A Pareto Optimal Solution to Set Consensus

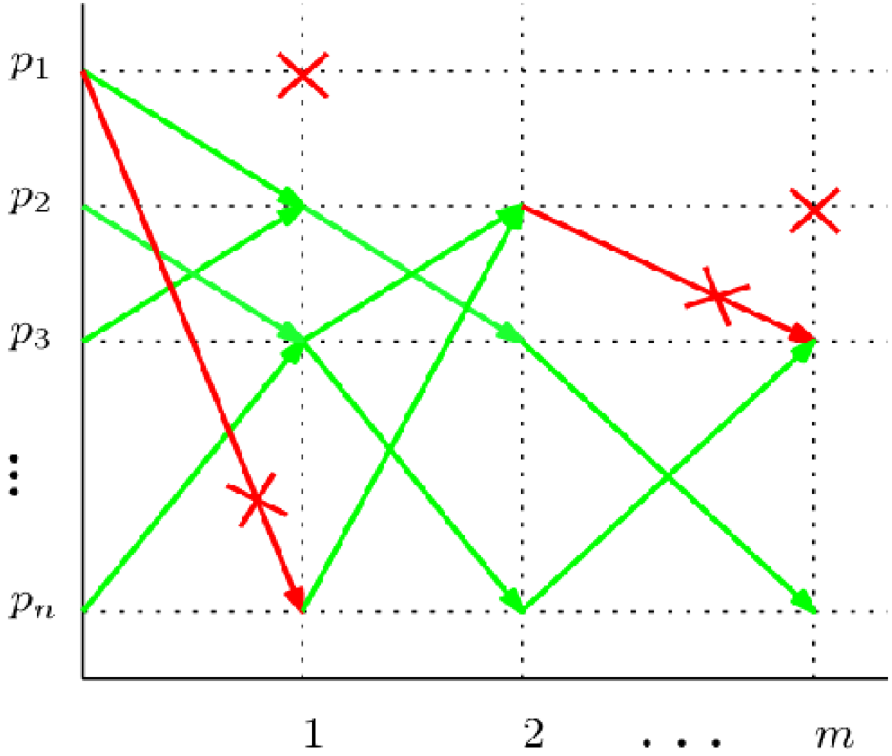
Armando Castañeda, Technion

Joint work with:

Yannai A. Gonczarowski, Hebrew U. of Jerusalem
Yoram Moses, Technion

Synchronous Message-Passing

- n sync. processes
- Synchronous rounds
- At most $t < n$ crash failures
- f = actual number of failures
- *Stopping* time ≠ *Decision* time



k-Set Consensus [Chaudhuri in 93]

- Generalization of the Consensus task
- Processes start with inputs from a domain $V = \{0, \ldots, k\}$
  - Termination: Each correct decides a value
  - *k*-Agreement: correct processes decide on at most *k* values
  - Validity: The decision of a process is the input of a process

Early Deciding Protocols

- Several k-Set Consensus protocols.

Several early deciding k-Set Consensus protocols

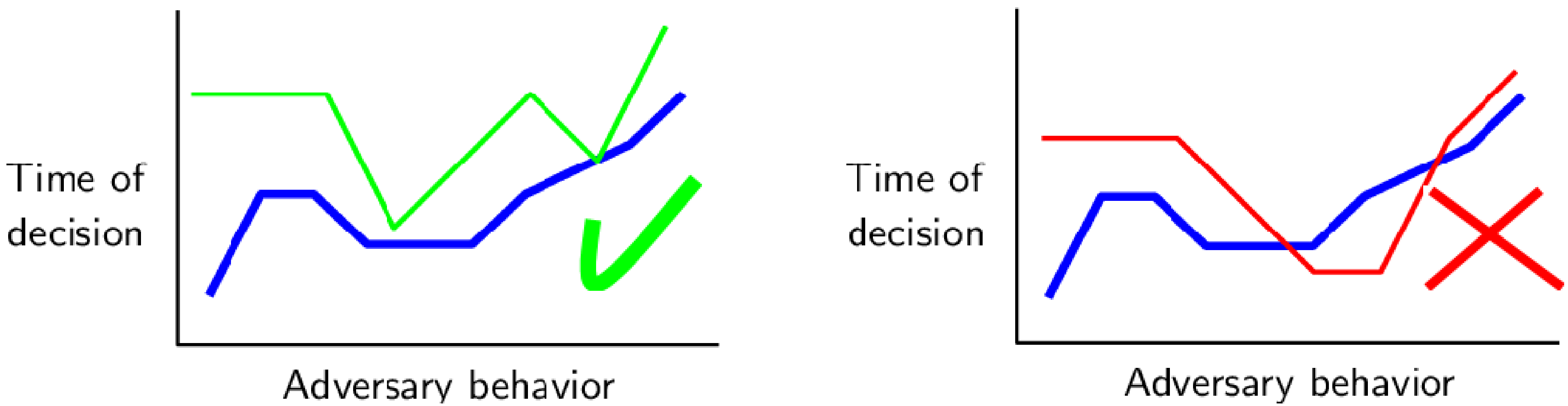
Which one is the best?

much earlier.

- Early deciding protocols: processes decide before the lower bound.

Comparing Protocols (1)

- *P* dominates *Q*, $P \leq Q$:



- *P* strictly dominates *Q*, $P < Q$: if $P \leq Q$ and a decision occurs strictly earlier in at least one case.

