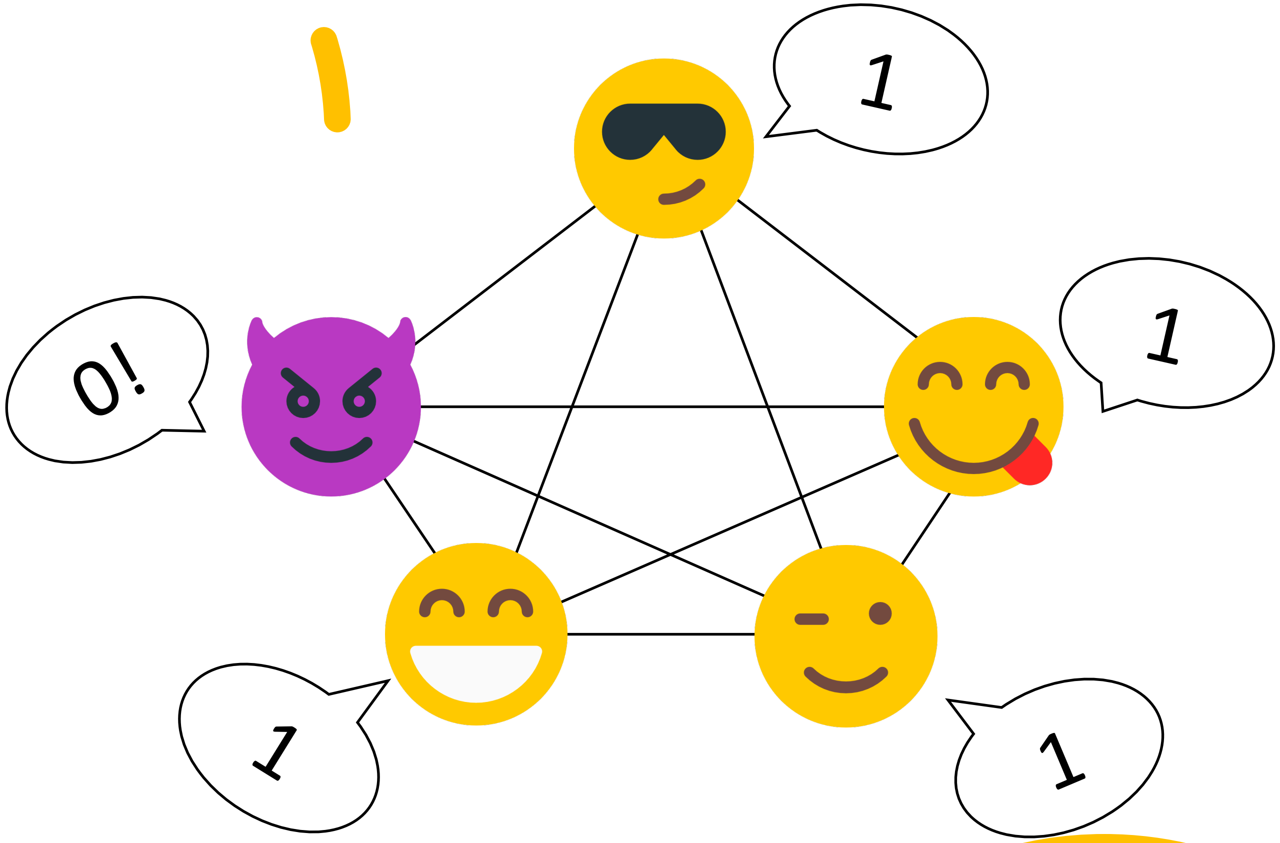
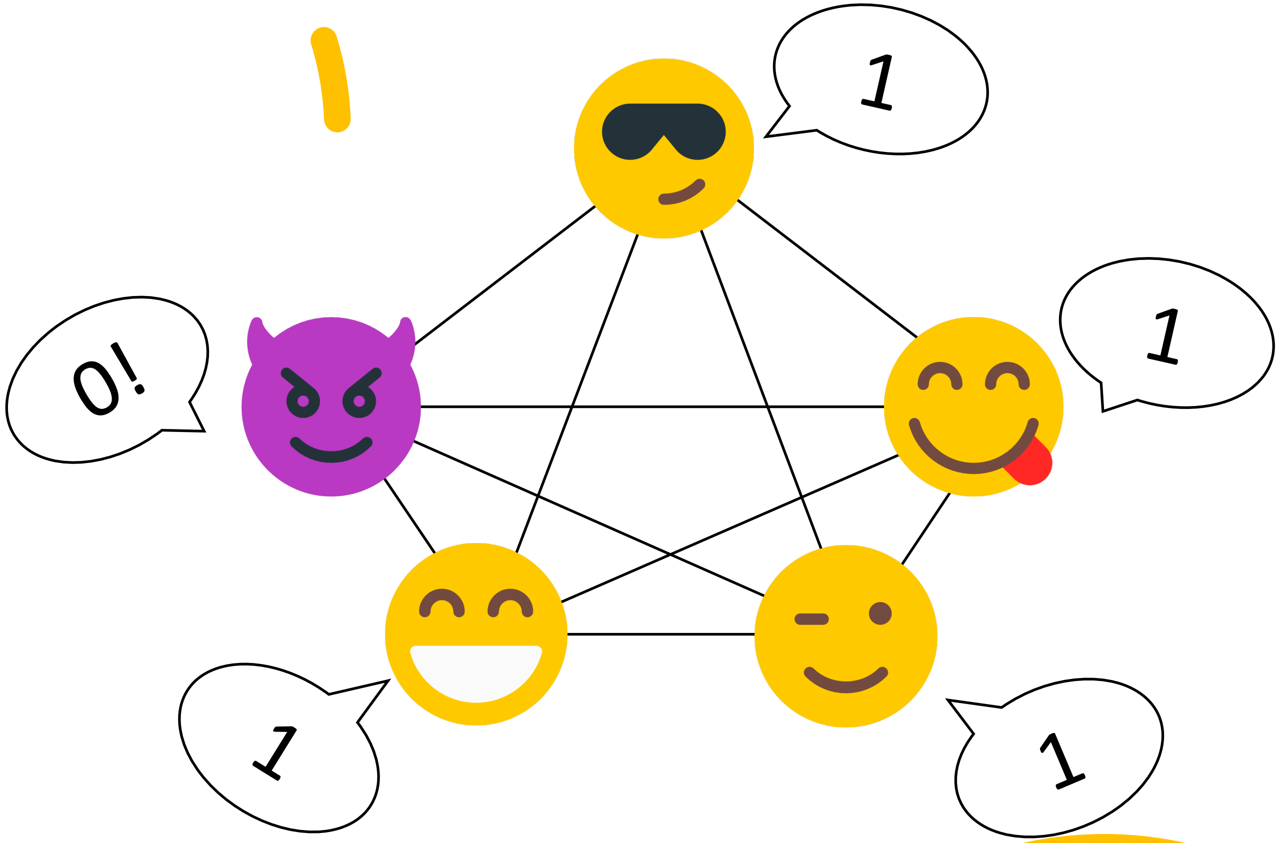
Approximate Agreement

