Fast Asynchronous Agreement Protocols

TECHNISCHE UNIVERSITÄT DARMSTADT

Dan Dobre and Neeraj Suri

Department of Computer Science - TU Darmstadt, Germany

www.deeds.informatik.tu-darmstadt.de

DEEDS Group

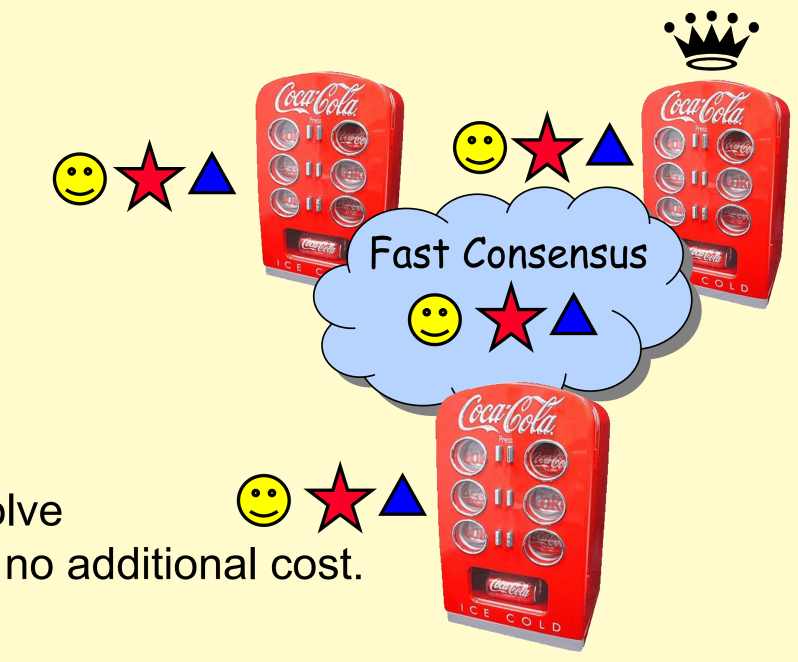
Objective

Aim of the Project is to develop efficient asynchronous distributed agreement protocols that are able to switch between "fast" and "conservative" executions depending on the number of actual faulty nodes and the current network conditions.

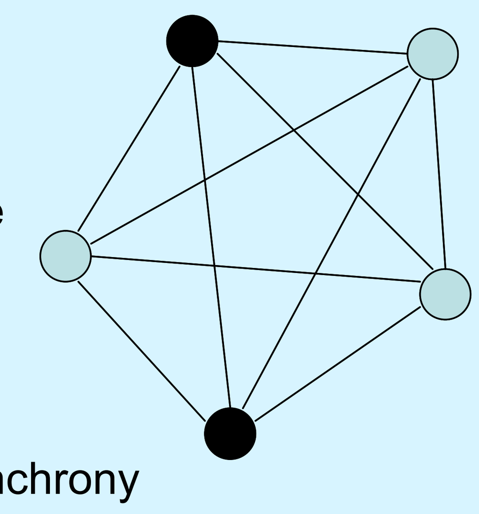
State machine approach
- + Input: a sequence of commands
- + Output: sequence of outputs and states

Fault Tolerance
- + Replicate the state machine
- + Execute **consensus**

We take advantage of spontaneous ordering to solve consensus in one round of message exchange at no additional cost.



System Model



- + Network of processors that synchronize by message passing
- + No assumption on message delays or relative processing time
- + Processors can fail by crashing ($f < n$)

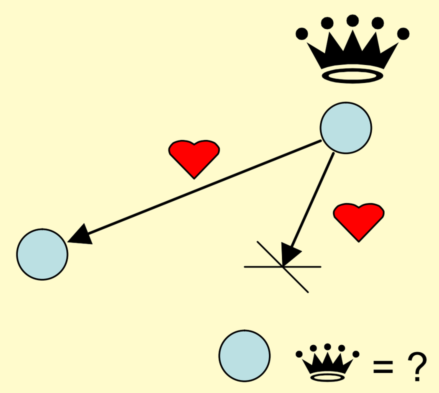
Asynchrony + crashes => consensus has no solution

But: Real systems alternate between periods of synchrony/asynchrony

Solving consensus requires:
1) A majority of correct processes, i.e., $f < n/2$ (safety part)
2) Enough synchrony for a sufficiently long time (liveness part)

Definitions

- + *Consensus:* processes need to choose a single value. Only a proposed value can be chosen.
- + *Atomic broadcast* guarantees that all processes deliver the same sequence of messages. It is equivalent to consensus.

