

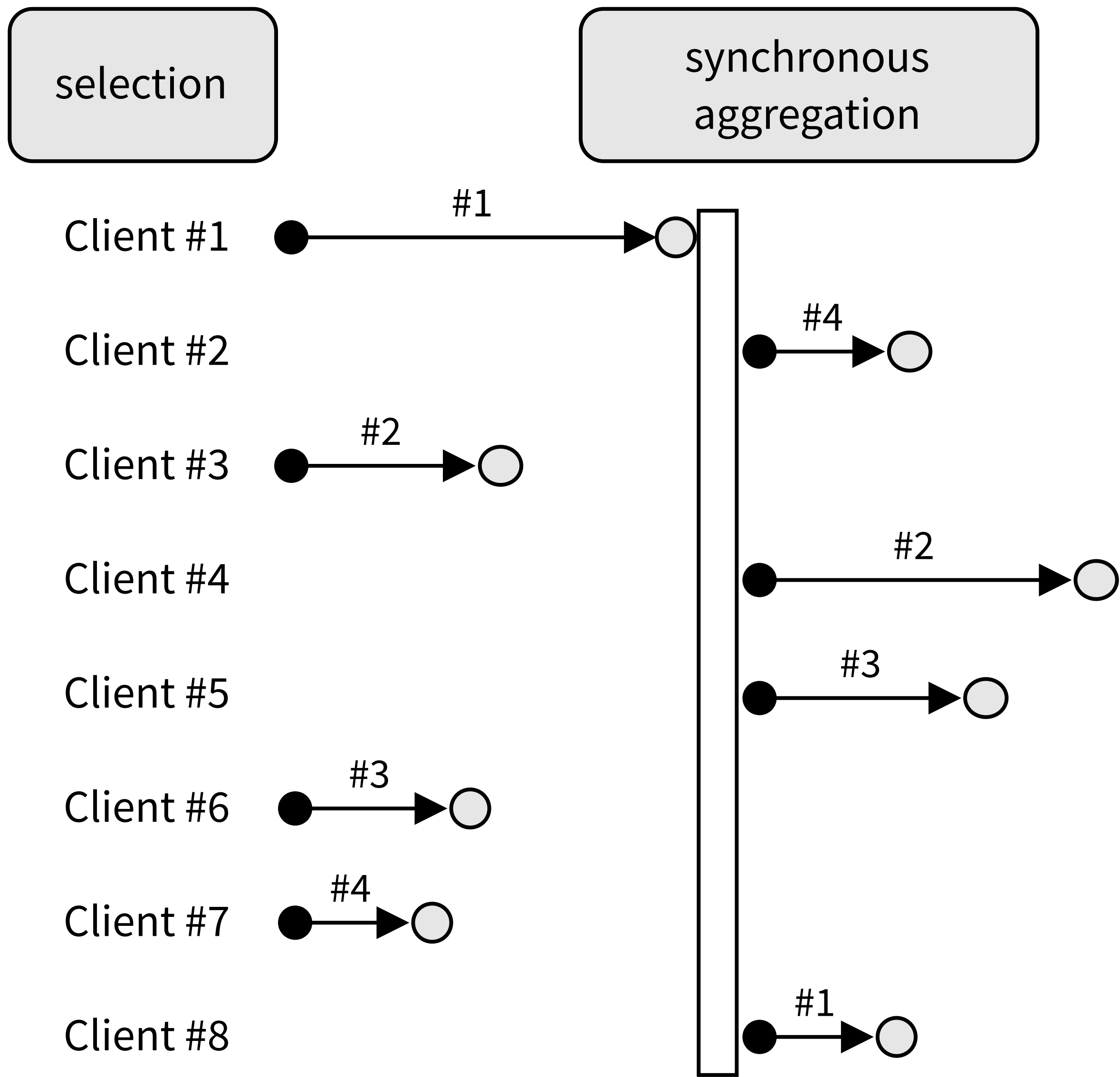
# How **Asynchronous** Should Federated Learning Be?

