Approximate Agreement
Recap: Byzantine Agreement
Recap: Byzantine Agreement

• n parties, out of which f may be byzantine

• Byzantine Agreement requires:
  • **Agreement**: honest parties obtain identical outputs
  • **Validity**: the honest parties’ output is one of their inputs
Recap: Byzantine Agreement

- $n$ parties, out of which $f$ may be byzantine
- Byzantine Agreement requires:
  - **Agreement**: honest parties obtain identical outputs
  - **Validity**: the honest parties’ output is one of their inputs

| Synchronous networks | • Deterministic protocols
|                      | • $f+1$ communication rounds |
| Asynchronous networks | • No deterministic protocols
|                      | • $\Rightarrow$ Variants   |
Approximate Agreement

• n parties, out of which f may be byzantine

• Approximate Agreement requires, for any given $\varepsilon$:
  • $\varepsilon$-Agreement: honest parties obtain $\varepsilon$-close outputs
  • Validity: honest parties’ outputs are within the range of their inputs
Approximate Agreement

($\varepsilon = 0.5$)
Approximate Agreement

($\varepsilon = 0.5$)
Approximate Agreement

(\(\varepsilon = 0.005\))
Algorithm outline

Iterations:

honest inputs’ range

\[ \leq \varepsilon \]
Algorithm outline

In iteration $i$:

1. Distribute your value $v$. Let $V$ denote the multiset of values received.
   
   $(-100000, 4.5, 10, 21, 20)$

2. Obtain $V'$ by discarding the outliers from $V$

3. Compute a new value $v' = \frac{1}{2} (\min V' + \max V')$
Discarding outliers
(a possible approach)

What would the byzantine parties do?

\[ V = (-100000, 4.5, 10, 20, 21) \]
\[ V = (4.5, 10, 20, 21, +100000) \]
\[ V = (4.5, 10, 15, 20, 21) \]
\[ V = (4.5, 10, 20, 21) \]
Discarding outliers
(a possible approach)

f corrupted parties involved
=> discard the lowest f and the highest f values

\[ V' = (-100000, 4.5, 10, 20, 21) \]

\[ V' = (4.5, 10, 20, 21, +100000) \]

\[ V' = (4.5, 10, 15, 20, 21) \]

\[ V' = (4.5, 10, 20, 21) \]
If even after discarding outliers, honest parties have some common range:

Convergence?
If even after discarding outliers, honest parties have some common range:

\[
\text{honest range} \leq \frac{1}{2} \cdot \text{honest range size}
\]
If the honest parties’ inputs are between A and B:

- After 1 iteration, their values are \( \left( \frac{B-A}{2} \right) \)-close.
- After 2 iterations, their values are \( \left( \frac{B-A}{4} \right) \)-close.
  ...
- After \( k \) iterations, their values are \( \left( \frac{B-A}{2^k} \right) \)-close.

\[ \Rightarrow \log_2 \left( \frac{B-A}{\varepsilon} \right) \] iterations are sufficient
A simple asynchronous algorithm

In iteration $i$:

1. Send your value $v$ to everyone via Reliable Broadcast and let $V$ denote the multiset of $\geq n - f$ values received.
2. Obtain $V'$ by discarding the lowest $f$ and the highest $f$ values from $V$.
3. Compute a new value
   
   $$v' = \frac{1}{2} (\min V' + \max V')$$

$f < n/4$?

• Validity ✓
• $\varepsilon$-Agreement:
  • Two honest parties have $(n - f) + (n - f) - n = n - 2f$ values in common.
  • At most $2f$ of these values are discarded.
  • $n - 4f > 0 \Rightarrow$ common range ✓
A simple asynchronous algorithm

In iteration $i$:

1. Send your value $v$ to everyone via Reliable Broadcast and let $V$ denote the multiset of $\geq n - f$ values received.

2. Obtain $V'$ by discarding the lowest $f$ and the highest $f$ values from $V$.

3. Compute a new value

$$v' = \frac{1}{2} (\min V' + \max V')$$

$f < n/3$?

- Validity ✓
- $\varepsilon$-Agreement:

Honest values: 4.5, 10, 10

- $(1000000, 4.5, 10)$ ✗
- $(4.5, 10, 10)$ ✗
Is \( f < \frac{n}{3} \) possible?