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

ε-Agreement

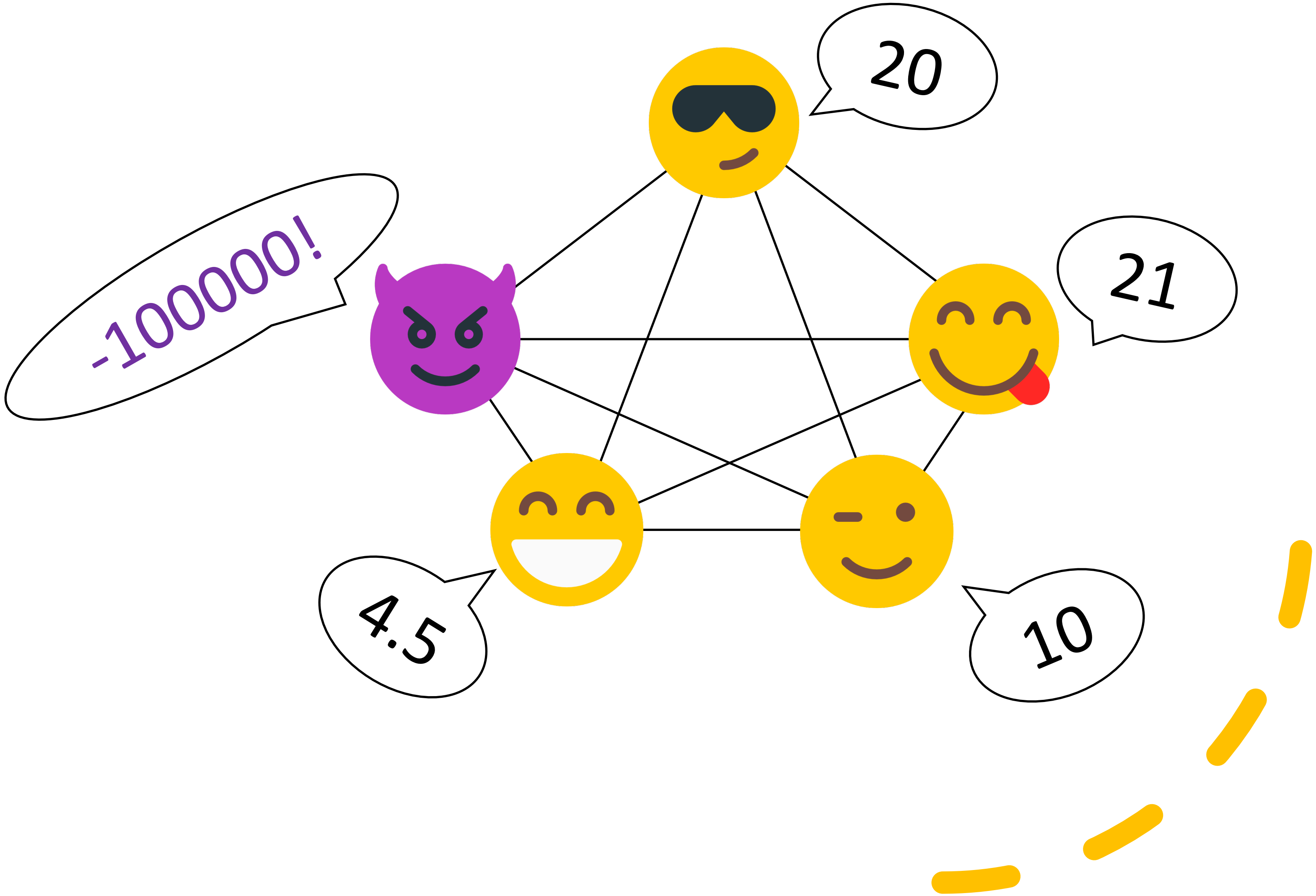
Synchronous networks	• Deterministic protocols • f+1 communication rounds
Asynchronous networks	• No deterministic protocols • => Variants