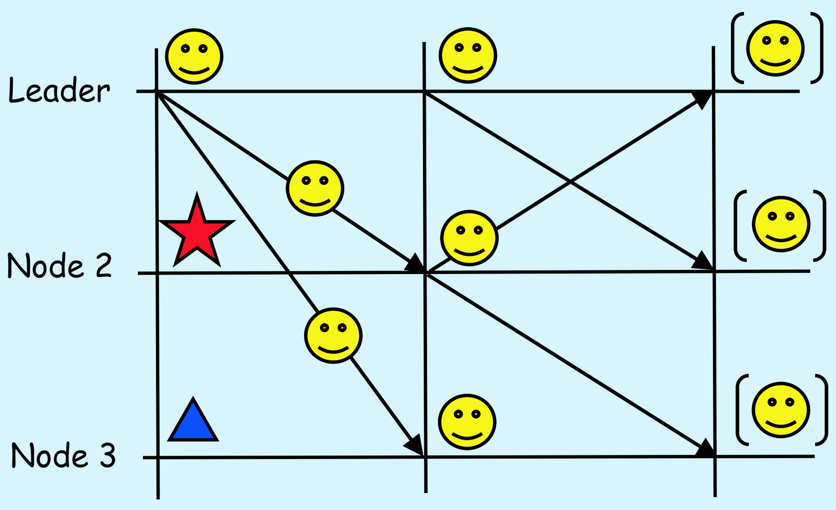


- + Failure detectors Ω and ◊P encapsulate the amount of synchrony necessary and sufficient to solve the above problems. They are defined by time-free properties:

Ω eventually outputs a single correct process (leader)

◊P eventually outputs exactly the crashed processes

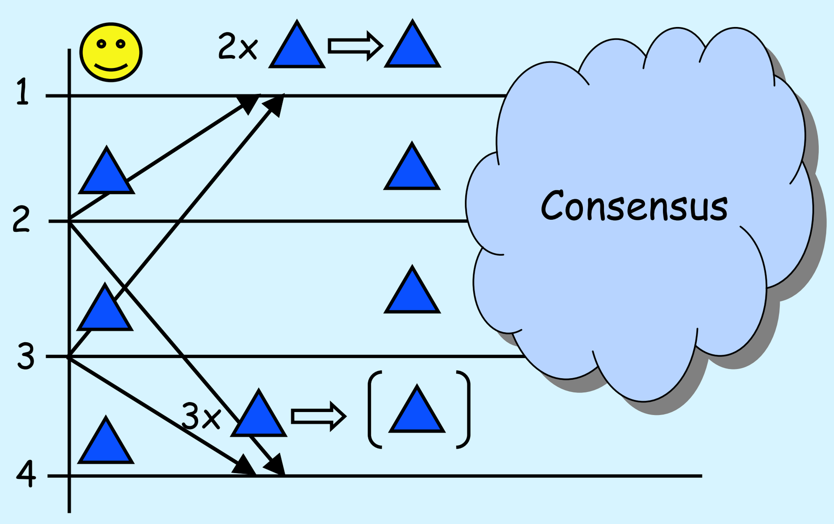
Context of Study



Question:
Is there any solution that switches between obtaining consensus in one and two communication steps?

2-step decision:
Consensus is obtained in two communication steps in every stable run.

1-step decision:
When all processes propose the same value one communication step is sufficient.