Ningxin Su and Baochun Li

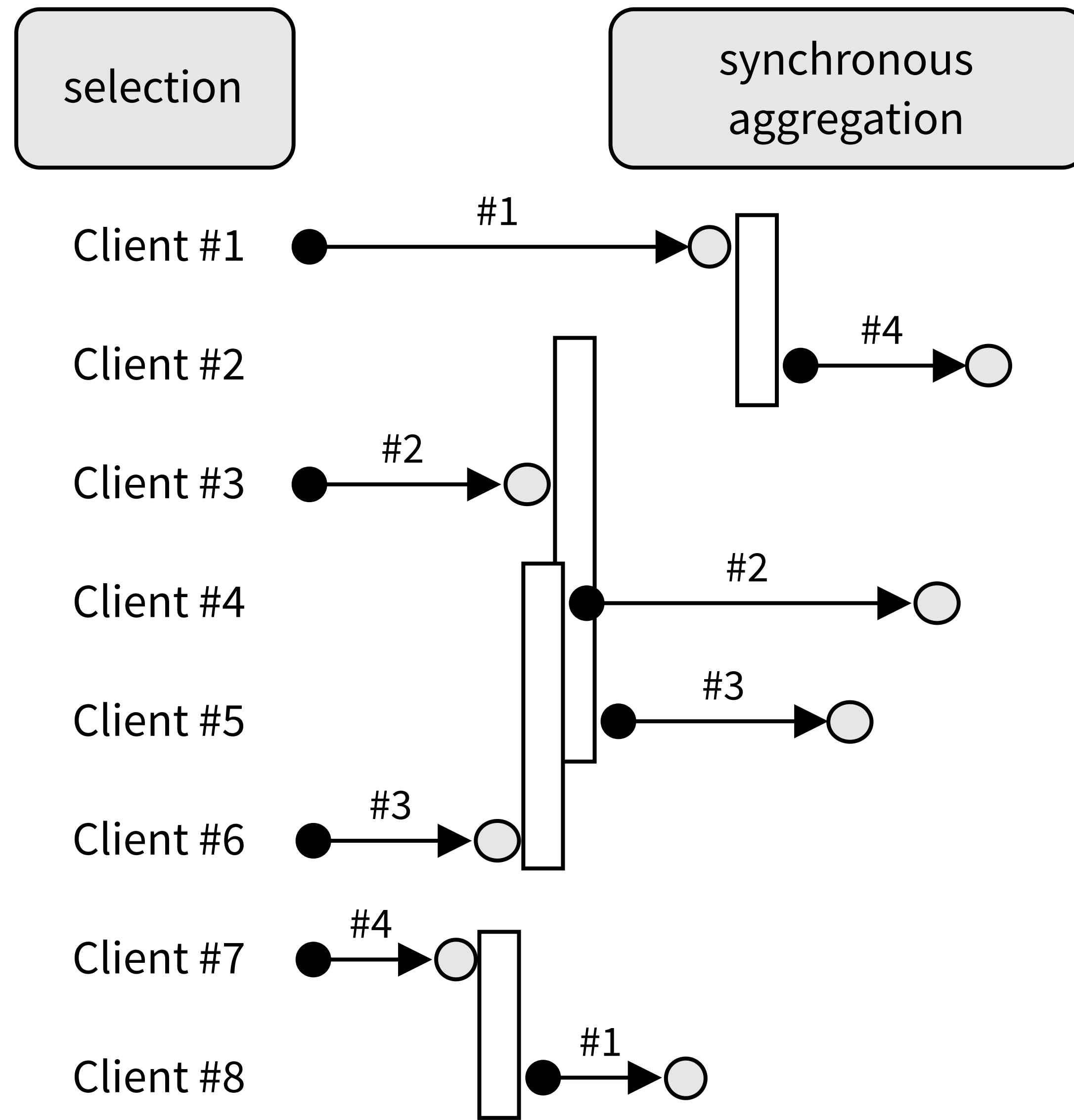
Edward S. Rogers Sr. Department of Electrical and Computer Engineering

University of Toronto

**Most existing papers assume  
synchronous federated learning, but  
shouldn't it be asynchronous?**



(a) **synchronous** federated learning



(b) **asynchronous** federated learning

# Several existing papers in the literature

## FedAsync

C. Xie, S. Koyejo, and I. Gupta, “Asynchronous Federated Optimization,” in Proc. NeurIPS Workshop on Optimization for Machine Learning (OPT), 2020.