Approximate Agreement

- n parties, out of which f may be byzantine
- Approximate Agreement requires, for any given ε:
  - **ε-Agreement**: honest parties obtain ε-close outputs
  - **Validity**: honest parties' outputs are within the range of their inputs

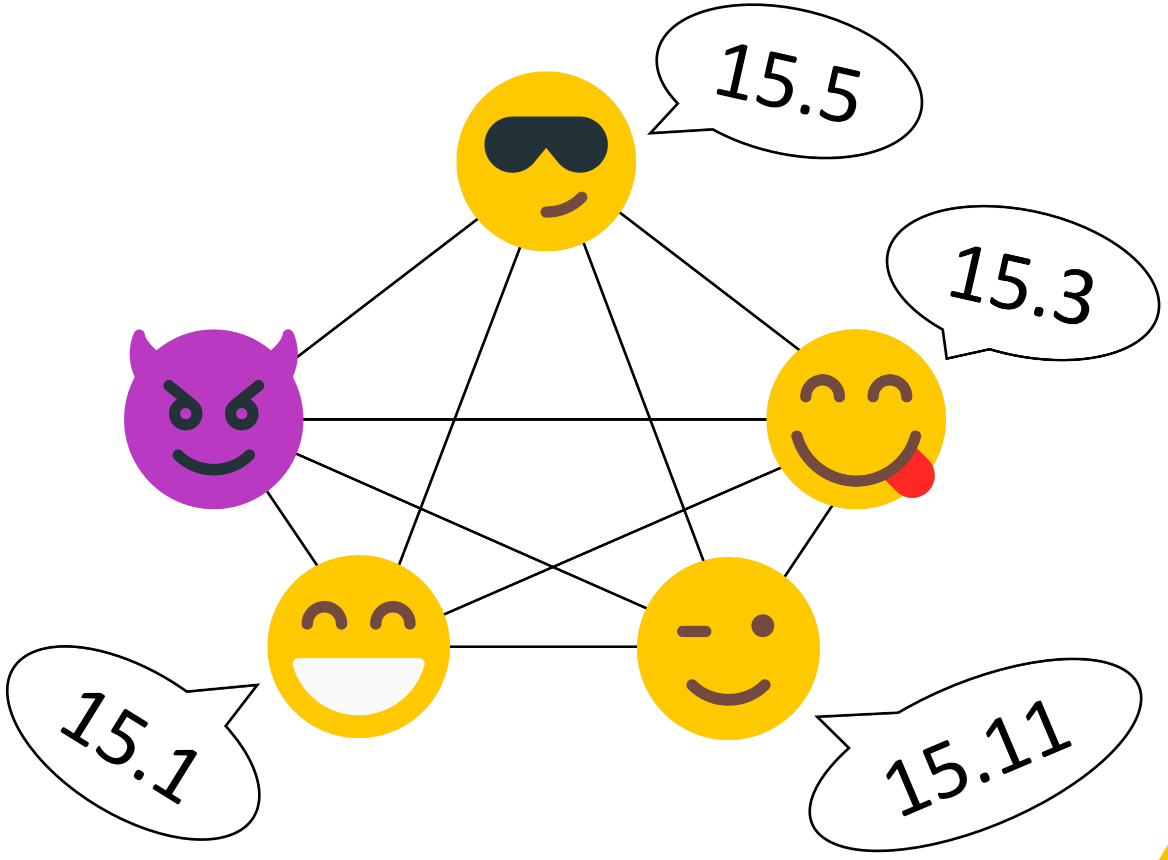
