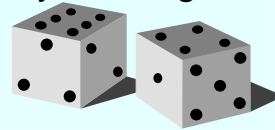


## A Randomize Protocol for Consensus

$n$  - total number of processes.  
 $f$  - total number of faulty processes.  
Assumption:  $n > 9f$ .



This algorithm solves a more complex problem where the failure model is **Byzantine**, i.e. the failed processes can send arbitrary messages to arbitrary processes (may lie), or may fail.



## The protocol (Ben-Or)

Iteration=0;  $x$  = initial value

Do **Forever**:

    Iteration = Iteration + 1

    Step 1

    Step 2



Step 1:

    Broadcast **Proposal**(Iteration, $x$ )

    wait for  $n-f$  messages of type **Proposal**(Iteration, $*$ )

    if at least  $n-2f$  messages have the same value  $v$

        then  $x = v$       (that value)

        else  $x = \text{undefined}$

## The Protocol (cont.)

Step 2:

Broadcast  $\text{Bid}(\text{Iteration}, x)$

wait for  $n-f$  messages of type  $\text{Bid}(\text{Iteration}, *)$

$v$  is the real value (0/1) occurring most often  
and  $m$  is the number of occurrences of  $v$

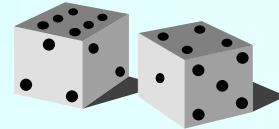
if  $m \geq n-2f$

then **Decide** ( $x=v$ )

else if  $m \geq n-4f$

then  $x = v$

else  $x = \text{random}$  (0 or 1)



Submission: Wednesday, March 12 at the beginning of class.  
Solution must be typed. No collaboration is allowed.

## Homework

1. Prove that the protocol is correct.  
i.e. that it satisfies the agreement, validity and termination (**with probability 1**) requirements.  
Termination means - **for reaching a decision**.  
**Assume at most  $f$  Byzantine failures.**

The communication is reliable FIFO broadcast, although messages from different processes can arrive at different order to different processes.

2. (**only 437**) Modify the algorithm so that eventually, all non faulty processes halt.