**BA for Stopping Failures**

**Assumptions:** Synchronicity; Generals may have fail-stop faults

**Algorithm:** pretty much like the case of authentication:

- **(send)** \( \text{send}(W_i) \)
- **(receive)** \( \text{receive}(u_1, \ldots, u_k) \)
- **(local)** \( W_i := W_i \cup \{u_1, \ldots, u_k\} \)

At round \( f + 1 \), add to \( \text{local} \):

\[
\text{if } |W_i| = 1 \text{ then } d_i = v_i \\
\text{else } d_i = \hat{v}
\]
Proof of Correctness

Claim. If all correct processes have $W_i^r = W$ after round $r \leq f$, they all have $W_i^{r+1} = W$

Proof. $W_i$'s are monotonically increasing. To increase $W_i$, $i$ has to receive a new $v_k \not\in W_i^r$, which can happen only if $v_k \in W_j^r$ for some correct $j$. This is impossible since $W_i^r = W_j^r$ for all correct $j$'s.

Claim. If there is a round $r \leq f + 1$ without failures, then all correct processes have the same $W$ after it

Proof. At the end of round $r$, for every correct $i$

$$W_i^r = \bigcup_{j \text{ is correct at round } r} W_j^{r-1}$$

Corollary. Upon termination, all correct processes have the same $W$
Complexity

**time.** $f + 1$ rounds

**messages** $\leq n^2$ in each round, thus $\mathcal{O}(n^2(f + 1))$ messages

**bits.** if values are at most $b$ bits, $\mathcal{O}(bn^2(f + 1))$ bits

if only first two values sent. $f + 1$ rounds and $\mathcal{O}(2 \cdot n(f + 1))$ messages

If decision is some initial value: decide minimal $v \in V$, or, for optimality, broadcast only minimal value observed
Byzantine Failure is Harder

Assume 3 processes (triangle) using EIG for two rounds ($f = 1$)

We construct three executions:

- $\alpha_1$ in which both 1 and 2 are correct and $v_1 = v_2 = 1$ so decision should be 1
- $\alpha_2$ in which 2 and 3 are correct and $v_2 = v_3 = 0$ so decision should be 0
- $\alpha_3$ in which 1 and 3 are correct, $v_1 = 1$, $v_3 = 0$; decision should be either 0 or 1

and show that $\alpha_3 \sim_1 \alpha_1$ and $\alpha_3 \sim_3 \alpha_2$, so in $\alpha_3$ 1 should decide 1 and 3 should decide 0!
Byzantine Failure is Harder

<table>
<thead>
<tr>
<th>Correct</th>
<th>Initial Values</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha_1)</td>
<td>1 and 2</td>
<td>(v_1 = v_2 = 1)</td>
</tr>
<tr>
<td>(\alpha_2)</td>
<td>2 and 3</td>
<td>(v_2 = v_3 = 0)</td>
</tr>
<tr>
<td>(\alpha_3)</td>
<td>1 and 3</td>
<td>(v_1 = 1; \ v_3 = 0)</td>
</tr>
</tbody>
</table>

- In \(\alpha_1\), \(v_3 = 0\), and only faulty behavior is that at second round 3 tells 1 that 2 sent it 0.
- In \(\alpha_2\), \(v_1 = 1\), and only faulty behavior is that at second round 1 tells 3 that 2 sent it 1.
- In \(\alpha_3\), at first round 2 tells 1 its initial value is 1 and 2 that its initial value is 0.
- \(\alpha_3 \sim_1 \alpha_1\) and \(\alpha_3 \sim_3 \alpha_2\)
Synchronous (Unauthenticated) BA

Assumptions:

- Source of messages is uniquely determined
- Only values are 0 and 1
- \( f < \frac{n}{3} \)
- There is a Consistent Broadcast mechanism for messages of the form \((m, i, r)\) guaranteeing:
  1. If \((m, i, r)\) is broadcast by a correct \(i\) at round \(r\), it is accepted by all correct processes by round \(r + 1\)
  2. If \((m, i, r)\) is not broadcast by a correct \(i\) at round \(r\), it is never accepted by any correct process
  3. If a message is accepted by some correct process, it is accepted by all correct processes at most one round later
Synchronous BA

The algorithm proceeds in $f + 1$ phases, each consisting of an odd round followed by an even round. The only messages sent are of the form $(r, i)$, and they are only sent at odd rounds.