Approximate Agreement

$(\varepsilon = 0.5)$



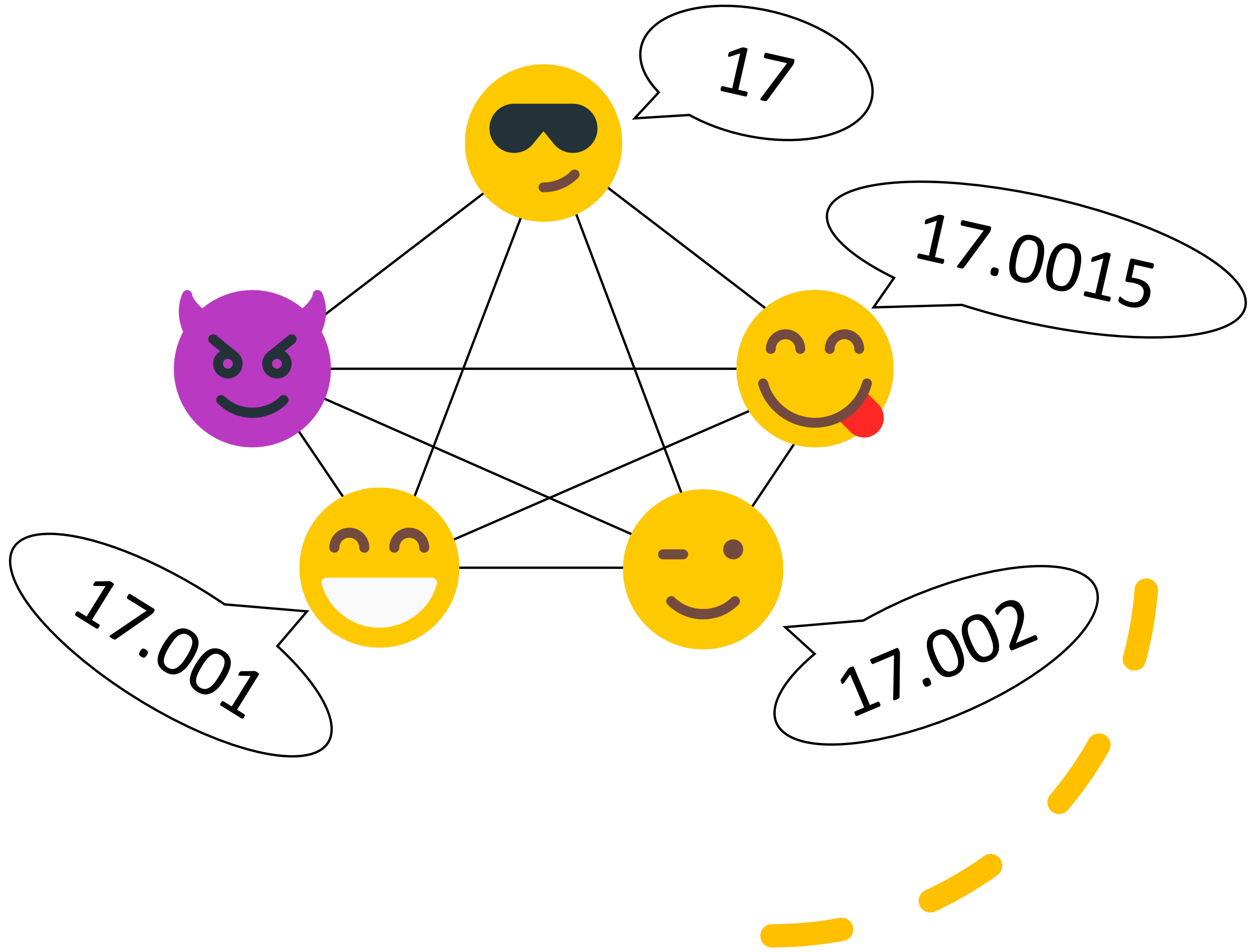
Approximate Agreement

$(\varepsilon = 0.5)$



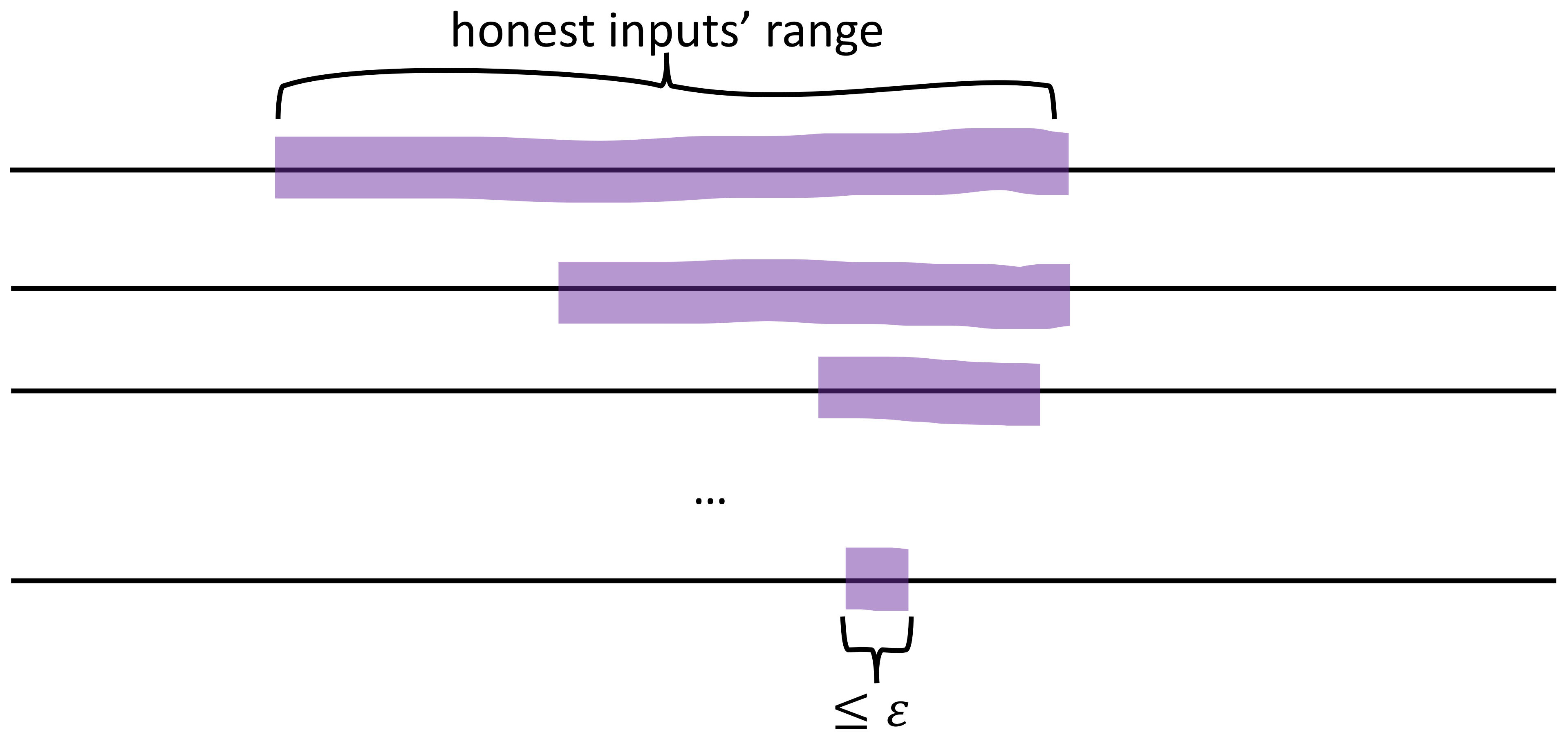
Approximate Agreement

$$(\boldsymbol{\varepsilon} = \mathbf{0.005})$$



Algorithm outline

Iterations:



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

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~~)
😈

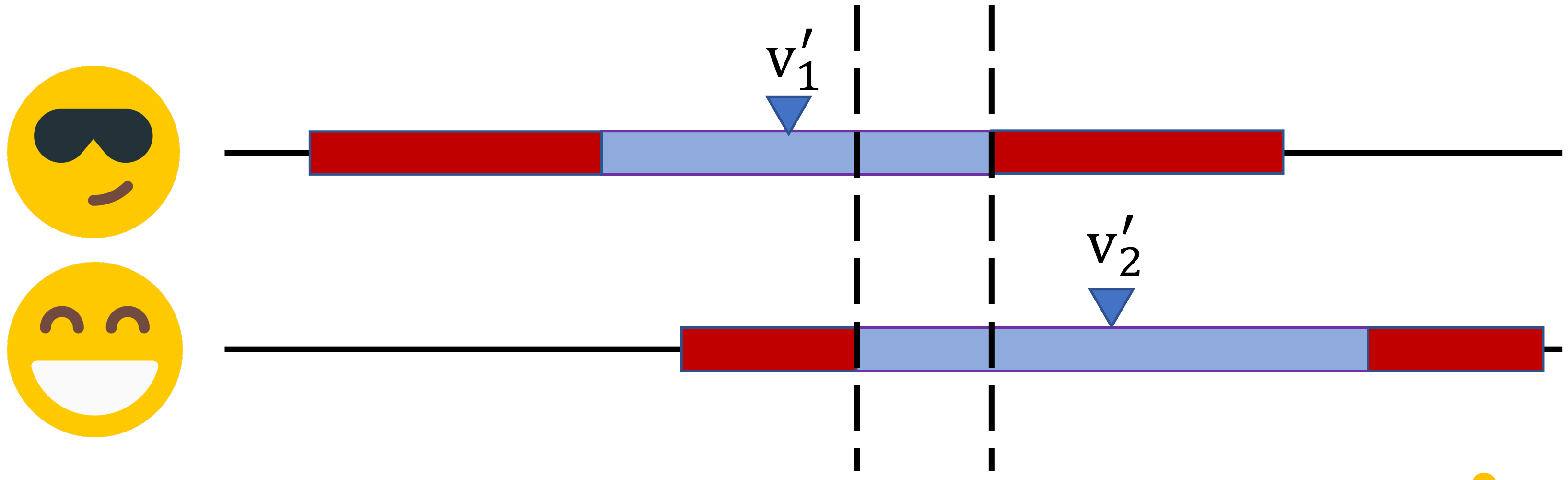
V’ = (~~4.5~~, 10, 20, 21, ~~+100000~~)
😈

V‘ = (~~4.5~~, 10, 15, 20, ~~21~~)
😈

V’ = (~~4.5~~, 10, 20, ~~21~~)

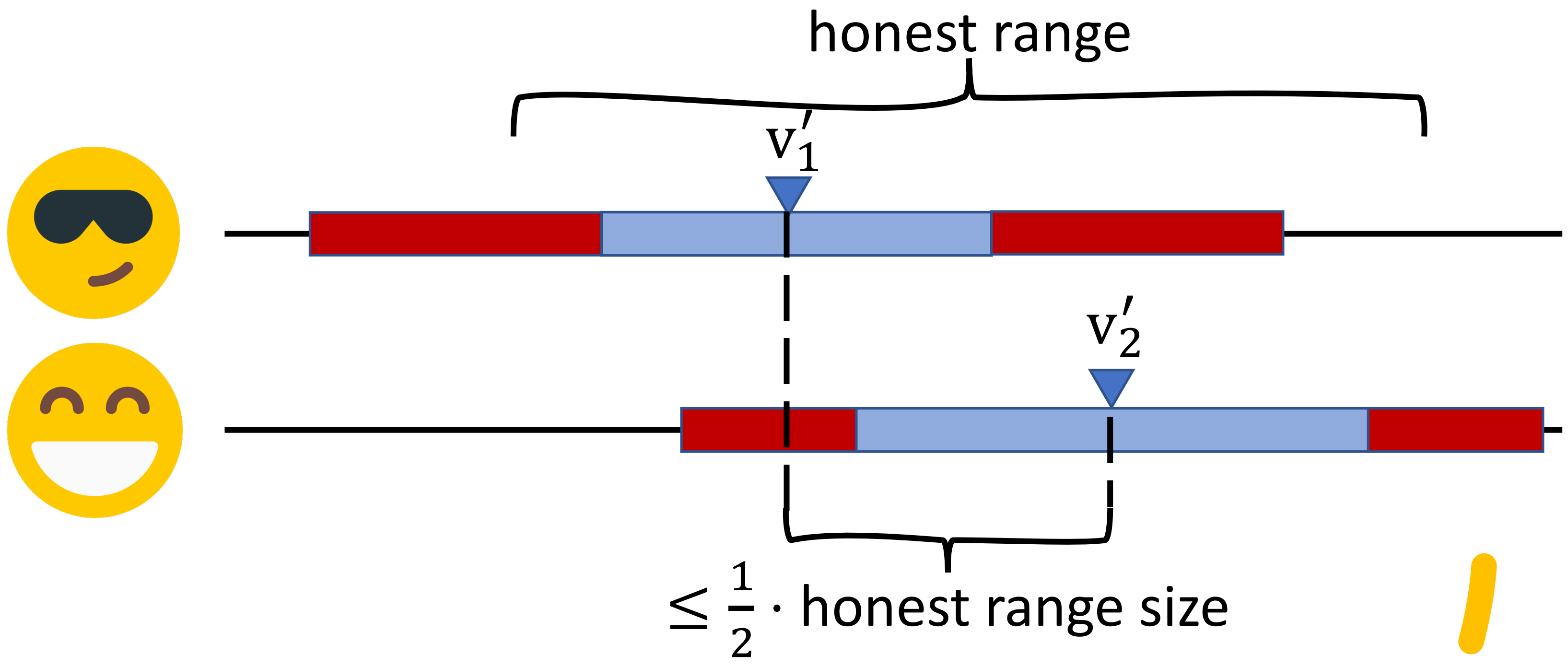
Convergence?