Yes, but we need to ensure common range, even after discarding outliers.

\[ \Rightarrow \text{Witness technique} \]
Witness technique

Code for party $P$ with input $v$:
1. Send $v$ to every party via Reliable Broadcast
2. When receiving $n-f$ values ($v_1$ from $P_1$, ..., $v_{n-f}$ from $P_{n-f}$):

   Reliable Broadcast guarantees that every party can receive these values as well.

   $\Rightarrow$ Let them know by sending a witness report
   $\Rightarrow (v_1, P_1, v_2, P_2, ..., v_{n-f}, P_{n-f})$
Witness technique

Code for party P with input v:

1. Send v to every party via Reliable Broadcast
2. When receiving n-f values (v₁ from P₁, ..., vₙ₋f from Pₙ₋f):
   - Send (v₁, P₁, v₂, P₂, ..., vₙ₋f, Pₙ₋f) to every party.
3. When receiving a witness report from P’:
   - When all values reported by P’ are received, mark P’ as a witness.
4. When n – f parties are marked as witnesses:
   - Output the values received via Reliable Broadcast.
Why do we have enough common values?

• Each honest party has $n - f$ witnesses
• Every two honest parties have at least:
  $$(n - f) + (n - f) - n = n - 2f > f$$
  witnesses in common
  $\Rightarrow$ at least one honest witness $P$ in common
  $\Rightarrow$ they received the same $n - f$ values in $P$’s witness report
  $\Rightarrow$ in Approximate Agreement, even after discarding outliers,
  they end up with $n - 3f > 0$ values in common
Asynchronous protocol

In iteration $i$:

1. Send your value $v$ to everyone using the Witness technique. Let $V$ denote the multiset of $\geq n - f$ values received.

2. Obtain $V'$ by discarding the lowest $f$ and the highest $f$ values from $V$.

3. Compute a new value $v' = \frac{1}{2} (\min V' + \max V')$

\begin{align*}
\text{f < n/3?} & \quad \text{Optimal} \\
\quad \text{• Validity} & \quad \checkmark \quad \text{• } \varepsilon\text{-Agreement} & \quad \checkmark
\end{align*}
• Approximate Agreement is interesting here: #rounds does not depend on f.
• The asynchronous protocol works for $f < \frac{n}{3}$.
• $f < \frac{n}{2}$ is also possible, using signatures.

Optimal
Issues when \( f < \frac{n}{2} \)

Discarding outliers?

- \( n - 2f \) may be one value or less
- But:
  - if \( n - f + k \) values are received
    - At most \( k \) out of these may be corrupted
Issues when $f < n/2$

Common range?

- Corrupted values might be inconsistent
  
  \((\overline{100000}, 0, \pm)\) 
  
  \((0, 1, 100000)\)

- How do we guarantee consistency?
  Weak Broadcast (with signatures)
Weak Broadcast

Code for sender S with input v:
1. Sign v and send \((v, \sigma)\) to every party

Code for receiver P:
1. If you received \((v, \sigma)\) from S, forward it to every party
2. If > f parties confirmed \((v, \sigma)\) and no other signed value was received, output v.

Guarantees:

• If S is honest, every honest party outputs v.
• If honest P and P’ output v and v’, then v = v’.
How would this guarantee common range?
Synchronous protocol \((f < n/2)\)

In iteration \(i\):

1. Send your value \(v\) to everyone via \textbf{Weak Broadcast}.
   
   Save the \(n - f + k\) received values in \(V\).

2. Obtain \(V'\) by discarding the lowest \(k\) and the highest \(k\) values from \(V\).

2. Compute your new value \(v' = \frac{1}{2} (\min V' + \max V')\).
Results so far
Is there a best-of-both worlds?

• The parties are not aware of the type of network the protocol runs in.

• Is there a protocol that achieves Approximate Agreement secure against:
  • $f_s < \frac{n}{2}$ byzantine parties when the network is actually synchronous, and
  • $f_a < \frac{n}{3} \leq f_s$ byzantine parties when the network is actually asynchronous?

Yes!

If $2 \cdot f_s + f_a < n$ (optimal).
• Approximate Agreement:
  • Allows an error of $\varepsilon$, but:
    • # synchronous rounds does not depend on $f$
    • Has deterministic asynchronous protocols
  • Synchronous protocol for $f < n/2$ (optimal)
  • Asynchronous protocol for $f < n/3$ (optimal)
• Best-of-both worlds protocols
• Happy holidays!