Comparing Protocols (2)

- Full-information protocols

Target: THE BEST protocol for *k*-Set Consensus

Impossible!! [Moses and Tuttle 88]

for every A, for every i, $P(A,i) \leq Q(A,i)$

- P strictly dominates Q, $P < Q$:

  $P \leq Q$ and there is A, there is i, $P(A,i) < Q(A,i)$

No All-Case Optimal Protocol (1)

- The case of Consensus (1-Set Consensus).

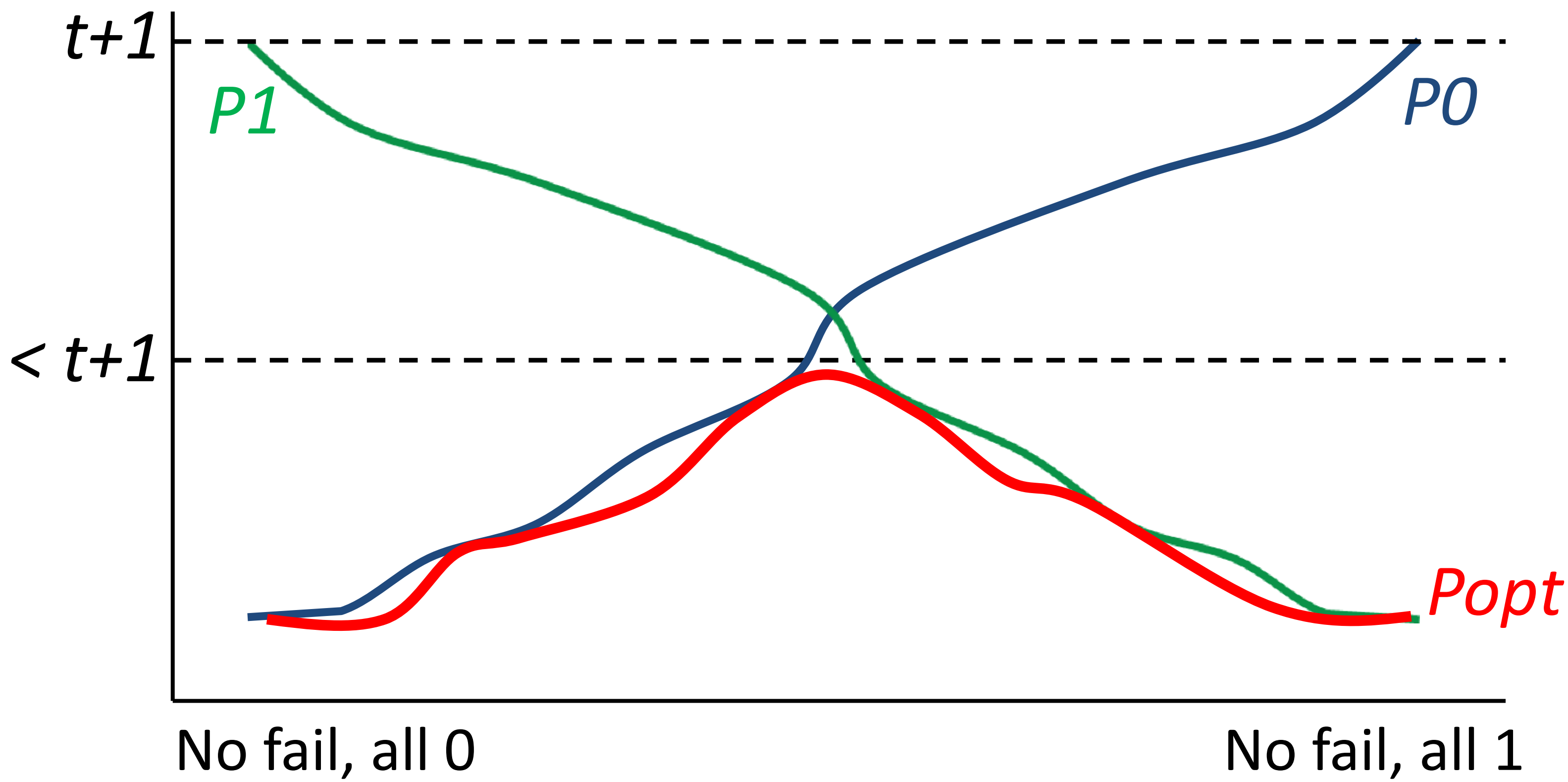
- Target: Dominates ALL Consensus protocols.

- Protocol P0:
  - A process decides 0 as soon it receives a 0.
  - Otherwise wait until round t+1 and decides 1.

- Protocol P1: similarly defined

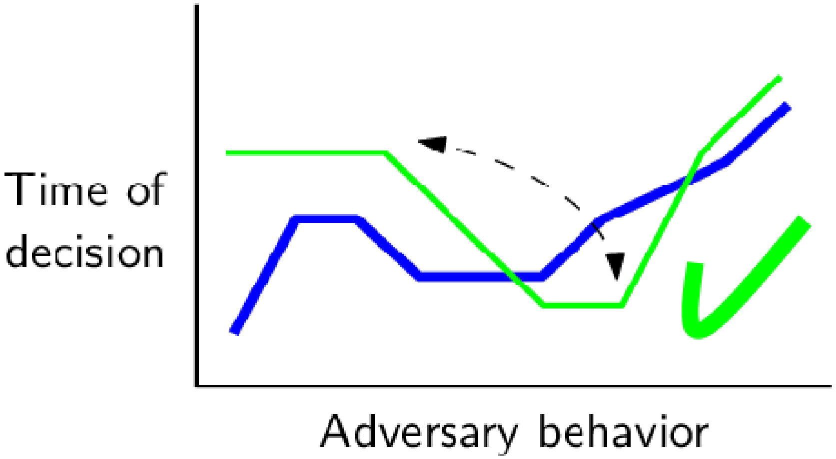
No All-Case Optimal Protocol (2)

