own local failure detector
  - If so, start a new run of leader election
- But failure detectors may not be both complete and accurate
  - Incompleteness in FD => N80's failure might be missed => Violation of Safety
  - Inaccuracy in FD => N80 mistakenly detected as failed
    - => new election runs initiated forever
    - => Violation of Liveness

Why is Election so Hard?

- Because it is related to the consensus problem!
- If we could solve election, then we could solve consensus!
  - Elect a process, use its id's last bit as the consensus decision
- But since consensus is impossible in asynchronous systems, so is election!
- Next lecture: Can't we use Paxos?