## FedBuff

J. Nguyen, K. Malik, H. Zhan, *et al.*, “Federated Learning with Buffered Asynchronous Aggregation,” in Proc. ICML, 2021.

**Existing papers were point solutions  
in the design space for asynchronous  
federated learning**

# The design space

# The minimum number of clients

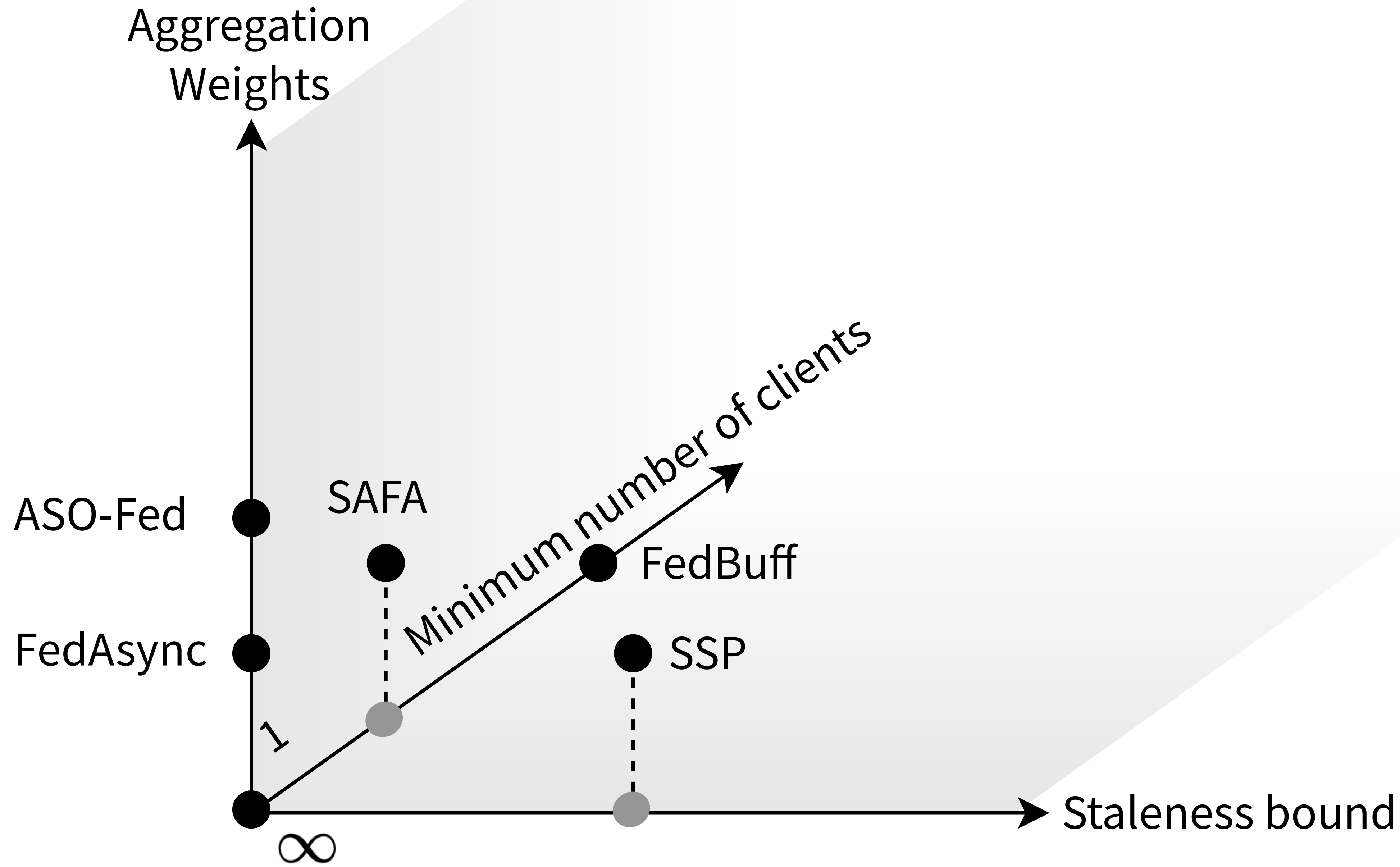
# The minimum number of clients

The minimum number of clients required to report before the server starts to aggregate these clients

