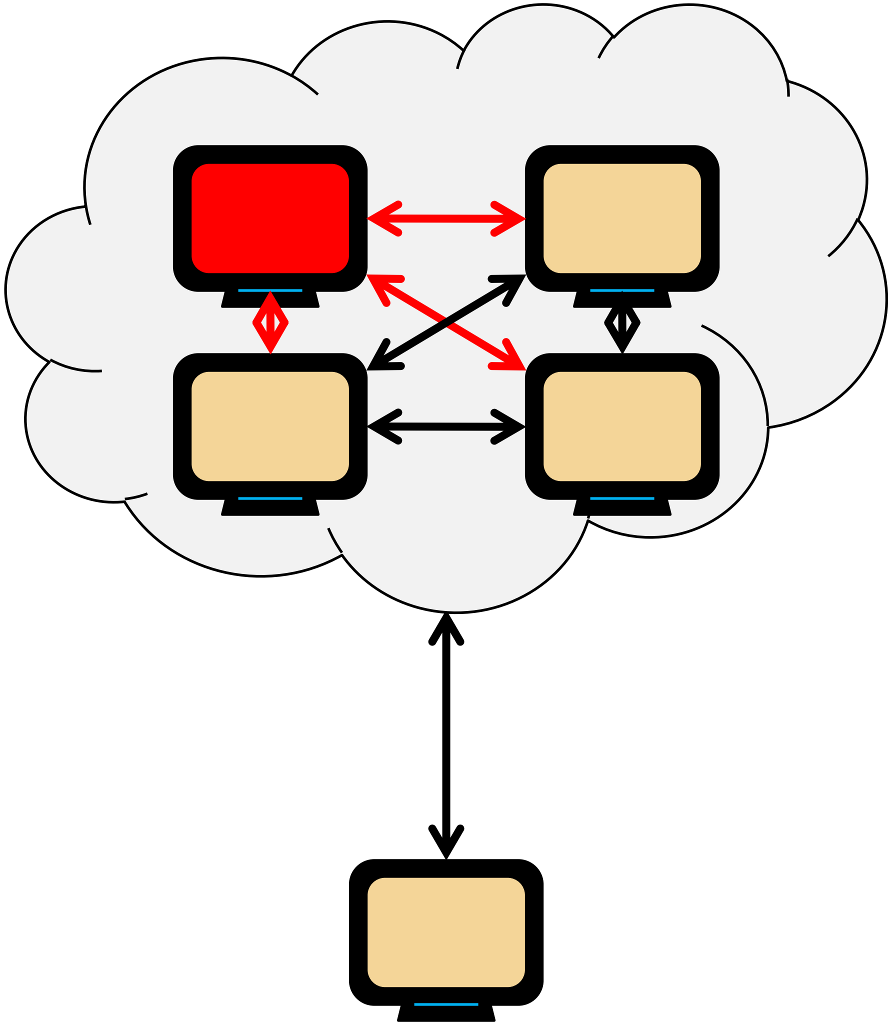
Consensus in the Asynchronous Hybrid Byzantine Model with Optimal Resilience

Megumi Ando Moses Liskov
{mando, mliskov}@mitre.org

MITRE

Byzantine fault tolerance

- Single adversary adaptively corrupts nodes
  - Corrupted (“Byzantine”) nodes send arbitrary messages
- Asynchronous network model
  - Honest messages can be delayed arbitrarily
- Lower bound: n > 3t
  - n: number of nodes
  - t: corruption limit
- Critical problems:
  - reliable broadcast
  - consensus



Crash tolerance and hybrid models

- In *crash tolerance* (fail stop)
  - Single adversary adaptively crashes nodes
  - Crashed nodes cannot send messages
  - Lower bound: $n > 2t$

Crash tolerance and hybrid models

- In *crash tolerance* (fail stop)
  – Single adversary adaptively crashes nodes
  – Crashed nodes cannot send messages
  – Lower bound: $n > 2t$

- Hybrid Byzantine model
  – Single adversary adaptively chooses to crash or corrupt nodes
  – Crashed nodes cannot send messages
  – Corrupted nodes send messages of adversary's choice
  – Up to b corruptions, up to t total crashes or corruptions

Crash tolerance and hybrid models

- In *crash tolerance* (fail stop)
  - Single adversary adaptively crashes nodes
  - Crashed nodes cannot send messages
  - Lower bound: $n > 2t$