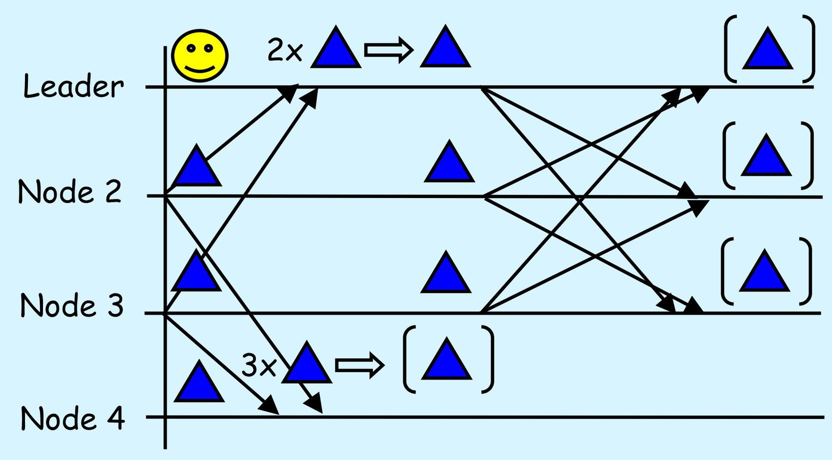
Results

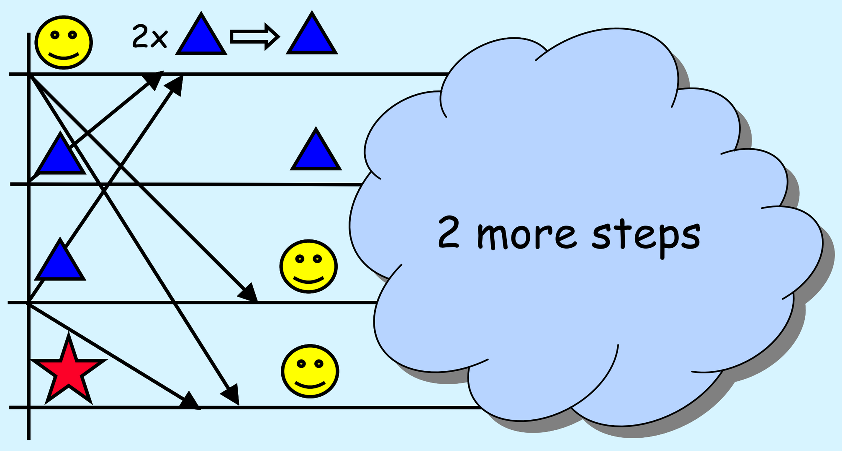
- + **Main idea**: *merge* 1-step consensus and the first round of regular consensus. (One might think that a simple algorithm that switches between the two algorithms above does the job)
- + **Tight bounds** for 1-step consensus time complexity:
  - Every Ω-based solution needs three communication steps. (Implies that the naive algorithm above does not work)
  - Every ◊P-based solution needs no more than two communication steps. (Indirect proof that Ω is strictly weaker than ◊P)
- + **Weak 1-step decision** to circumvent the lower bound:
  - When all processes including the leader propose the same value 1 communication step is sufficient.
  - Every Ω -based solution needs no more than two communication steps.

A Lower Bound

Every Ω-based protocol that satifies 1-step decision needs otherwise three communication steps.

The intuition:
1) If some process decides X then every process must accept X (locking) such that no other value can be decided
2) If no process decides then every process accepts the leader value Y
3) Processes have inconsistent perceptions of 1) and 2) and X ≠ Y





Circumventing the Lower Bound

We identify two necessary conditions for the above lower bound to hold:
(**C1**) leader proposal ≠ decision value
(**C2**) processes read from different quorums

Idea: If we invalidate either C1 or C2 then we obtain the desired property.
¬ C1 => weak 1-step decision
¬ C2 => ◊P