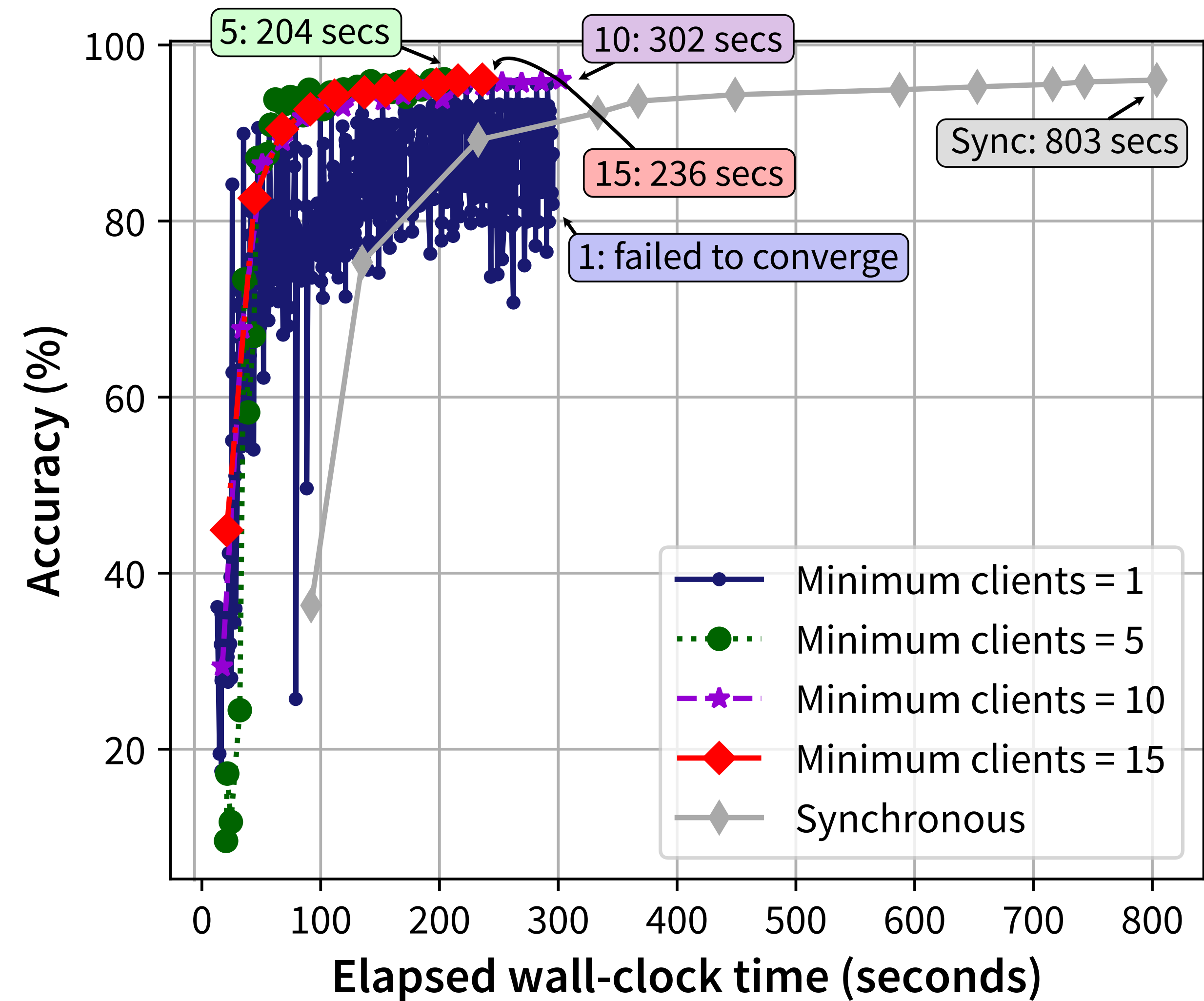


# The staleness bound

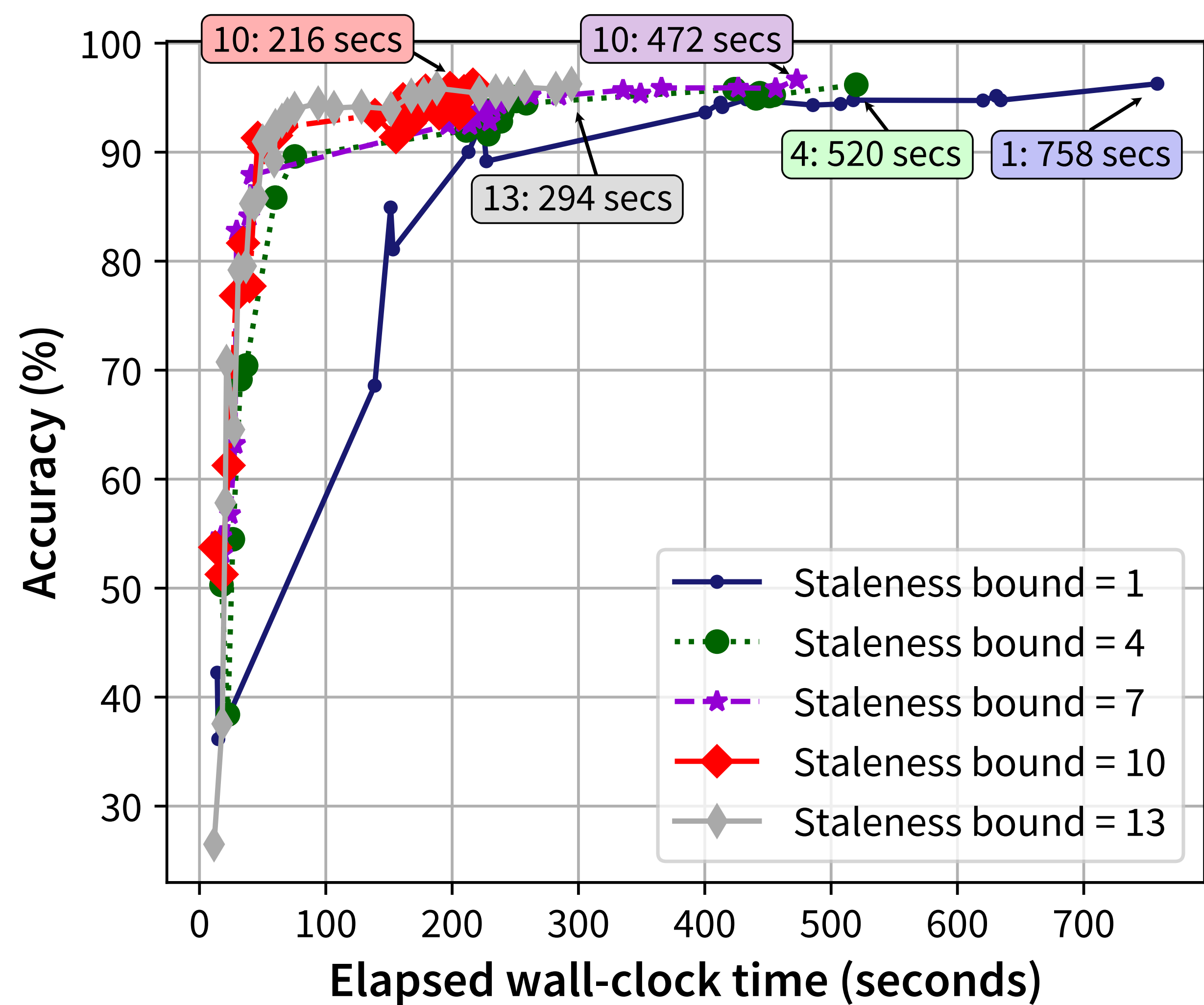
The **staleness bound** has been proposed in *synchronous parallel mechanism* (SSP): stale clients beyond a certain bound are waited for during the aggregation process



