Reading: 6.3-6.6, Ch. 7

Next: 

Last time: Distributed agreement with stopping failures

Simple alg: 
- Send whole set
- Send first 2 sets
- E1E

Lower bd: \( f+1 \) sets.

Proved by constructing a chain of execs, each with \( \leq f \) failures,

1st wrt dec. value 0; last 1, \( f \) any careless, pair looks

same to some non-faulty process.

(\( n \), look same to all-but-one process)

Today: On to Byzantine (most case failures)

Recall problem: Byz m-mode complete undirected graph

Agreement: No 2 nonf ports decide differently.

Val: If all nonf have same init, then all nonf that is the

only possible decision value

Term: All nonf. ports eventually terminate
Byzantine failures: (worst-case failures)

- node complete graph

We'll see

**EIG algorithm:**

- Exponential comm. (in 1)
  - $f+1$ nds
  - as also $m > 3f$

**Example:**

Suppose $p_1, p_2, p_3$ solve BA and tolerate 1 failure.

Consider (in this example only) restricted case where

- algorithm consists of 2 rounds
- each sends its own val in rd 1 + what the other process told it in rd 2

(similar to EIG structure)

**Exec. 1:**

1. $1 + 2$ nonfaulty, init val 1
2. 3 faulty, start with 0

rd 1: all act correct

rd 2: $p_3$ lies + tells $p_1$ that $p_2$ sent 0 in rd 1

Then validity requires that both $1 + 2$ decide 1.
x₂: Symmetric:
  2: nonfaulty, start with 0
  1: faulty, start with 1
  rd 1: all act correct
  rd 2: p₁ lies, tells 3 that 2 sent 1

All symmetric, now validity requires that both 2 + 3 decide 0.

Now get a contradiction by mixing up these 2 executions:

x₃: 1 + 3 nonf., start with 1 + 0 resp.
  2: faulty
  rd 1: 2 tells 1 its initial value = 1
  tells 3 "..." = 0
  inconsistent
  rd 2: all relay truthfully.

Notice
  x₃ 1 ≠ x₁
  x₃ 1 ≠ x₂
  x₃ 1 ≠ x₃
  So p₁ believes same in both, decides 1 in x₃

Contradicts agreement.

This wasn't a formal proof, but could be cast that way.
It shows that no alg. of this form 3 fronts, 2 relays
send + relay
what can way is the
can solve BA with 1 failure. Decision rule: keep no
rule will work
Note: The provs can tell something is wrong

correct

E.g. in $p_1$, $p_1$ sees that 3 sends 1

but

3 tells it 2 sent 0

So $p_1$ can tell that either $p_2$ or $p_3$ is faulty.

But doesn't know which.

Since alg. has to tolerate 1 fault, has to decide something, but

nothing works right in all possible cases.

Can extend the idea of this detailed construction to prove that 3 provs

can't solve BA for 1 failure (any form of alg., any number

of rds).

Come back to this after showing the EIG algorithm.

---

EIG alg for BA

Assumes $m > 3f$ (recall didn't need this for stopping alg.)

Same EIG tree, propagate values as before, $f+1$ rds

(but now provs should "throw away" any "ill-formed" msg + )

replace with 1

Now use different decision rule:

Replace all 1 with default value $v_0$.

Decorate nodes again, this time bottom-up, with new vals:

Leaf: $\text{neural}(x) = \text{val}(x)$

Non-leaf: $\text{neural}(x) = \begin{cases} \text{majority of children}, & \text{if exists} \\ v_0, & \text{else} \end{cases}$

Final decision $= \text{neural}(\lambda) \quad (\text{root label})$
Example: \[ T_{4,1} \]

Label with values in several (here, 2) rounds:
- \[ p_3\text{ faulty} \]
- Can get:

Then each separately calculates new-val, bottom-up, choosing majority values, default if no strict majority.
Use new-val at top as decision.

Here: \[ v_0 = 0 \text{ (default)} \]
Correctness:

Lemma 1: If $i, j, k$ nonfaulty, then $\text{val}(x)_i = \text{val}(x)_j$ for every label $x$ ending in $k$.

Above modes $x$ are

\[ 1 \quad 2 \quad 3 \quad 4 \]

If $i$ sends same to both, + they "decide" accordingly. (on the way down)

Lemma 2: If $x$ ends with nonf. process index, then $\exists v$ such that

$$\text{val}(x)_i = \text{neural}(x)_i = v$$

for all nonfaulty $i$.

Above, any of the $x$ modes have $\text{val} = \text{neural} = \text{same everywhere}$.

Pf: Induction on lengths of tree labels, from leaves up.

Basis: Leaf.

Then Lemma 1 implies all nonfaulty have same $\text{val}(x)$, so $v$.

+ $\text{neural} = \text{val}$ for leaves.

Inductive: $|x| = r \leq f$  \hspace{1cm} ($|x| = f+1$ at the leaves)

Again, Lemma 1 implies all nonfaulty have same $\text{val}(x)$, so $v$.

But need $\text{neural}(x)$ same too.

Every nonf. proc. has same $v$ for $x$ at $\text{val}(x+1)$, so

$$\text{val}(x+1)_i = v$$

for all nonfaulty $i$ and $i$.

By inductive hypothesis, also $\text{neural}(x+1)_i = v$ for all nonf. $i$.