- Hybrid Byzantine model
  - Single adversary adaptively chooses to crash or corrupt nodes
  - Crashed nodes cannot send messages
  - Corrupted nodes send messages of adversary's choice
  - Up to b corruptions, up to t total crashes or corruptions

- Our results:
  - Lower bound: $n > 2t+b$
  - Optimal size protocols for reliable broadcast and consensus

MITRE

Principles

- With n = 3t+1 nodes and up to t Byzantine:
  - Can wait for n-t responses in an asynchronous network
  - Of the ones we get responses from, at least t+1 are honest
  - t+1 honest nodes must be sufficient to force progress

MITRE

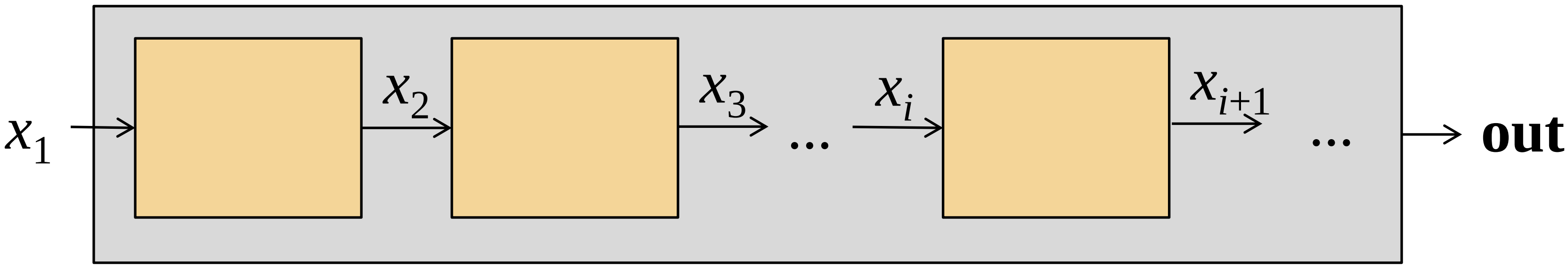
Principles

- With n = 3t+1 nodes and up to t Byzantine:
  - Can wait for n-t responses in an asynchronous network
  - Of the ones we get responses from, at least t+1 are honest
  - t+1 honest nodes must be sufficient to force progress

- With n = 2t+b+1 nodes in the Hybrid Byzantine model:
  - Can still wait for n-t responses in an asynchronous network
  - Of them, at least t+1 were honest *at the time*
  - But only b+1 must *remain* honest
  - Thus, b+1 honest nodes must be sufficient to force progress