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



Convergence?

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



How many iterations do we need?

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.

=> $\log_2\left(\frac{B-A}{\varepsilon}\right)$ 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 $\geq$ $\mathrm{n}-\mathrm{f}$ values received.
2. Obtain V' by discarding the lowest **f** and the highest **f** values from V
3. Compute a new value $\mathrm{v}' = \frac{1}{2}(\min \mathrm{V}' + \max \mathrm{V}')$

f < n/4?

- Validity ✓
- ε-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 => 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 $\geq \mathrm{n}-\mathrm{f}$ values received.
2. Obtain V' by discarding the lowest **f** and the highest **f** values from V
3. Compute a new value $\mathrm{v}' = \frac{1}{2}(\min \mathrm{V}' + \max \mathrm{V}')$

f < n/3?

- Validity ✓
- ε-Agreement:

Honest values: *4.5, 10, 10*

- *(~~-100000~~, 4.5, ~~10~~)*
- *(~~4.5~~, 10, ~~10~~)* ✗

Is $f < n/3$ possible?

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

$\Rightarrow$ **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 $P_1, \dots, v_{n-f}$ 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**

$$\Rightarrow (v_1, P_1, v_2, P_2, \dots, v_{n-f}, P_{n-f})$$

Witness technique

Code for party P with input v:

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

   Send $(v_1, P_1, v_2, P_2, \dots, v_{n-f}, P_{n-f})$ 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 $\geq \mathrm{n}-\mathrm{f}$ values received.
2. Obtain V' by discarding the lowest **f** and the highest **f** values from V
3. Compute a new value $\mathrm{v}' = \frac{1}{2}(\min \mathrm{V}' + \max \mathrm{V}')$

f < n/3? *Optimal*

- Validity ✓
- ε-Agreement ✓

Synchronous protocol?

- Approximate Agreement is interesting here:

  #rounds does not depend on f.
- The asynchronous protocol works for $f < n/3$.
- $f < n/2$ is also possible, using signatures.

Optimal

Issues when $f < n/2$

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

(~~-100000~~, 0, ~~1~~)