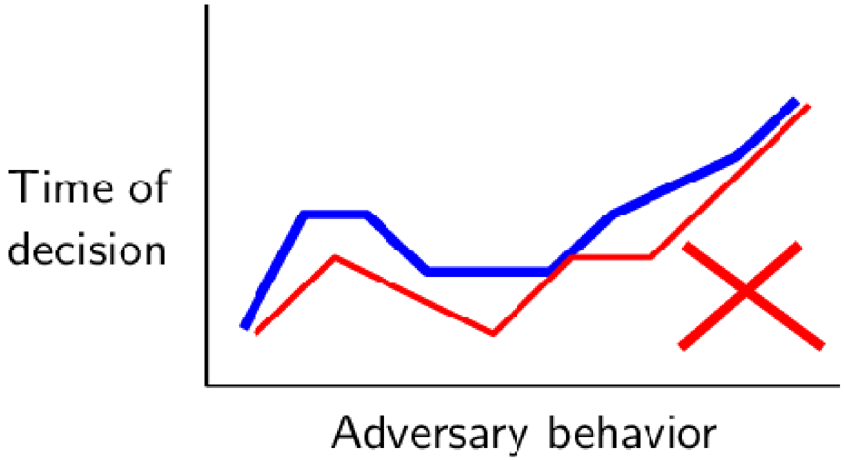
No All-Case Optimal Protocol (2)



Contradicts the *t+1* Consensus lower bound!!

Pareto Optimality (1)





- Improve at some point → Loss at another point
- *P* is Pareto optimal if for every *Q*, *not* $Q \leq P$ [Halpern et al. 2001]

Pareto Optimality (2)

- There exist Pareto optimal protocols for Consensus [Halpern et al. 2001]

- For every consensus protocol *P*, there is a Pareto Optimal consensus protocol *Q* that dominates *P*.

- Cumbersome construction.

Results (1)

- A Pareto Optimal Protocol to *k*-Set Consensus
- In executions with *f* failures:
  - Decision time: $f/k + 1$
  - Stopping time: $min(f/k + 2 , t/k + 1)$
- Pareto optimal → Cannot strictly be improved

Results (2)

- Our protocol strictly dominates all published *k*-Set Consensus Solutions [Chaudhuri et al. 2000, Gafni et al. 2011, Guerraoui and Pochon 2009, Halpern et al. 2001, Raipin Parvédy et al. 2005]

- Optimality proof: Knowledge-based analysis, NO reductions, NO topology

The Case of Consensus (1)

- Inputs $V = \{0,1\}$
- Protocol based in rules for each input value

- For process i (full-information):

  FOR round $r = 0, \ldots, t+1$ DO

  IF i is undecided THEN

  IF Rule0 THEN decide 0

  IF Rule1 THEN decide 1

The Case of Consensus (1)

- Rule0 = ∃0 = *i* receives a 0.

Processes decide 0 as soon as possible

<u>Target:</u> Decide 1 as soon as it is safe to decide 1

IF *i* is undecided THEN

IF Rule0 THEN decide 0

IF Rule1 THEN decide 1

The Rule1 (1)

- P = Consensus protocol, processes decide as soon as $\exists 0$
- **Lemma 1**. For every Consensus protocol $Q \leq P$, each process i decides 0 in Q as soon as $\exists 0$

The Rule1 (1)