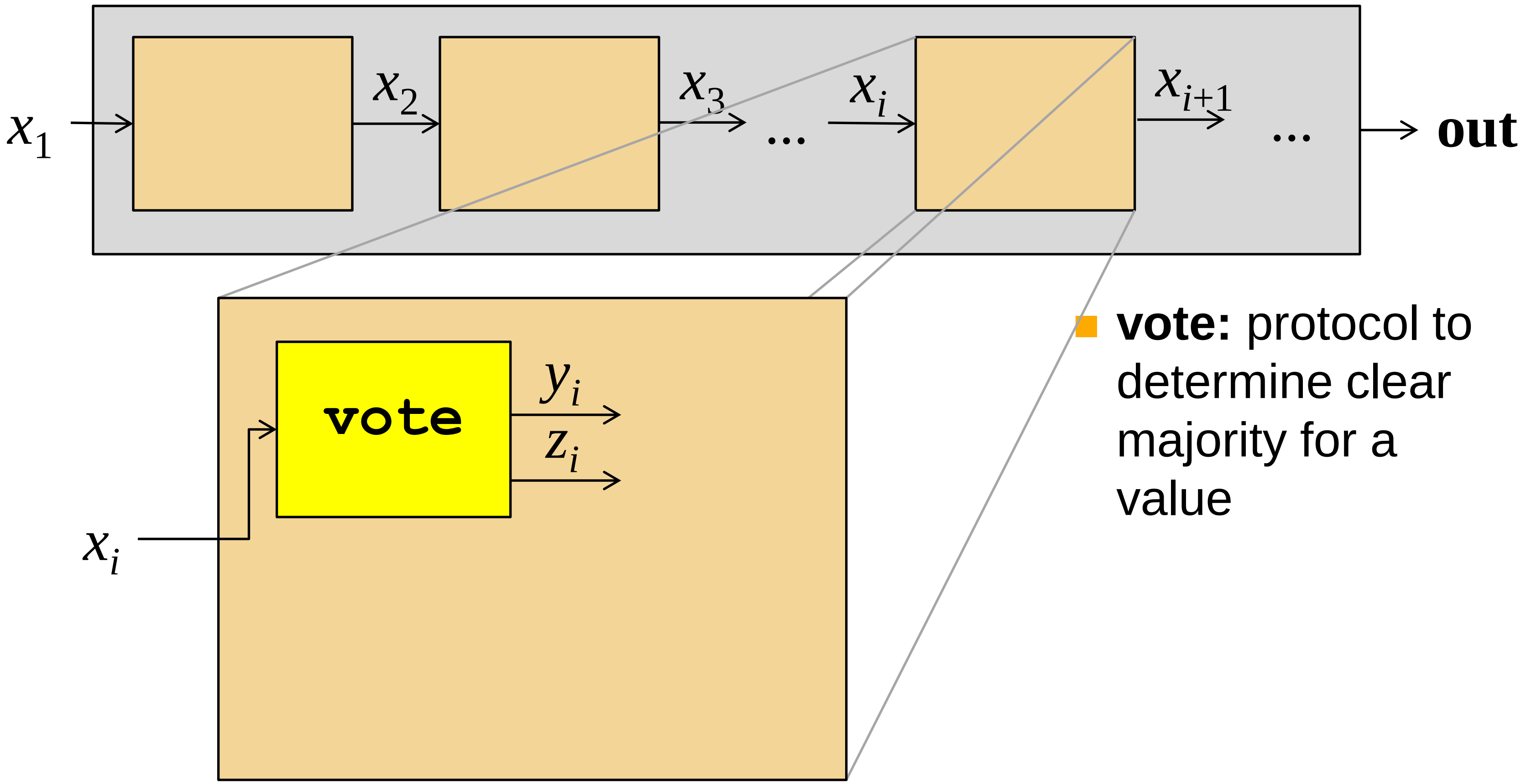
Consensus protocol: approach

Approach from Canetti and Rabin and earlier papers



Consensus protocol: approach

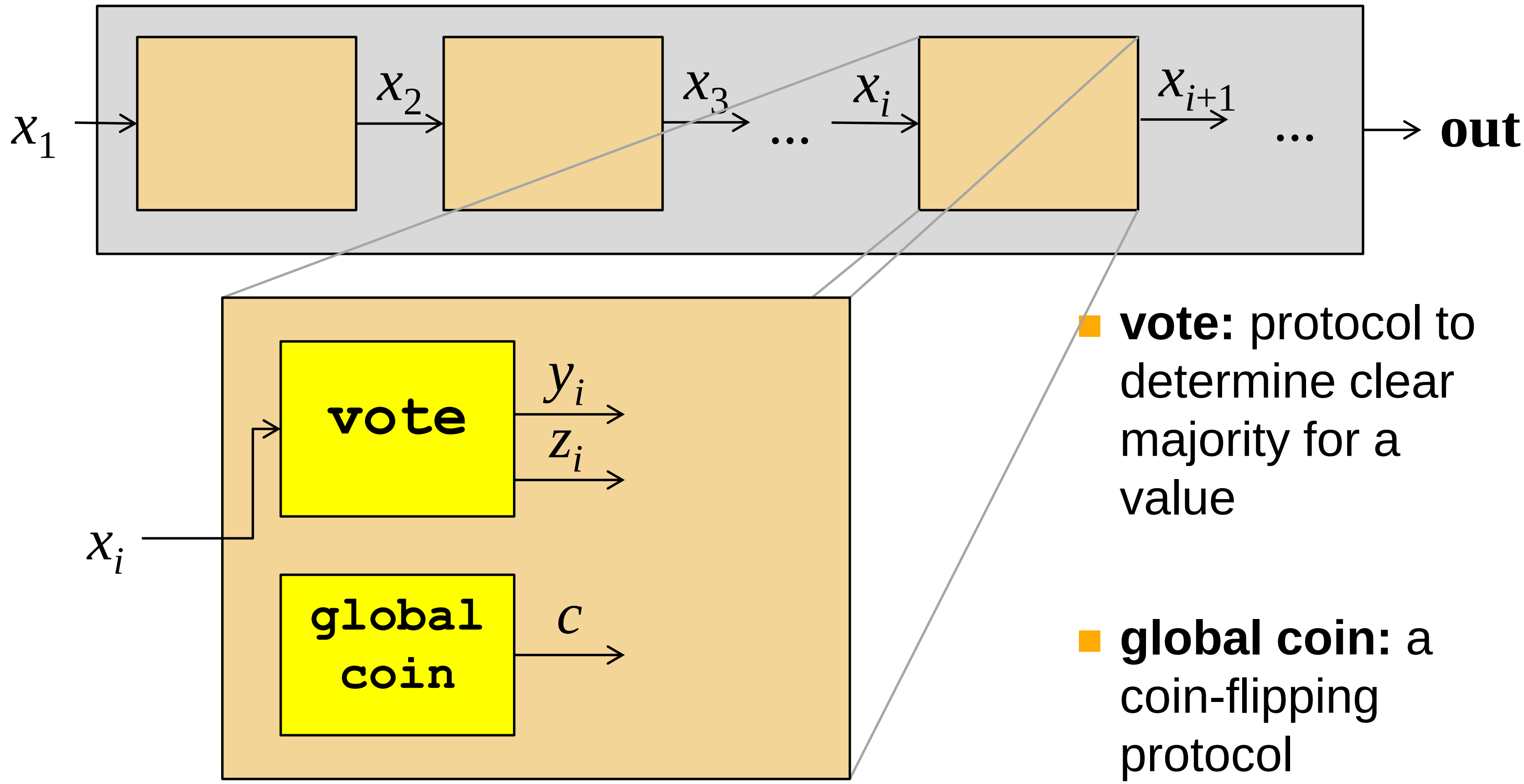
Approach from Canetti and Rabin and earlier papers



