Asynchronous ASMs

Echo Algorithm

Egon Börger

Dipartimento di Informatica, Universita di Pisa
http://www.di.unipi.it/~boerger

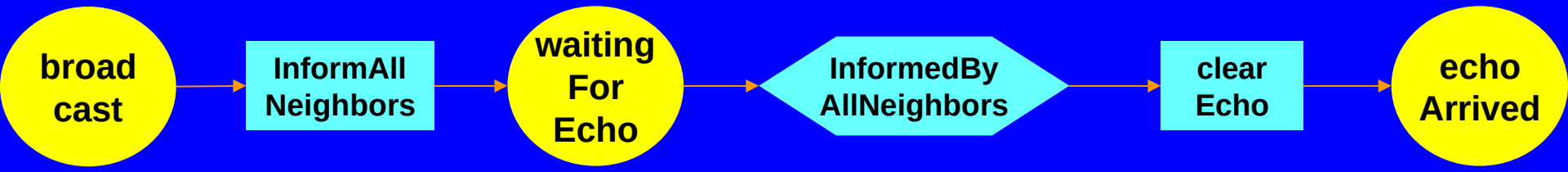
Echo Algorithm : problem statement

- Goal: Design a distributed algorithm that guarantees an initiator's message being broadcast and acknowledged ('echoed') through a connected network, using only communication between neighbors
- Algorithmic Idea:
  - the initiator (a distinguished node) broadcasts an info to all his neighbors and waits for their acknowledgements
  - every node which has been informed by some neighbor ('parent') node informs all his other neighbor nodes and waits for their acknowledgements to come back, to then forward his own acknowledgement back to the parent node

Echo ASM: Agent Signature

- Agent : connected graph of agents
  - initiator: Agent distinguished element with control states {broadcast, waitingForEcho, echoArrived} and special ASM
- Each agent is equipped with:
  - neighb $\subseteq$ Agent (external function)
  - informedBy : Agent $\rightarrow$ Bool (abstract message passing)
    - indicating whether a message (with info or acknowledgement) from a neighbor agent has been sent (arrived)
  - parent : Agent (building a spanning tree for acks)
    - yields a neighbor node who has sent a messg which is to be acknowledged, once all other neighbor nodes have acknowledged this mssg which has been forwarded to them
  - ctl_state:{listeningToInfo,waitingForAck,informed }
- Initially initiator in ctl_state = broadcast, for all other agents:
  - ctl_state = listeningToInfo, informedBy everywhere false, parent = undef

Initiator ASM for Echo algorithm



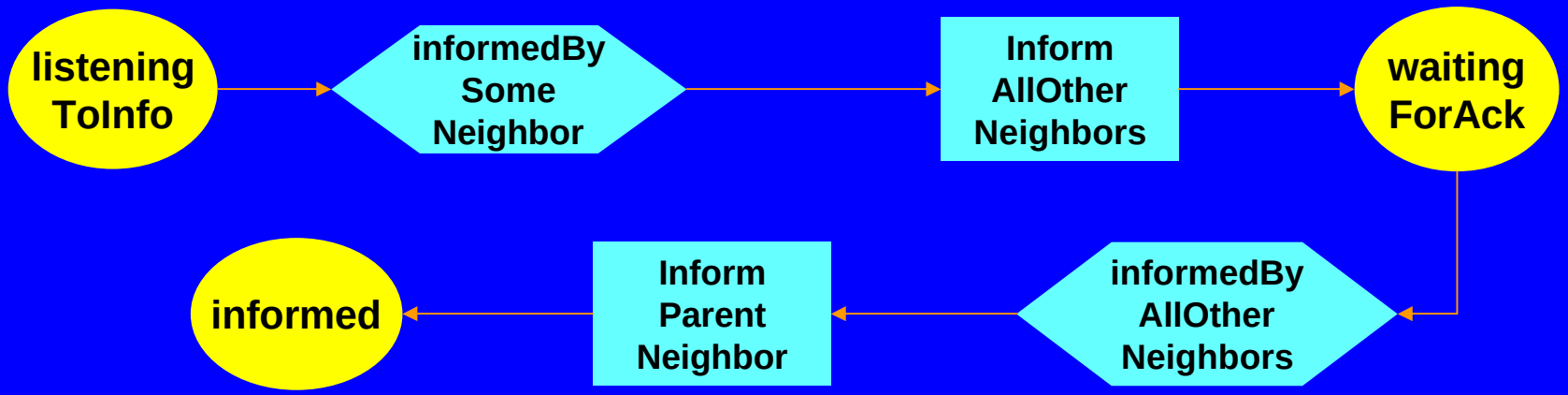
InformAllNeighbors ° **forall** u ∈ neighb u.informedBy(self):=true

InformedByAllNeighbors ° forall u ∈ neighb informedBy(u) = true

clearEcho ° forall u ∈ neighb informedBy(u) : = false

NB. Clear info when it gets redundant to prepare for later iterations

Echo Agent ASM for Agent ≠ Initiator



informedBySomeNeighbor ° for some u ∈ neighb informedBy(u) = true

InformAllOtherNeighbors °
choose u ∈ neighb **with** informedBy(u) **in**
forall v ∈ neighb-{u} v.informedBy(self) := true
informedBy(u):= false parent := u

informedByAllOtherNeighbors °
forall v ∈ neighb-{parent}
informedBy(v) = true

InformParentNeighbor °
parent.informedBy(self) := true
clearAcknowledgement

clearAcknowledgement °
forall u∈ neighb - {parent} informedBy(u) : = false
parent:= undef

Echo ASM (initiator and agents): Correctness

- Proposition: In every run of a set of agents including an initiator, all equipped with the corresponding echo ASMs:
  - the initiator terminates (termination)
  - the initiator terminates only when all other agents have been informed about its originally sent mssg (correctness)
- Proof. Follows by run induction from two lemmas.

Echo ASM (initiator and agents): Correctness

- Lemma 1. In every echo ASM run, each time an agent executes InformAllOtherNeighbors, in the spanning tree of agents waitingForAck the distance to the initiator grows until leafs are reached.
  - Proof. By downward induction on echo ASM runs
- Lemma 2. In every echo ASM run, each time an agent executes InformParentNeighbor, in the spanning tree the distance to the initiator of nodes with a subtree of informed agents shrinks, until the initiator is reached.
  - Proof. By upward induction on echo ASM runs

Exercise

- Refine the echo ASM to the case of a network with more than one initiator.
  - Hint: Use a pure data refinement, recording the initiator's identity when informing neighbors and letting the initiator wait for the echo to is own initiative.

Reference

- W.Reisig: Elements of Distributed Algorithms
  Springer-Verlag 1998
  - See the definition of a Petri net in Section 33 and a detailed termination and correctness proof in Section 77
- E. Börger, R. Stärk: Abstract State Machines. A Method for High-Level System Design and Analysis Springer-Verlag 2003, see http://www.di.unipi.it/AsmBook
  - See Chapter 6.1