What voting gives us

- You might get system-wide agreement or you might get stuck
- Can’t vote directly on consensus question (i.e., log entry)
- What can we vote on without jeopardizing liveness?
  1. Statements that never get stuck (irrefutable), and
  2. Statements whose hold on consensus question can be broken if stuck (neutralizable)
Paxos [Lamport]

- **A ballot is a pair** \( \langle n, x \rangle \)
  - \( n \) – a counter to ensure arbitrarily many ballots exist
  - \( x \) – a candidate output value for the consensus protocol

- Conceptually vote to **commit** and **abort** ballots
  - If a quorum votes to commit \( \langle n, x \rangle \) for any \( n \), it is safe to output \( x \)

- **Invariant:** all committed and stuck ballots must have same \( x \)

- To preserve: can’t vote to commit a ballot before **preparing** it
  - Prepare \( \langle n, x \rangle \) by aborting all \( \langle n', x' \rangle \) with \( n' \leq n \) and \( x' \neq x \).
  - **PREPARED** message votes to abort all lower ballots not containing \( x \)
    (or all lower ballots period if previous is **NULL**)

- **If ballot** \( \langle n, x \rangle \) **stuck**, **neutralize by restarting with** \( \langle n + 1, x \rangle \)
  - Can prepare \( \langle n + 1, x \rangle \) even if \( \langle n, x \rangle \) is stuck
Paxos example

0. Initially, all ballots are bivalent

1. Agree that \( \langle 1, g \rangle \) is prepared and vote to commit it

2. Lose vote on \( \langle 1, g \rangle \); agree \( \langle 2, f \rangle \) prepared and vote to commit it

3. \( \langle 2, f \rangle \) is stuck, so agree \( \langle 3, f \rangle \) prepared and vote to commit it

4. See \( T \) votes to commit \( \langle 3, f \rangle \) (commit-valent) and externalize \( f \)
   - At this point nobody cares about \( \langle 2, f \rangle \)—neutralized

5. Node failure makes \( \langle 3, f \rangle \) stuck, prepare and commit \( \langle 4, f \rangle \)
Paxos example

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0. Initially, all ballots are bivalent

1. Agree that \(\langle 1, g \rangle\) is prepared and vote to commit it

2. Lose vote on \(\langle 1, g \rangle\); agree \(\langle 2, f \rangle\) prepared and vote to commit it

3. \(\langle 2, f \rangle\) is stuck, so agree \(\langle 3, f \rangle\) prepared and vote to commit it

4. See \(T\) votes to commit \(\langle 3, f \rangle\) (commit-valent) and externalize \(f\)
   - At this point nobody cares about \(\langle 2, f \rangle\)—neutralized

5. Node failure makes \(\langle 3, f \rangle\) stuck, prepare and commit \(\langle 4, f \rangle\)
### Paxos example

**Candidate values**

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

#### Counter

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>4</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

#### Voting Sequence

0. Initially, all ballots are bivalent

1. Agree that \( \langle 1, g \rangle \) is prepared and vote to commit it

2. Lose vote on \( \langle 1, g \rangle \); agree \( \langle 2, f \rangle \) prepared and vote to commit it

3. \( \langle 2, f \rangle \) is stuck, so agree \( \langle 3, f \rangle \) prepared and vote to commit it

4. See \( T \) votes to commit \( \langle 3, f \rangle \) (commit-valent) and externalize \( f \)
   - At this point nobody cares about \( \langle 2, f \rangle \)—neutralized

5. Node failure makes \( \langle 3, f \rangle \) stuck, prepare and commit \( \langle 4, f \rangle \)
Paxos example

0. Initially, all ballots are bivalent
1. Agree that $\langle 1, g \rangle$ is prepared and vote to commit it
2. Lose vote on $\langle 1, g \rangle$; agree $\langle 2, f \rangle$ prepared and vote to commit it
3. $\langle 2, f \rangle$ is stuck, so agree $\langle 3, f \rangle$ prepared and vote to commit it
4. See $T$ votes to commit $\langle 3, f \rangle$ (commit-valent) and externalize $f$
   - At this point nobody cares about $\langle 2, f \rangle$—neutralized
5. Node failure makes $\langle 3, f \rangle$ stuck, prepare and commit $\langle 4, f \rangle$
### Paxos example

<table>
<thead>
<tr>
<th>counter</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>?</td>
<td>x</td>
</tr>
<tr>
<td>3</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>?</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