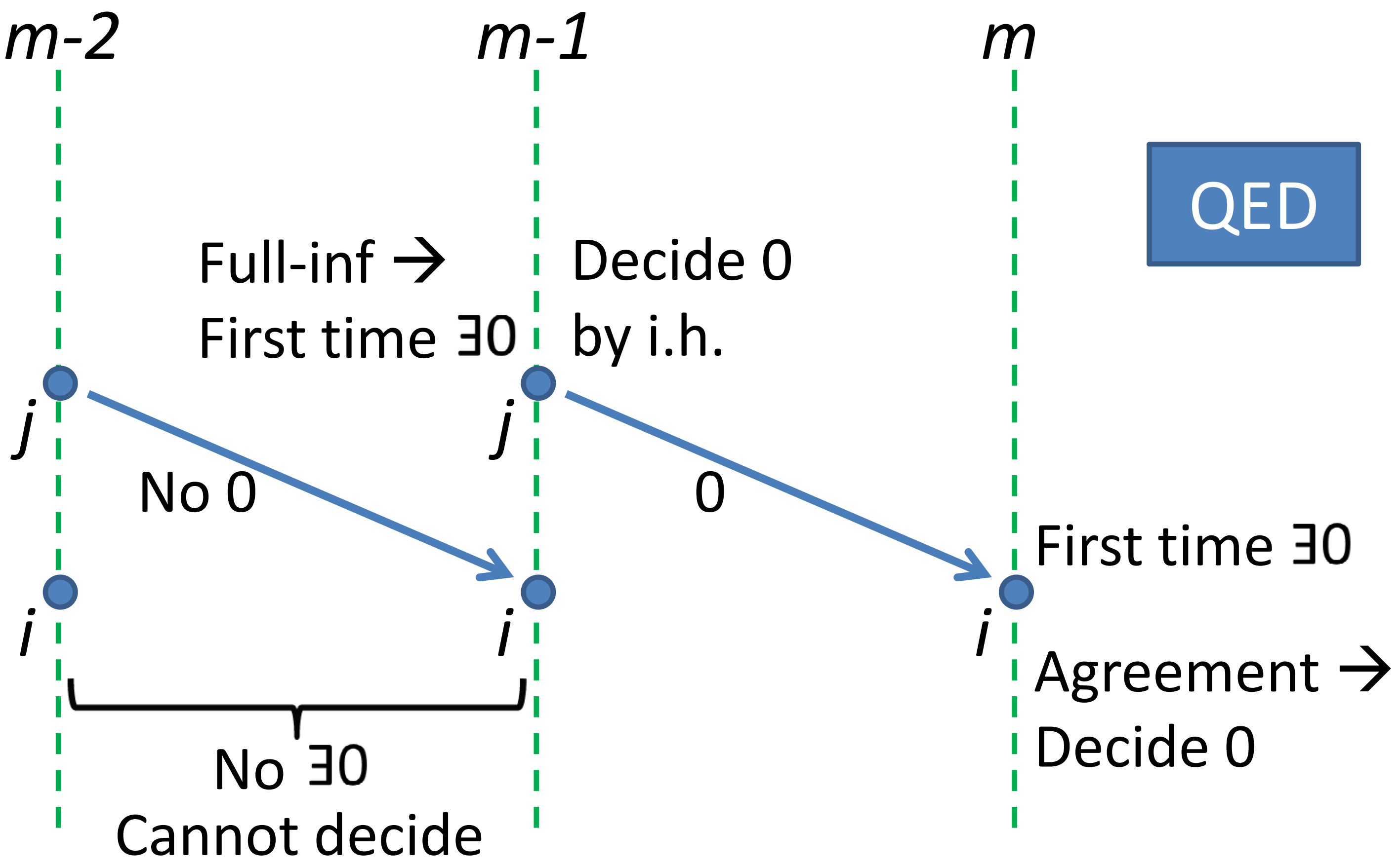
- P = Consensus protocol, processes decide as soon as $\exists 0$

- **Lemma 1**. For every Consensus protocol $Q \leq P$, each process i decides 0 in Q as soon as $\exists 0$
- **Proof:** By induction on the time m.

<u>Base $m = 0$:</u> Since $Q \leq P$, if i decides at time 0 in P, then i decides in Q at time 0. Process i starts with 0.

The Rule1 (1)

<u>Inductive step:</u>



The Rule1 (2)

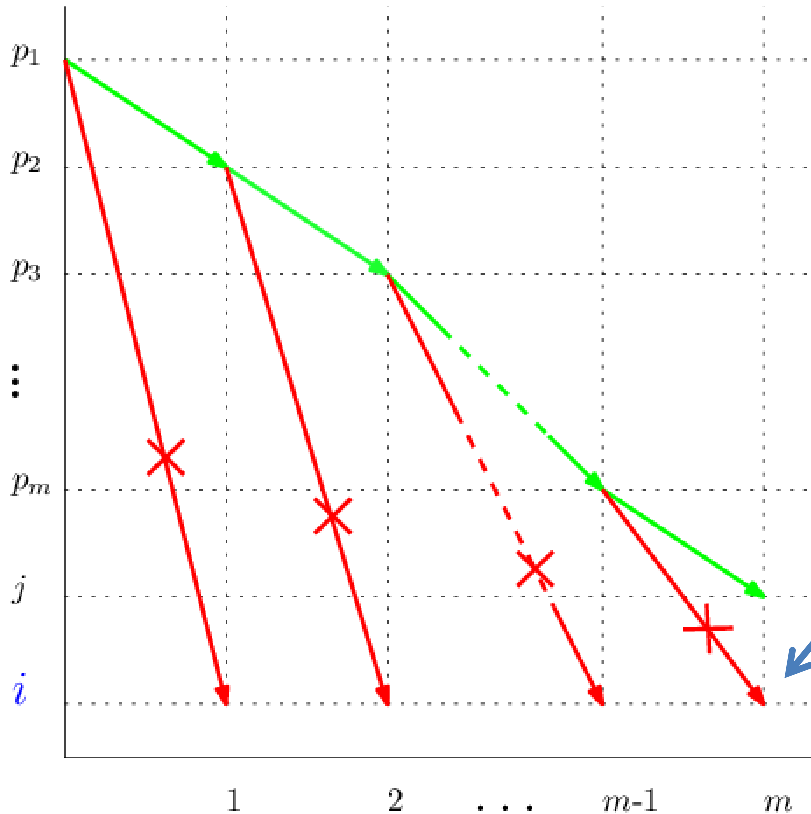
- **Lemma 2**. For every Consensus protocol $Q \leq P$, if at time m NO ∃0 for i and there is a hidden path w.r.t. i, then i cannot decide in Q at m.

The Rule1 (2)

- **Lemma 2**. For every Consensus protocol $Q \le P$, if at time m NO ∃0 for i and there is a hidden path w.r.t. i, then i cannot decide in Q at m.