- **vote:** protocol to determine clear majority for a value

Consensus protocol: approach

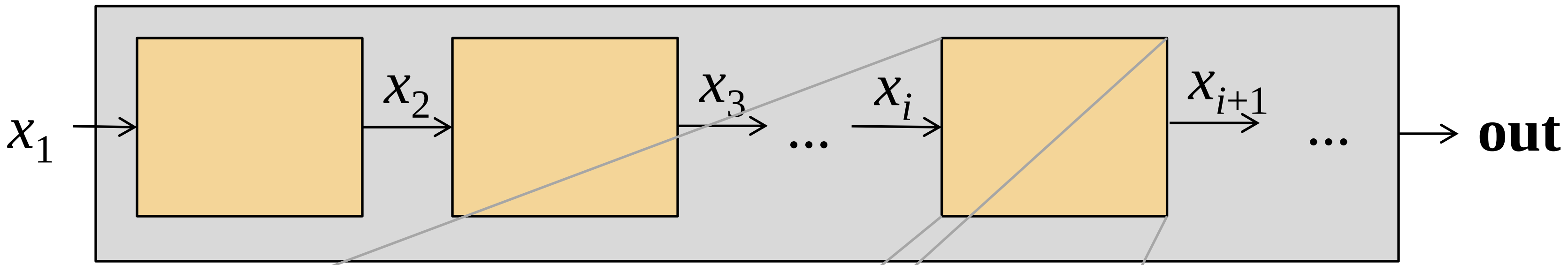
Approach from Canetti and Rabin and earlier papers

