

# Distributed Systems

## Exercise



# Multi-Paxos

Proposer

Acceptor 1

Acceptor 2

Acceptor 3

$$x_{last} = 0$$

$$n_{last} = 0$$

$$n_{max} = 0$$

$$x_{last} = 0$$

$$n_{last} = 0$$

$$n_{max} = 0$$

$$x_{last} = 0$$

$$n_{last} = 0$$

$$n_{max} = 0$$

# Multi-Paxos

Proposer

Acceptor 1

Acceptor 2

Acceptor 3

1. Prepare  
( $n=5$ ,  $x=8$ )

$$x_{last} = 0$$

$$n_{last} = 0$$

$$n_{max} = 5$$

$$x_{last} = 0$$

$$n_{last} = 0$$

$$n_{max} = 5$$

$$x_{last} = 0$$

$$n_{last} = 0$$

$$n_{max} = 5$$

# Multi-Paxos

Proposer

Acceptor 1

Acceptor 2

Acceptor 3

1. Prepare  
( $n=5$ ,  $x=8$ )

$$\begin{aligned}x_{last} &= 0 \\ n_{last} &= 0 \\ n_{max} &= 5\end{aligned}$$

$$\begin{aligned}x_{last} &= 0 \\ n_{last} &= 0 \\ n_{max} &= 5\end{aligned}$$

$$\begin{aligned}x_{last} &= 0 \\ n_{last} &= 0 \\ n_{max} &= 5\end{aligned}$$

1. Propose  
( $n=5$ ,  $x=8$ )

$$\begin{aligned}x_{last} &= 8 \\ n_{last} &= 5 \\ n_{max} &= 5\end{aligned}$$

$$\begin{aligned}x_{last} &= 8 \\ n_{last} &= 5 \\ n_{max} &= 5\end{aligned}$$

$$\begin{aligned}x_{last} &= 8 \\ n_{last} &= 5 \\ n_{max} &= 5\end{aligned}$$

# Multi-Paxos

Proposer

Acceptor 1

Acceptor 2

Acceptor 3

1. Prepare  
( $n=5$ ,  $x=8$ )

$$\begin{aligned}x_{last} &= 0 \\ n_{last} &= 0 \\ n_{max} &= 5\end{aligned}$$

$$\begin{aligned}x_{last} &= 0 \\ n_{last} &= 0 \\ n_{max} &= 5\end{aligned}$$

$$\begin{aligned}x_{last} &= 0 \\ n_{last} &= 0 \\ n_{max} &= 5\end{aligned}$$

1. Propose  
( $n=5$ ,  $x=8$ )

$$\begin{aligned}x_{last} &= 8 \\ n_{last} &= 5 \\ n_{max} &= 5\end{aligned}$$

$$\begin{aligned}x_{last} &= 8 \\ n_{last} &= 5 \\ n_{max} &= 5\end{aligned}$$

$$\begin{aligned}x_{last} &= 8 \\ n_{last} &= 5 \\ n_{max} &= 5\end{aligned}$$

2. Propose:  
( $n=5$ ,  $x=2$ )

$$\begin{aligned}x_{last} &= 2 \\ n_{last} &= 5 \\ n_{max} &= 5\end{aligned}$$

$$\begin{aligned}x_{last} &= 2 \\ n_{last} &= 5 \\ n_{max} &= 5\end{aligned}$$

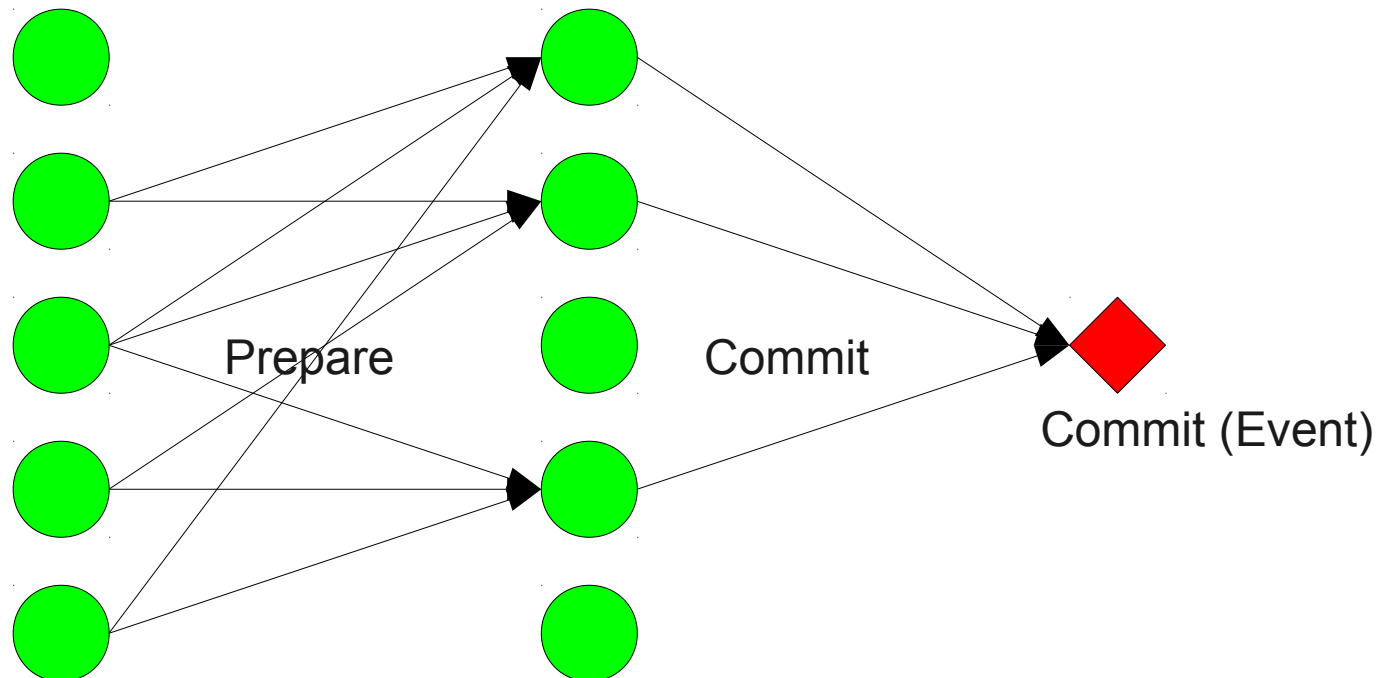
$$\begin{aligned}x_{last} &= 2 \\ n_{last} &= 5 \\ n_{max} &= 5\end{aligned}$$

# Multi-Paxos

- Another proposer?
  - Has to use another sequence number  $n!$

# PBFT: Prepare

- Orders requests in a view
- Collect so many servers in prepare-state, that there is no place for any other action.



# PBFT: Prepare → Commit

- Slides

- $n = 3f + 1$
- if backup **accepted pre-prepare** and  **$2f$  prepare** messages, it multicasts **commit** message.
- $2f + 1 = \text{quorum!}$

- Exercise

- $n = kf + 1$
- if backup **accepted pre-prepare** and **quorum prepare** messages, it multicasts **commit** message.