**But are there *sweet spots* in the design space?**



(a) Varying number of clients required before aggregation



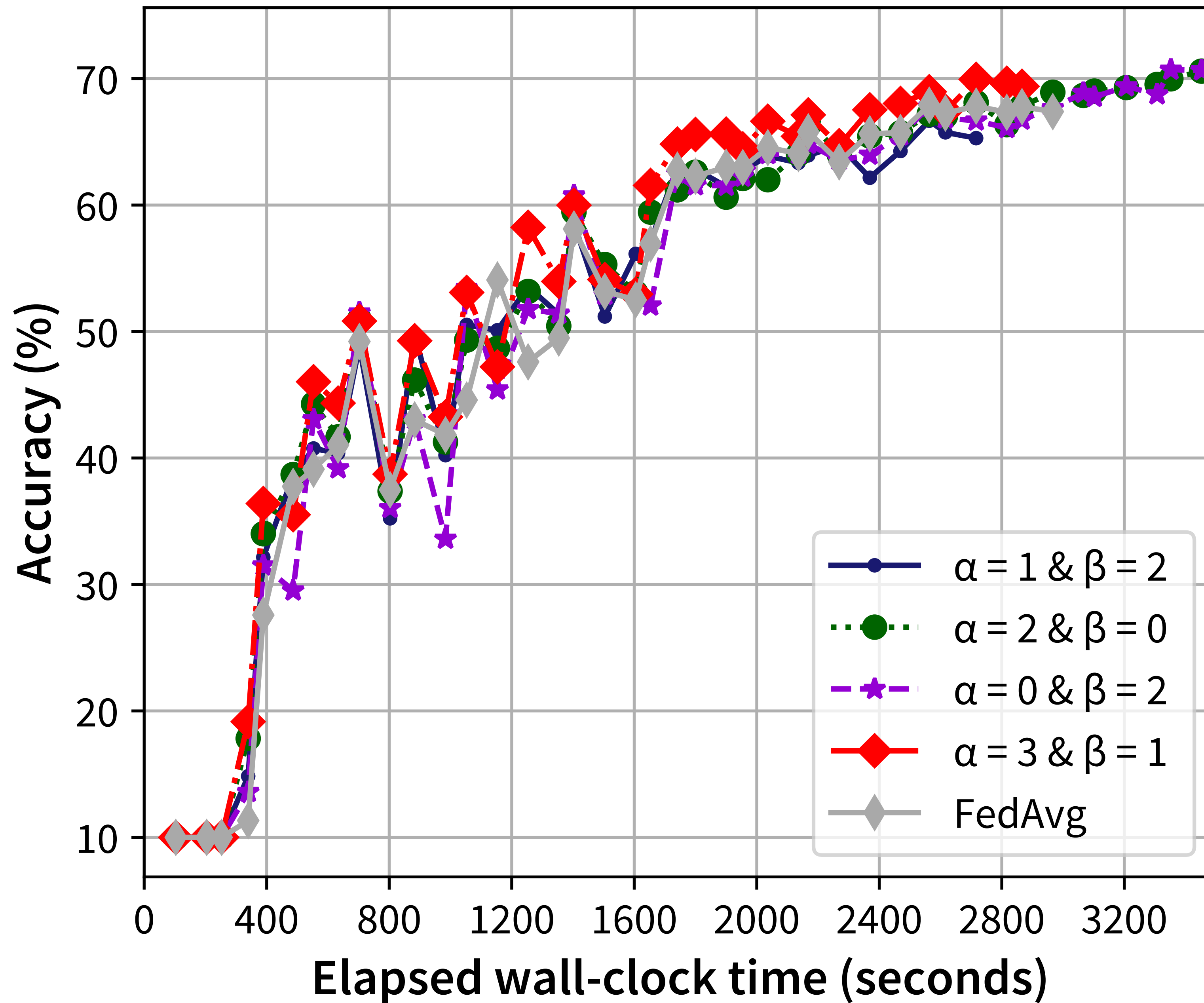
(b) Varying bounds of staleness

**Port:** our proposed mechanism

$$p_{\tau}^k = \frac{|D_k|}{|D|} \left( \alpha \cdot \frac{\Omega}{S^k + \Omega} + \beta \frac{\Theta(\Delta_{\tau}^k, w_{\tau} - w_{\tau-1}) + 1}{2} \right)$$