- Hidden path

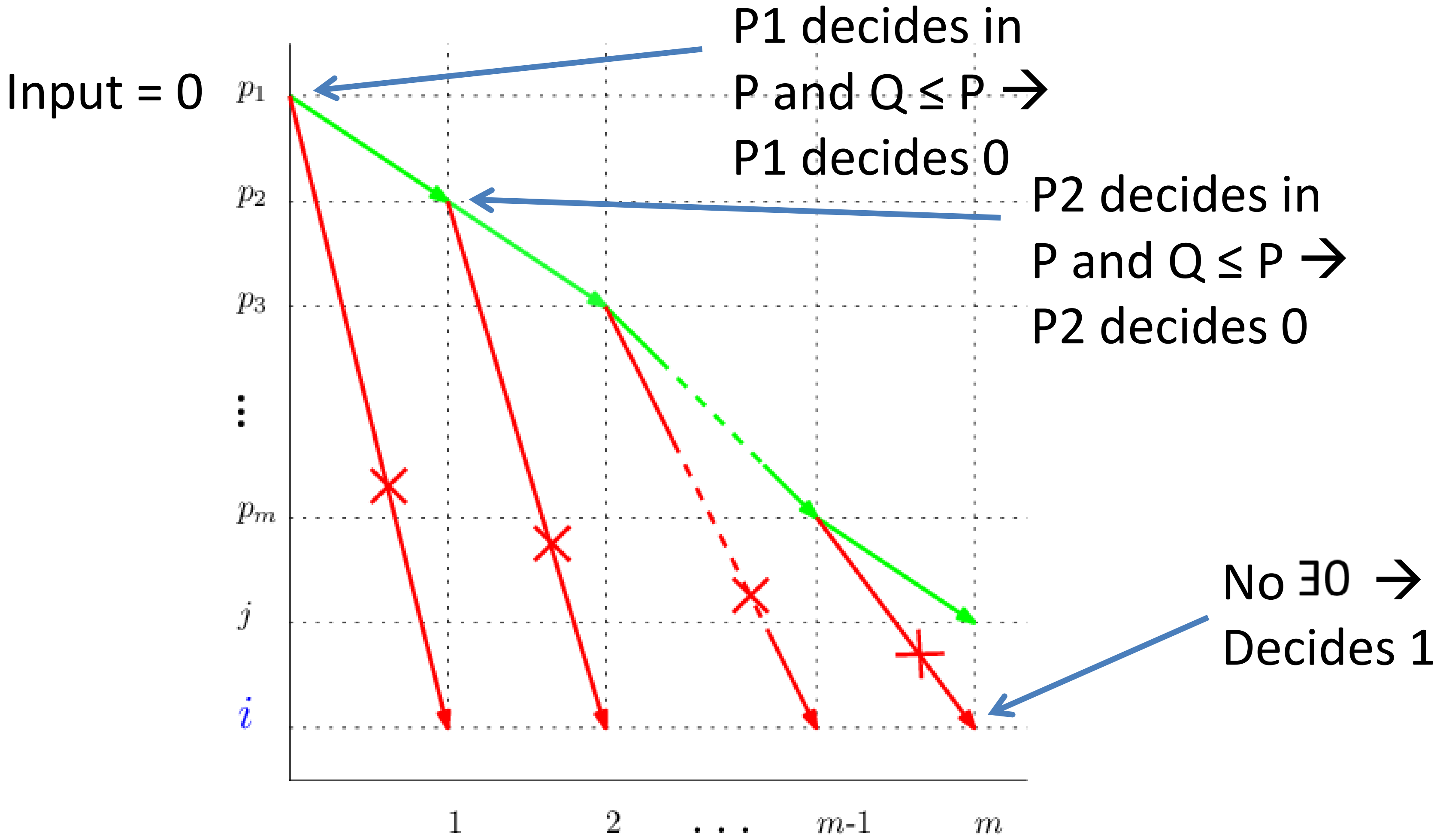
w.r.t. i at m:



i may not know some input values

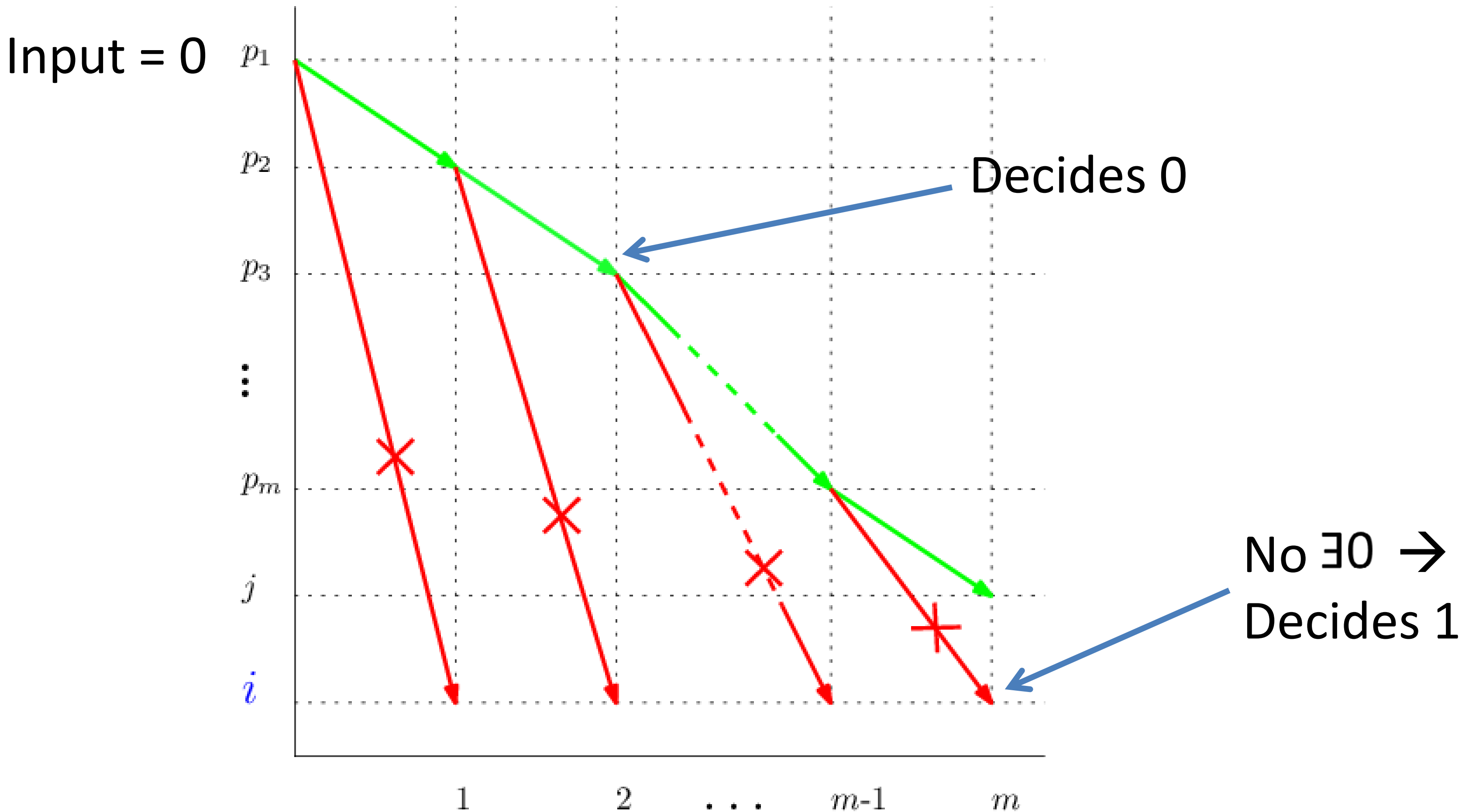
The Rule1 (2)

- **Proof:** By contradiction, *i* decides at *m*.



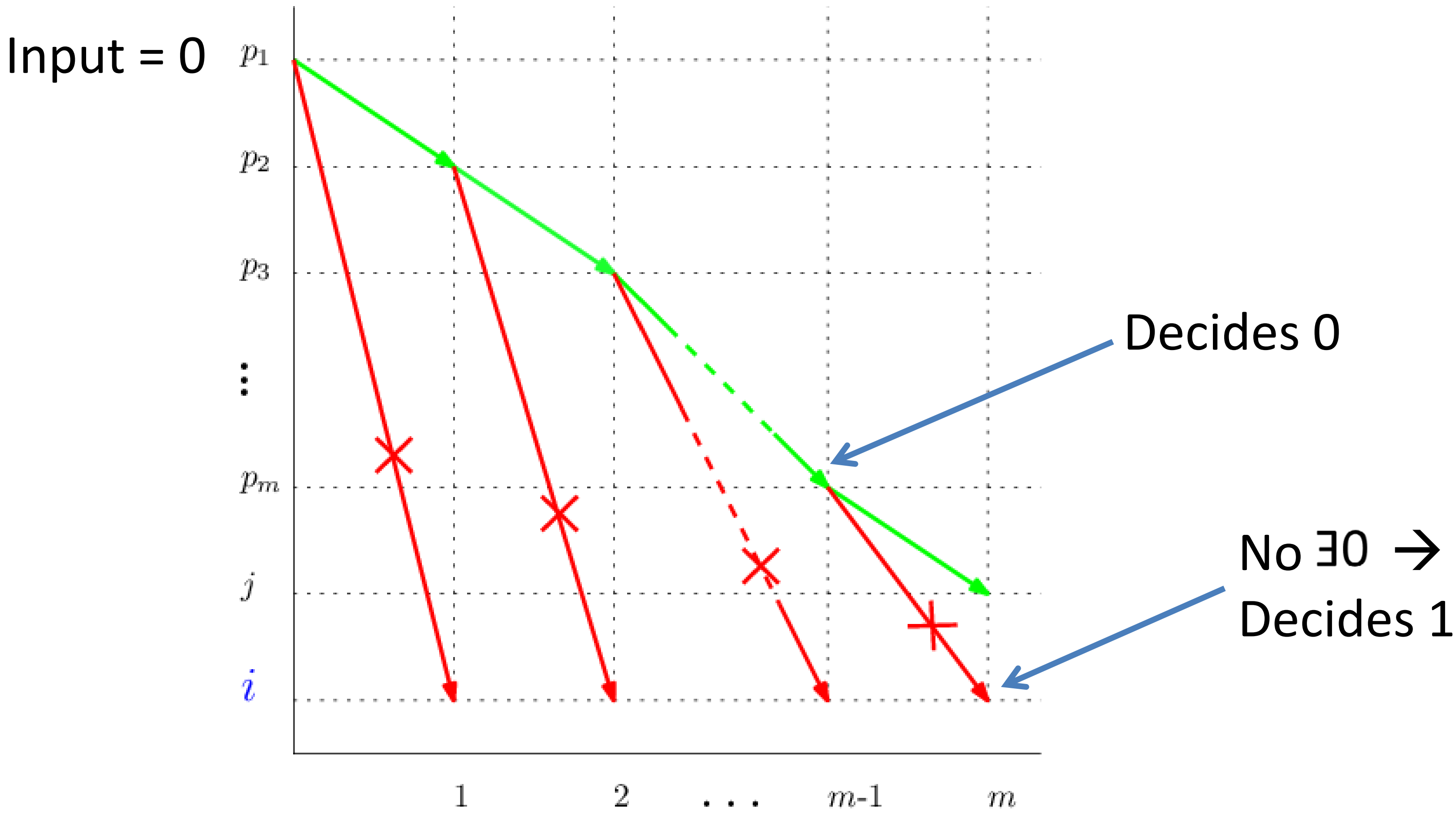
The Rule1 (2)

- **Proof:** By contradiction, *i* decides at *m*.