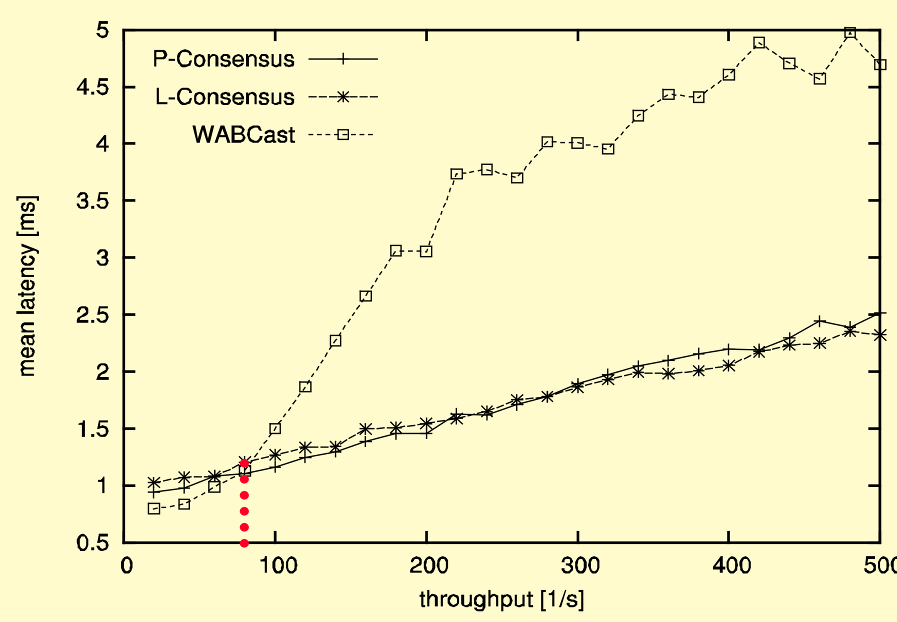
Performance Evaluation

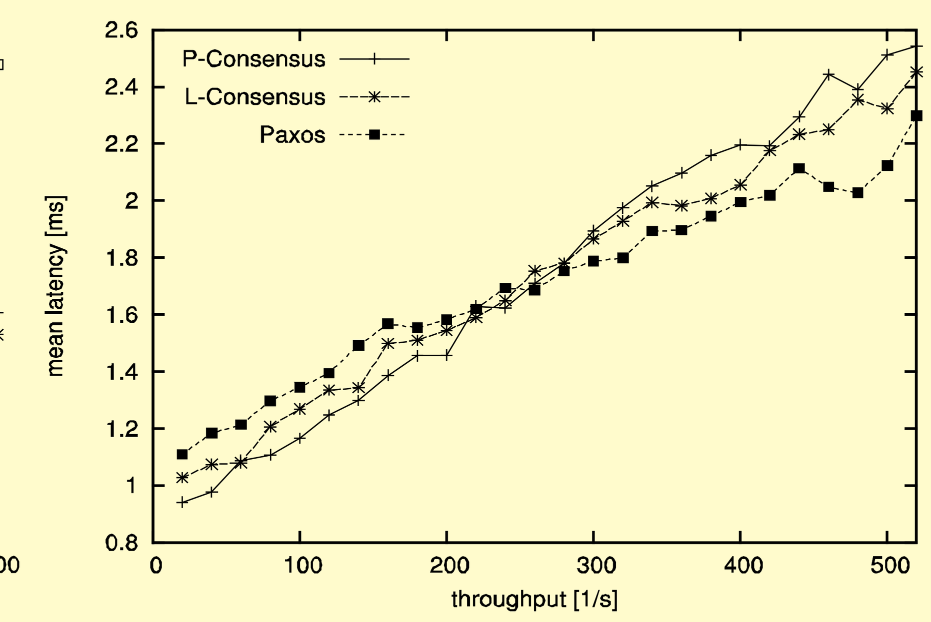
The metrics of interest are summarized in the table below

Protocol	Latency	#p-t-p msgs	Resilience	FD
	with ; without spontaneous ordering			
Paxos (centralized)	3 (4)	$n^2 + n + 1$ (3n)	$f < n/2$	Ω
WABCast	2 ; unbounded	$n^2 + n$; unbounded	$f < n/3$	WAB
LC/PC	2 ; 3	$n^2 + n$; $2n^2 + n$		WAB+Ω/◊P

Experimental Setup

- + Debian Linux Workstations (1GB MM, 2GHz CPU) interconnected by 100Mb LAN
- + Implementation is based on the Neko framework
- + (f = 1) => Paxos needs 3 nodes, our protocols and WABCast need 4
- + Throughput varies from 20 msg/sec to 500 msg/sec
- + We measure the latency of atomic broadcast as a function of throughput





Future Work

+ The generality of a protocol is given by its ability to adapt to different network conditions. We are currently working on fast switching agreement protocols in WANs.

+ Modelling all faults as crashes/byzantine is overoptimistic/overpessimistic. How fast can consensus be obtained in the hybrid fault model?

+ Most leader selection mechanisms are based on binary assesment of the state faulty or nonfaulty of a node. At the cost of a more sophisticated monitoring and assesment process, QoS-based leader election promises faster consensus rounds.

Publications

One Step Consensus with Zero Degradation - *Dan Dobre and Neeraj Suri*
International Conference on Dependable Systems and Networks (DSN) 2006. (to appear)

Acknowledgements

This work is partly funded by Microsoft Research through the MSR Cambrige European PhD Scholarship Award.

Dan Dobre
dan@informatik.tu-darmstadt.de — tel: +49 6151 16 5657
www.deeds.informatik.tu-darmstadt.de/dan — fax: +49 6151 16 4310
Prof. Neeraj Suri
suri@informatik.tu-darmstadt.de — tel: +49 6151 16 3513
www.deeds.informatik.tu-darmstadt.de/suri — fax: +49 6151 16 4310