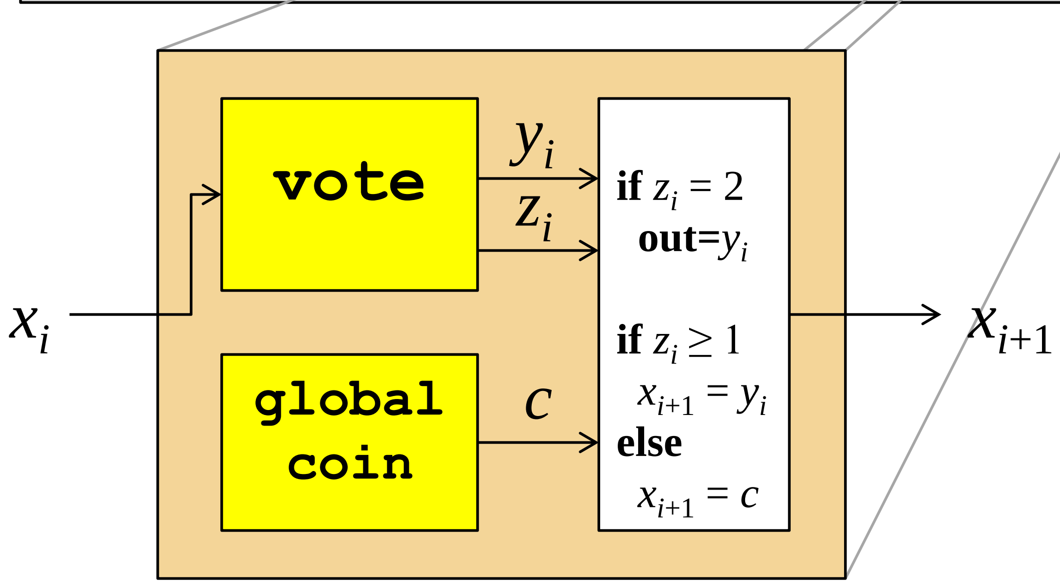


- **vote:** protocol to determine clear majority for a value
- **global coin:** a coin-flipping protocol

Consensus protocol: approach

Approach from Canetti and Rabin and earlier papers





- **vote:** protocol to determine clear majority for a value
- **global coin:** a coin-flipping protocol

Vote protocol: fully Byzantine

Protocol of Canetti-Rabin 1993: Let $n \geq 3t+1$.

Player P_i with input x_i:

1. a-cast (**input**, i, x_i)
2. Complete n-t **input** a-casts; set vote v_i to majority of input values.
3. a-cast (**vote**, i, v_i)
4. Wait to complete n-t consistent **vote** a-casts; set revote rv_i to majority of vote values.
5. a-cast (**re-vote**, i, rv_i)
6. Wait to complete n-t consistent **re-vote** a-casts.
7. If all **vote**s agree on σ, output (σ,2). Else if all **re-vote**s agree on σ, output (σ,1). Otherwise, output (0,0).

Size of intersection (n-2t) of two honest nodes' views guarantees unanimity in one is a majority in the other: **2(t+1) > n-t.**

***n*-2*t* = *b*+1 not large enough.**

Vote protocol: hybrid Byzantine

Our Protocol. Let $n \geq 2t + b + 1$.

Player P_i with input x_i:

1. a-cast (**input**, i, x_i)
2. Complete n-t **input** a-casts; set vote v_i to majority of input values.
3. a-cast (**vote**, i, v_i)
4. Wait to complete *n-t* consistent **vote** a-casts; set revote rv_i to majority of vote values.
5. a-cast (**re-vote**, i, rv_i)
6. Wait to complete *n-t* consistent **re-vote** a-casts.
7. If all **vote**s agree on σ, output (σ,2). Else if all **re-vote**s agree on σ, output (σ,1). Otherwise, output (0,0).

Vote protocol: hybrid Byzantine

Our Protocol. Let $n \geq 2t + b + 1$.

Player P_i with input x_i:

1. a-cast (**input**, i, x_i)
2. Complete n-t **input** a-casts; set vote v_i to majority of input values.
3. a-cast (**vote**, i, v_i)
4. Wait to complete n-t consistent **vote** a-casts; set revote rv_i to majority of vote values.
5. a-cast (**re-vote**, i, rv_i)
6. Wait to complete n-t consistent **re-vote** a-casts.
7. If all **vote**s agree on σ, output (σ,2). Else if all **re-vote**s agree on σ, output (σ,1). Otherwise, output (0,0).

Vote protocol: hybrid Byzantine

Our Protocol. Let $n \geq 2t + b + 1$.

Player P_i with input x_i:

1. a-cast (**input**, i, x_i)
2. Complete n-t **input** a-casts; set vote v_i to majority of input values.
3. a-cast (**vote**, i, v_i)
4. Wait to complete n-t consistent **vote** a-casts; set S_i to set of **vote** senders.
5. a-cast (**set**, i, S_i)
6. Wait to complete n-t consistent **set** a-casts; set re-vote rv_i to majority of votes from members of all sets.
7. a-cast (**re-vote**, i, rv_i)
8. Wait to complete n-t consistent **re-vote** a-casts.
9. If all **vote**s agree on σ, output (σ,2). Else if all **re-vote**s agree on σ, output (σ,1). Otherwise, output (0,0).

set messages guarantee the intersection of two honest nodes' views has size at least *n-t*.