The Rule1 (2)

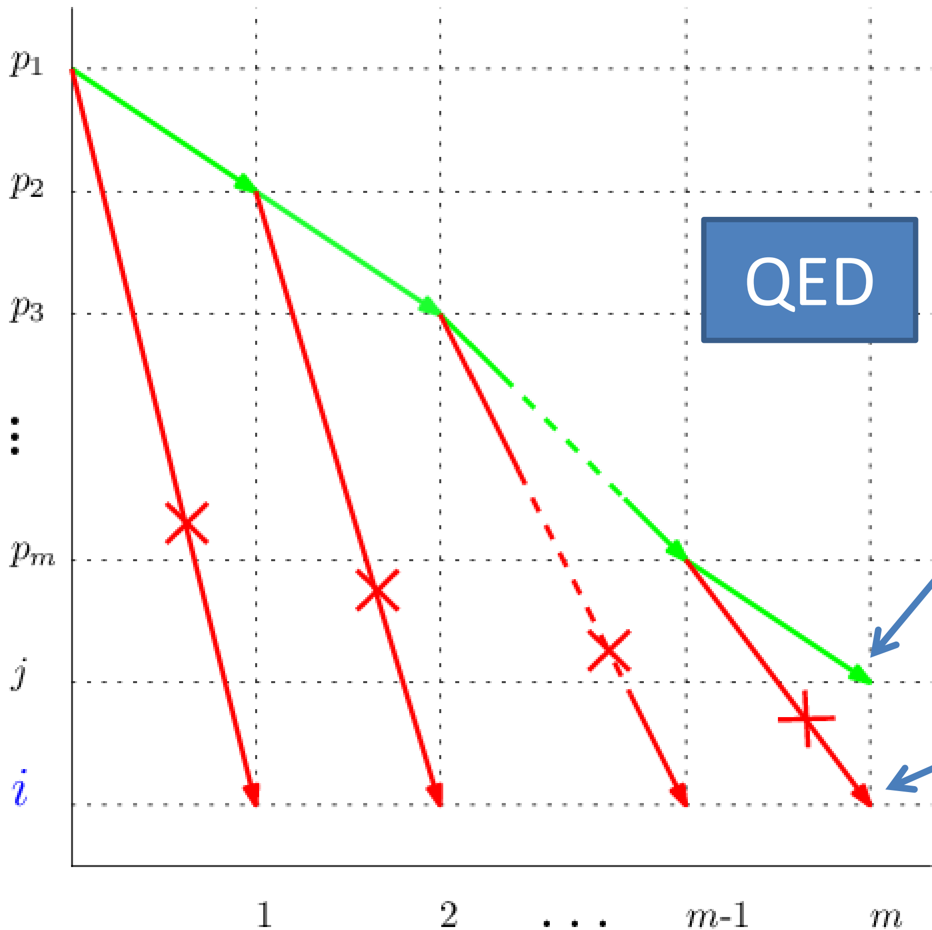
- **Proof:** By contradiction, *i* decides at *m*.



The Rule1 (2)

- **Proof:** By contradiction, *i* decides at *m*.

Input = 0



j is correct → *Q* does not solve Consensus!!

Decides 0

No ∃0 → Decides 1

The Rule1 (3)

- **Lemma 1**. For every Consensus protocol $Q \leq P$, each process *i* decides *0* in *Q* as soon as ∃0
- **Lemma 2**. For every Consensus protocol $Q \leq P$, if at time *m* NO ∃0 for *i* and there is a hidden path w.r.t. *i*, then *i* cannot decide in *Q* at *m*.

- Lemma 1 → Rule0 is unavoidable.
- Lemma 2 → Gives Rule1, which cannot be improved.

A Pareto Optimal Consensus Protocol

- Rule0 = $\exists 0$ = *i* receives a 0.
-
-

Stopping Time: If decided in round $r < t+1$, go one more round and then stop. Otherwise stop immediately.

IF *i* is undecided THEN

IF Rule0 THEN decide 0

IF Rule1 THEN decide 1

The *k*-Set Consensus Case

- Rule*v* = $\exists$ *v* = *i* receives a *v*, for *v=0,..,k-1*
- *k*
-

Stopping Time: If decided in round $r \le t/k+1$

Optimality Proof: Extends Lemma 1 and Lemma 2. Elementary analysis, NO reductions, NO topology.