(~~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, σ) to every party

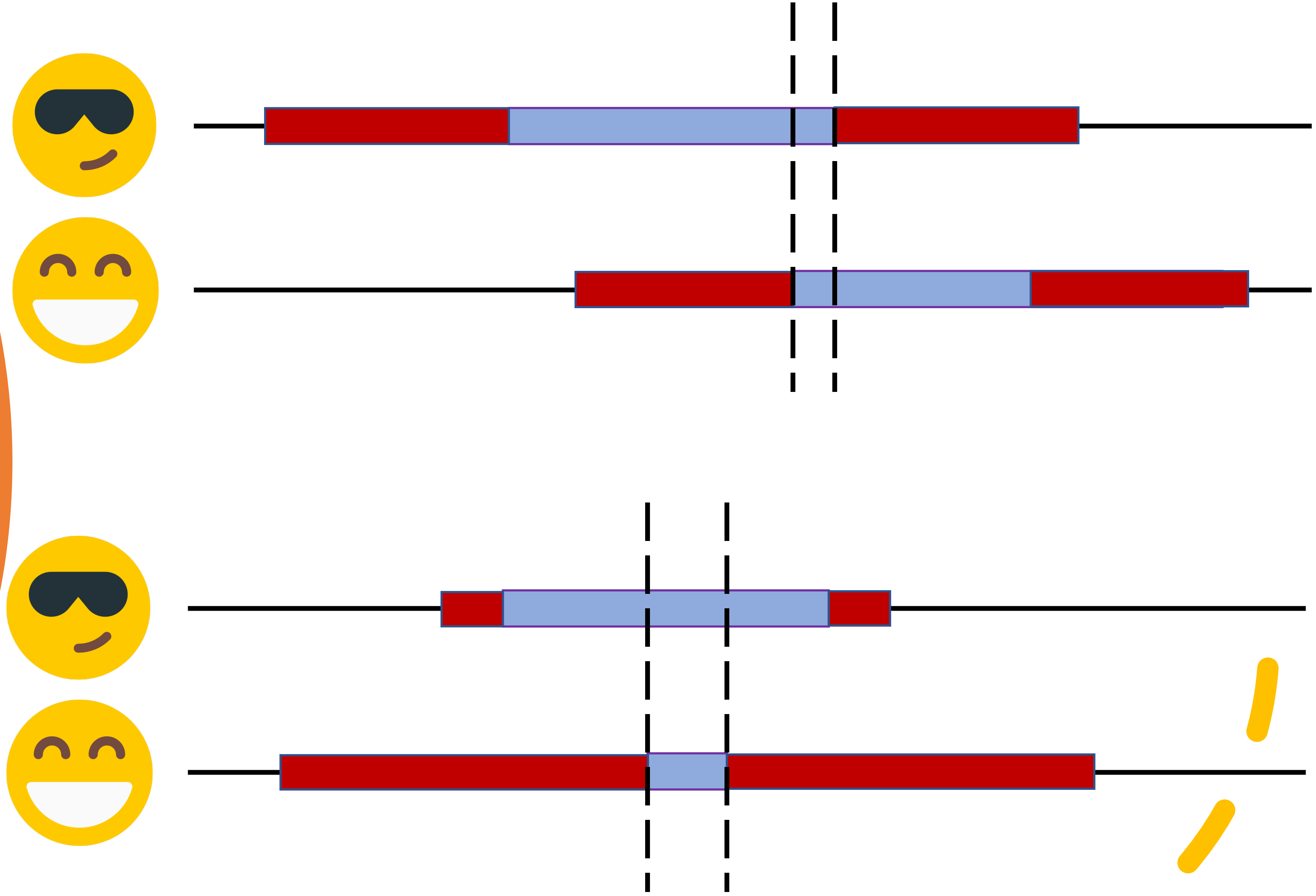
Code for receiver P:

1. If you received (v, σ) from S, forward it to every party
2. If > f parties confirmed (v, σ) 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?

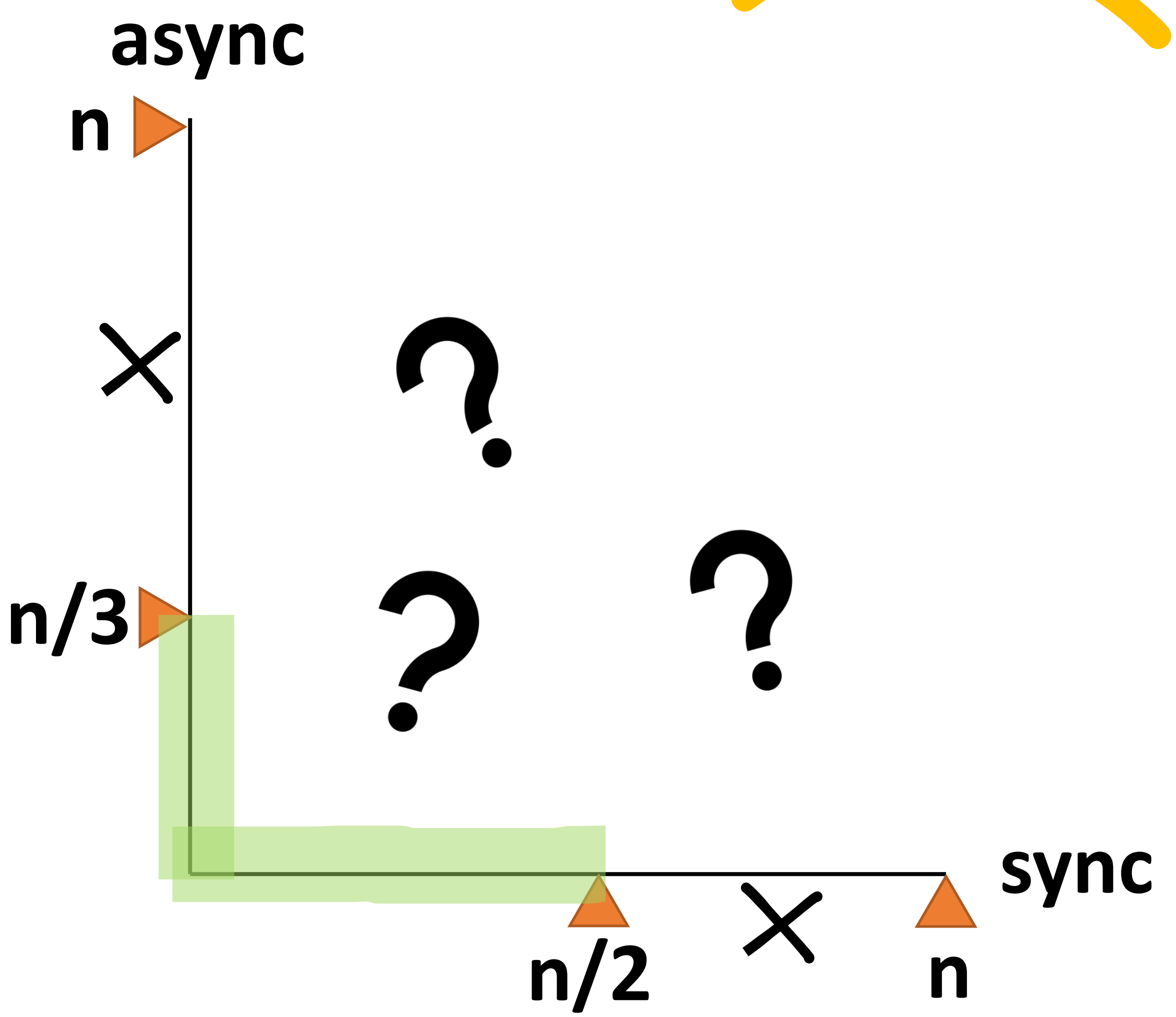


Synchronous protocol (f < n/2)