Staleness Cosine similarity Client's model update

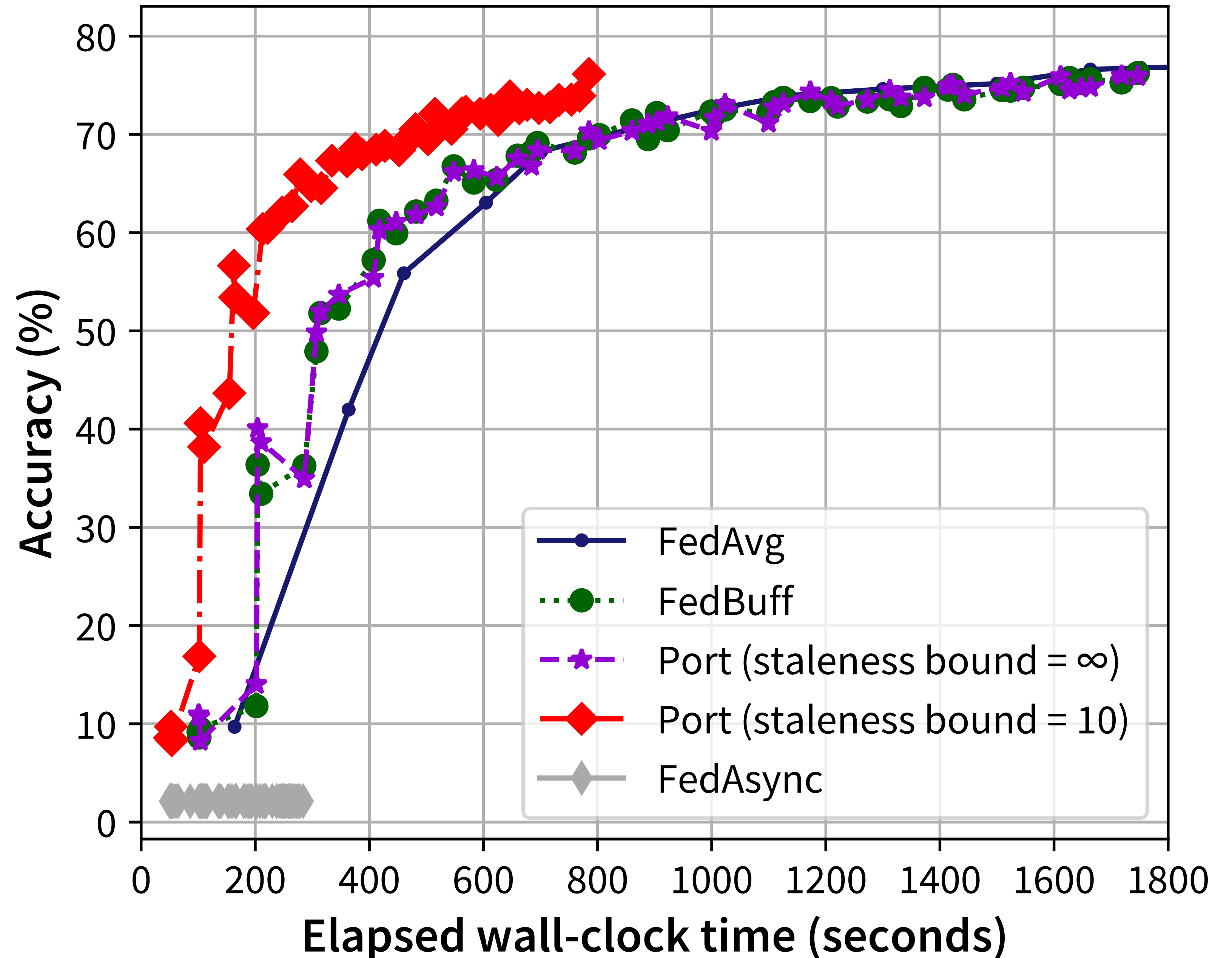
# Evaluating $\alpha$ and $\beta$ with CIFAR10