IF Rule*v* THEN decide *v*

IF Rule*k* THEN decide *k*

Arbitrary Large Input Domain

- $V = \{0, \ldots, h\}, h \geq k$.
- RuleA = $\exists v$ = i receives a v, for $v=0,..,k-1$
- RuleB = Less than k disjoint hidden paths

- For process i (full-information):

  FOR round $r = 0, \ldots, t/k+1$ DO

  IF i is undecided THEN

  IF RuleA OR RuleB THEN

  decide min known value

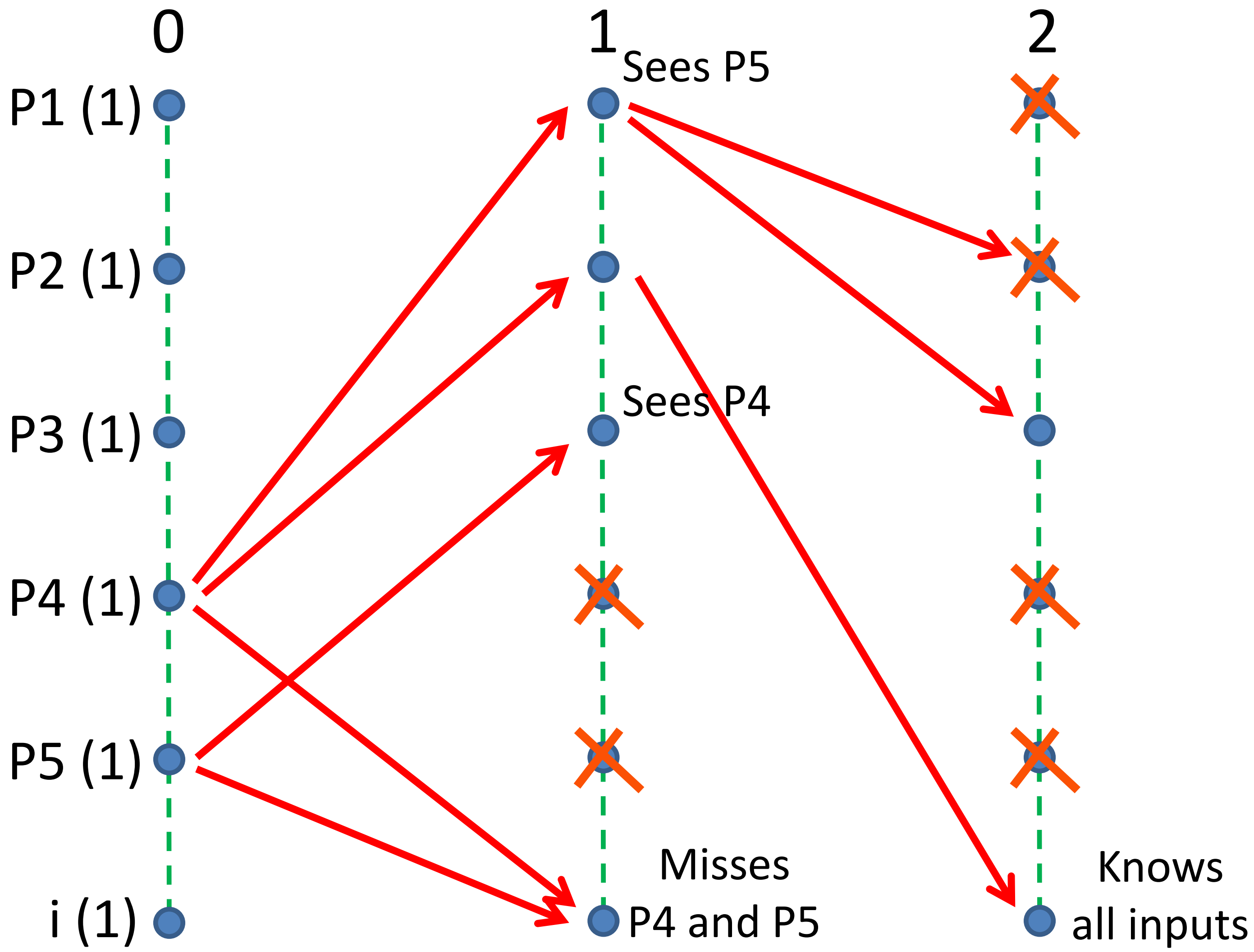
Size of Messages

- Full-information protocols only for analysis.
- Crash failures → Non-exponential size messages.
- In every round, each process only sends new information.
- Messages of polynomial size.

Previous Protocols (1)

- Our protocol strictly dominates all previous *k*-Set Consensus solutions.
- They only look at the current round.
- Our protocol looks at the past.

Previous Protocols (2)



Lower Bounds for Set Consensus (1)

- Our protocol performance contradicts published lower bounds [Alistarh et al. 2012, Guerraoui et al. 2009, Gafni et al. 2011]
- They claim: In every protocol NOT ALL correct processes can decide in round *f/k+1* or earlier.
- In our protocol: ALL correct processes decide in round *f/k+1* or earlier.
- <u>Source of the problem?</u>

Lower Bounds for Set Consensus (1)

- Our protocol performance contradicts published lower bounds [Alistarh et al. 2012, Guerraoui et al. 2009, Gafni et al. 2011]

- They claim: In every protocol NOT ALL **correct** processes can decide in round $f/k+1$ or earlier.

- In our protocol: ALL **correct** processes decide in round $f/k+1$ or earlier.

- Source of the problem?

Lower Bounds for Set Consensus (2)

- Non-uniform Set Consensus:
  - Correct processes decide at most k values.
- Uniform Set Consensus:
  - Faulty and correct processes decide at most k values.
- Alistarh et al. 2012 and Guerraoui et al. 2009 (implicitly) assume Uniform Set Consensus.
- Gafni et al. 2011 (implicitly) assume Uniform Set Consensus in different model.

No Topology but …

- Guerraoui and Pochon 2009, challenge for topology techniques.
- Optimality can be proved using topology.
- Not needed because the analysis is local.
- Needed when the analysis is on global decision lower bounds.

Thanks!!