- At round 1, all processes $i$ with $v_1 = 1$ (consistent) broadcast $(1, i)$
- At an odd round $r$, $i$ broadcasts a message if:
  1. it hadn’t done that before; and
  2. it had accepted messages from at least $f + \frac{r-1}{2}$ different processes
- A process decides 1 iff it had accepted messages from $2f + 1$ different processes
Correctness of Protocol

Claim: The protocol solves strict BA if \( n > 3f \).

Proof:

- **Termination.** Obvious

- **Validity.** If all correct processes start with 1, then, according to property 1 of consistent broadcast, all correct processes accept \( n - f \geq 2f + 1 \) messages by round 2; thus, they all decide 1. Similarly, if they all start with 0, it is easy to show, by induction on number of rounds using property 2 of consistent broadcast, that no correct process ever broadcasts since it never accepts a message, so they all decide 0.

- **Agreement.** Assume that a correct \( i \) decides 1...
Correctness of Protocol (cont.)

Assume that a correct $i$ decides 1

$i$ accepted messages from $2f + 1$ different processes, out of whom at least $f + 1$ are correct. Let $C$ denote the set of such correct processes, and $C_b$ denote the subset of $C$ processes whose initial value is $b$

$C_1$ includes at least one process

For every $j \in C_1$, $(1, j)$ is accepted by all correct processes by round 2

If $|C_1| \geq f + 1$, then all correct processes accept $f + 1$ messages by round 2, b’cast at round 3, and accept from at least $n - f$ processes by round 4. Hence, all correct processes decide 1
Correctness (cont.)

We still need to deal with $1 \leq |C_1| \leq f$:

- Assume $j \in C_0$ b’casts at phase $p$

- Thus $j$ accepts from at least $f + p - 1$ (non-$j$) processes by the beginning of phase $p - 1$

- From property 3, all correct processes accept from $f + p - 1$ (non-$j$) processes by the end of phase $p$

- All correct processes accept from $f + p$ processes by the end of $p$

- If $p < f + 1$, they all b’cast by $p + 1$ and accept from $n - f$ processes by the end of $p + 1$

- Else $f + p = 2f + 1$ and they all accept the required number of messages by the end of phase $f + 1$

- In any event, all correct processes decide $1$
It’s easy to see that the protocol requires $2(f + 1)$ rounds, and $n$ broadcast invocations. As we shall see, each invocation of consistent broadcast requires $O(n^2)$ messages, thus the communication complexity is $O(n^3)$ messages.
Consistent Broadcast

To broadcast \((m, i, r)\) at round \(r\), process \(i\) sends ("init", \(m, i, r\)) to all processes at round \(r\).

Every process that receives ("init", \(m, i, r\)) from \(i\) at round \(r\), sends ("echo", \(m, i, r\)) to all processes at round \(r + 1\).

If before round \(r' \geq r + 2\), a process receives ("echo", \(m, i, r\)) from at least \(f + 1\) processes, it sends ("echo", \(m, i, r\)) to all at round \(r'\).

If before round \(r' \geq r + 1\), a process receives ("echo", \(m, i, r\)) from at least \(n - f\) processes, it accepts \((m, i, r)\).
Correctness of Consistent Broadcast

- If a correct \( i \) broadcasts \( M = (m, i, r) \) at round \( r \), all correct processes receive at least \( n - f \) ("echo", \( M \)’)s by end of \( r + 1 \); all accept \( M \)

- If a correct \( i \) does not broadcast \( M \) at round \( r \), no correct process receives ("init", \( M \)), none (ever) sends ("echo", \( M \)). Thus no correct process accepts \( M \)

- If a correct process accepts \( M \) at round \( r' \), then it receives \( \geq n - f \) ("echo", \( M \)), at least \( f + 1 \) from correct processes. Each correct process receives at least \( f + 1 \) ("echo", \( M \)’)s by round \( r' \), and \( \geq n - f \) ("echo", \( M \)’)s by round \( r' + 1 \). Hence, all correct process accept \( M \) by round \( r' + 1 \)