In iteration i:

1. Send your value v to everyone via **Weak Broadcast**.
   Save the $\mathbf{n} - \mathbf{f} + \mathbf{k}$ received values in V.
1. Obtain V' by discarding the lowest $\mathbf{k}$ and the highest $\mathbf{k}$ values from V.
2. Compute your new value $\mathrm{v}' = \frac{1}{2}(\min \mathrm{V}' + \max \mathrm{V}')$.

Results so far



Is there a best-of-both worlds?

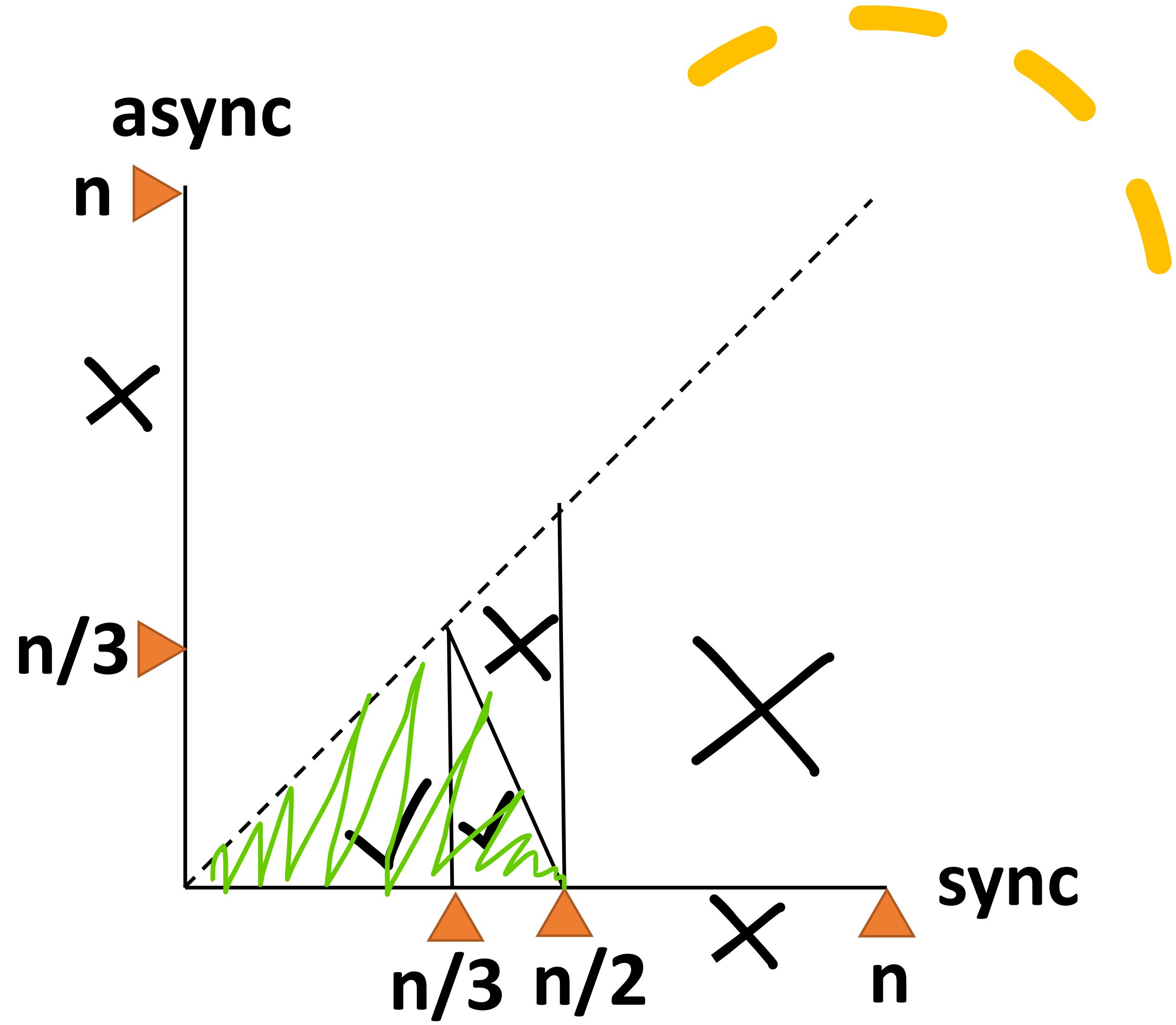
- The parties are not aware of the type of network the protocol runs in.

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

Yes!

If $\mathbf{2} \cdot \mathbf{f_s} + \mathbf{f_a} < \mathbf{n}$ (optimal).


Results


async
n
n/3
sync
n/3
n/2
n

Summary

- Approximate Agreement:
  - Allows an error of ε, 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**!