Claim majority of labels of $x$'s children and with nonf. process indices:

Counting: \# of children $\geq m - f > 3f - f = 2f$

Only $f$ faulty

So majority rule applied by $i$ leads to $\text{neural}(x)_i = v$, for all nonfaulty $i$.  

\[ \boxed{} \]
This is already enough to see validity:

Validity: If all begin with $V$, then nonf. start $V$ at rol 1,
so $\text{val}(j)_i = V$ for all nonf. $i, j$.

By Lemma 2, also $\text{neural}(j)_i = V$ for all nonf. $i, j$.

Majority rule implies neural $(x)_i = V$ for all nonf. $i$.

So $i$ decides $V$.

Agreement: Needs a little more work.

Path covering: (of a tree) Subset of nodes containing at least one node on each path from root to leaf.

Common mode: One for which all nonf. roots have same neural.

(Does not necessarily end in nonf. root index)

Lemma 3: 3 path covering all of whose nodes are common.

Proof: Let $C = \text{all modes with labels of the form } x_i, i \text{ nonfaulty}.$

Lemma 2 implies these are common.

$
\leq 1 \text{ failures means these form a path covering}$

(each path contains $j+1$ distinct indices, $j$ at least one must be nonfaulty)

Now show that common modes "propagate up the tree":

Lemma 4: If there is a common path covering of subtree rooted at $x$ (any node) then $x$ is common.
Pf: Induction, from bases up.

Base: If \( x \) is a leaf, the only p.c. of subtree is \( x \) itself.
So \( x \) is common, as needed.

Inductive step: \( 1 \times 1 = n \leq f \)

Suppose \( \exists \) common p.c. \( C \) of \( x \)'s subtree.

If \( x \in C \), done.

If \( x \notin C \), then \( C \) includes a p.c. for each top-level subtree

(rooted at a child of \( x \))

say \( x_l \)

By ind. hypothesis, each such \( x_l \) is common.
So, all children of \( x \) are common.
Then def of neural implies \( x \) is common.
(all use same data to compute it)

Therefore, \( x \) is common.

Theorem: BA correct (recap)
Pf: Zem: Obvious

Validity: already argued
agreement: because \( x \) is common

Complexity: As for E16, in stopping model.
Plus \( n > 3f \) requirement.
Now show why we have to have \( m \geq 3f \) to solve BA.

### Number of processes for BA:

- **Alg. requires** \( m \geq 3f \)
- Can prove as a lower bound - holds for any graph with \( m \) nodes.
- But for graphs with low connectivity, may not even be able to tolerate this many failures.

### Number of failures that can be tolerated for BA in an undirected graph is completely characterized

- Depends on combination of \# of nodes + connectivity.

Start by showing 3 proc cannot solve BA with 1 fault.

**Proof:** By contradiction.

- Suppose Alg. A, proc 1, 2 + 3, solves BA in 1 fault.
- Construct new system 5 of 2 copies, start off with init values as follows:

![Diagram of system 5](image)

- What is 5? A synchronous system of some kind, but not required to satisfy any special card. cards.
- But we can use it in getting a contradiction anyway.

Start with 0's, 1's as above.

Runs, does something.

- Consider \((2 + 3)\). Looks like:
- In \( \Delta \), must decide 0.
- So they do in 5 also.
Consider 3 + 1: Must agree in $A$ vs $S$.

But one dec. 0 and the other 1, in $S$.

Contrad.

Discuss

Even get the contradiction if the original algorithm is allowed to "know $n$".

This just means the passes in $A$ have 3 million someplace...

... and their validity + agreement + termination conditions are only required to hold if they actually are placed in a $A$.

But that's all we used in the proof!

That's $3 \neq 1$.

But the same idea extends to $3f$ vs $f$ for any $f$.

Can do a similar construction, with $f$ passes playing the role of each 1.

Or, can do a reduction: Show how to transform a $3f$ vs $f$ solution

to a $3vs1$ solution.

Since we already know $3vs1$ doesn't exist, that yields a contradiction.

If $m = 2$, easy to see impossible? LITR

$0 \leq \frac{2}{2}$

Each can be faulty, requiring 0, 1 resp.

On both non-faulty, requires $\ldots$
So assume $3 \leq m \leq 3f$, solve $A$.  
Transform to $B$, solving $3m > 1$.  
Partition $A$'s procs into $3$ groups, each $1 \leq i \leq f \lor i \in I_1, I_2, I_3$.  
Each of the $3$ procs simulates $1$ group.  
Initializes all with the input value.

Each round: Simulate sending of msgs: 
- Local: just sim 
- Remote: package + send

If any simulated process decides, decide same.  

Consider exec. of $B$ with $\leq 1$ fault.  
" Mimics" execution of $A$ with $\leq f$ faults.  
So agreement, validity, termination must hold for the emulated execution.

Show properties carry over to $B$'s exec.

Termination: If $i$ is monf. proc of $B$, then simulates at least 1 (monf) proc of $A$.  So terminates, so $i$ does.

Validity: If all monf. procs of $B$ start with $v$, then so do all monf. procs of $A$.  Validity for $A$ then implies all monf. decide $v$, so part in $B$.

Agreement: If $i, j$ monf. procs of $B$, simulate only monf. procs of $A$.  Agreement in $A$ implies these all agree, so $i \leftrightarrow j$ agree.