#### Candidate values

- `a`
- `b`
- `c`
- `d`
- `e`
- `f`
- `g`
- `h`

0. Initially, all ballots are bivalent

1. Agree that \(\langle 1, g \rangle\) is prepared and vote to commit it

2. Lose vote on \(\langle 1, g \rangle\); agree \(\langle 2, f \rangle\) prepared and vote to commit it

3. \(\langle 2, f \rangle\) is stuck, so agree \(\langle 3, f \rangle\) prepared and vote to commit it

4. See \(T\) votes to commit \(\langle 3, f \rangle\) (commit-valent) and externalize \(f\)
   - At this point nobody cares about \(\langle 2, f \rangle\)—neutralized

5. Node failure makes \(\langle 3, f \rangle\) stuck, prepare and commit \(\langle 4, f \rangle\)
### Paxos example

**Table: Candidate Values**

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>?</td>
<td>x</td>
</tr>
<tr>
<td>3</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>4</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
</tbody>
</table>

**0.** Initially, all ballots are bivalent

**1.** Agree that $\langle 1, g \rangle$ is prepared and vote to commit it

**2.** Lose vote on $\langle 1, g \rangle$; agree $\langle 2, f \rangle$ prepared and vote to commit it

**3.** $\langle 2, f \rangle$ is stuck, so agree $\langle 3, f \rangle$ prepared and vote to commit it

**4.** See $T$ votes to commit $\langle 3, f \rangle$ (commit-valent) and externalize $f$
   - At this point nobody cares about $\langle 2, f \rangle$—neutralized

**5.** Node failure makes $\langle 3, f \rangle$ stuck, prepare and commit $\langle 4, f \rangle$
• Instead of voting on \( \text{op}_1 \), \ldots directly, vote on \( \langle \text{view 1}, \text{op}_1 \rangle \), \ldots
  - Each \( \langle \text{view}, \text{op} \rangle \) selected by a single leader for view, so irrefutable
  - E.g., chose leader by round-robin using \( \text{view} \# \mod N \)

• What if votes on \( \text{op}_4 \) and \( \text{op}_5 \) are stuck (e.g., leader fails)?
  - Neutralize by agreeing view 1 had only 3 meaningful operations
  - Vote to form view 2 that immediately follows \( \langle \text{view 1}, \text{op}_3 \rangle \)

• Failed to form view 2 (e.g., because a node wants \( \langle \text{view 1}, \text{op}_4 \rangle \))?  
  - Just go on to form view 3 after \( \langle \text{view 1}, \text{op}_4 \rangle \)
Viewstamped replication [Oki]

- Instead of voting on $\text{op}_1, \ldots$ directly, vote on $\langle \text{view } 1, \text{op}_1 \rangle, \ldots$
  - Each $\langle \text{view}, \text{op} \rangle$ selected by a single leader for view, so irrefutable
  - E.g., chose leader by round-robin using $\text{view}\# \mod N$

- What if votes on $\text{op}_4$ and $\text{op}_5$ are stuck (e.g., leader fails)?
  - Neutralize by agreeing view 1 had only 3 meaningful operations
  - Vote to form view 2 that immediately follows $\langle \text{view } 1, \text{op}_3 \rangle$

- Failed to form view 2 (e.g., because a node wants $\langle \text{view } 1, \text{op}_4 \rangle$)?
  - Just go on to form view 3 after $\langle \text{view } 1, \text{op}_4 \rangle$
View stamped replication [Oki]

- **Instead of voting on** $\text{op}_1, \ldots$ **directly, vote on** $\langle \text{view }1, \text{op}_1 \rangle, \ldots$
  - Each $\langle \text{view, op} \rangle$ selected by a single *leader* for view, so irrefutable
  - E.g., chose leader by round-robin using $\text{view# mod } N$

- **What if votes on** $\text{op}_4$ **and** $\text{op}_5$ **are stuck** (e.g., leader fails)?
  - Neutralize by agreeing view 1 had only 3 meaningful operations
  - Vote to form view 2 that immediately follows $\langle \text{view }1, \text{op}_3 \rangle$

- **Failed to form view 2** (e.g., because a node wants $\langle \text{view }1, \text{op}_4 \rangle$)?
  - Just go on to form view 3 after $\langle \text{view }1, \text{op}_4 \rangle$