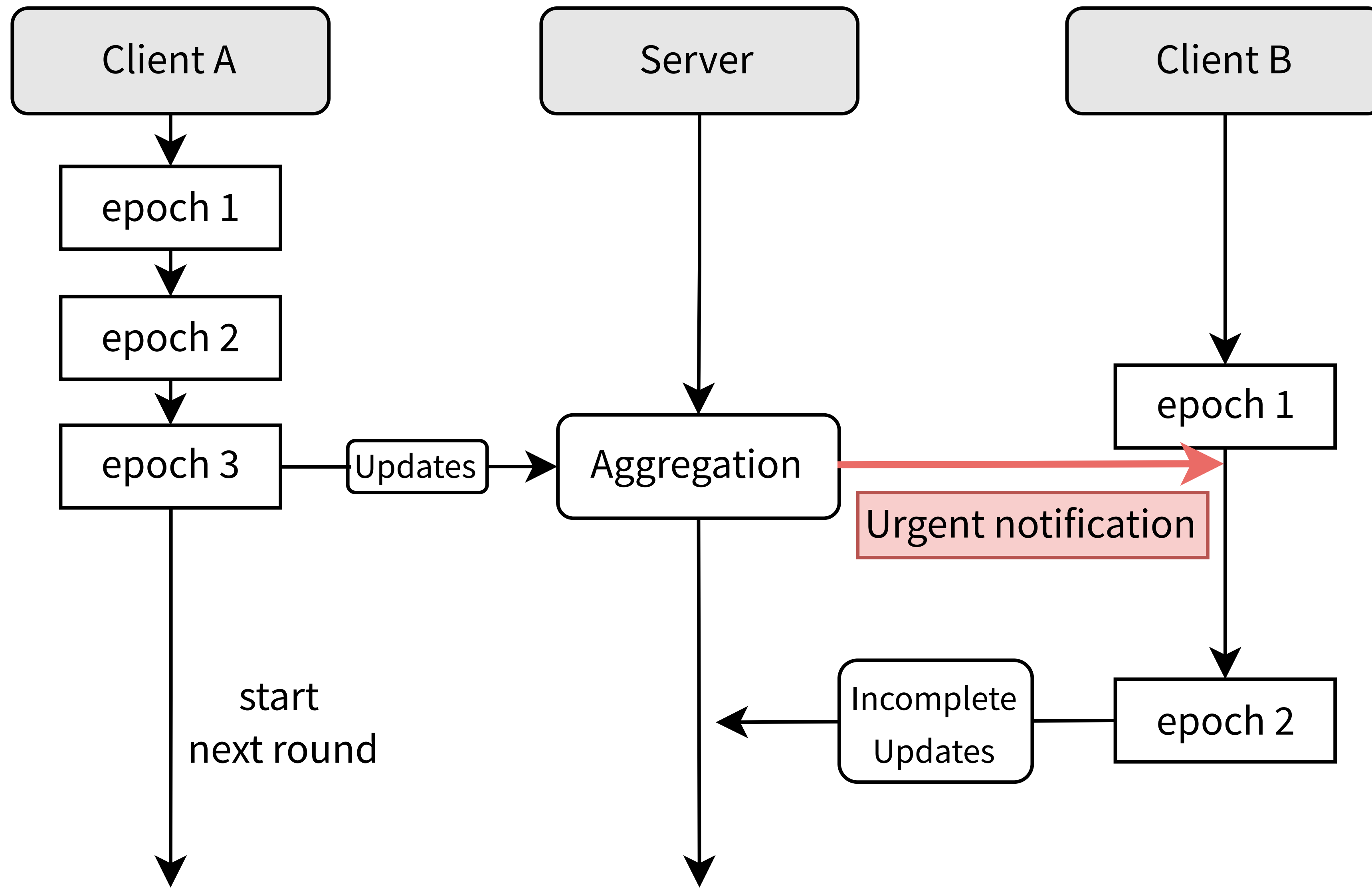
# **Port and other competitors**



# Port vs. its competitors with EMNIST

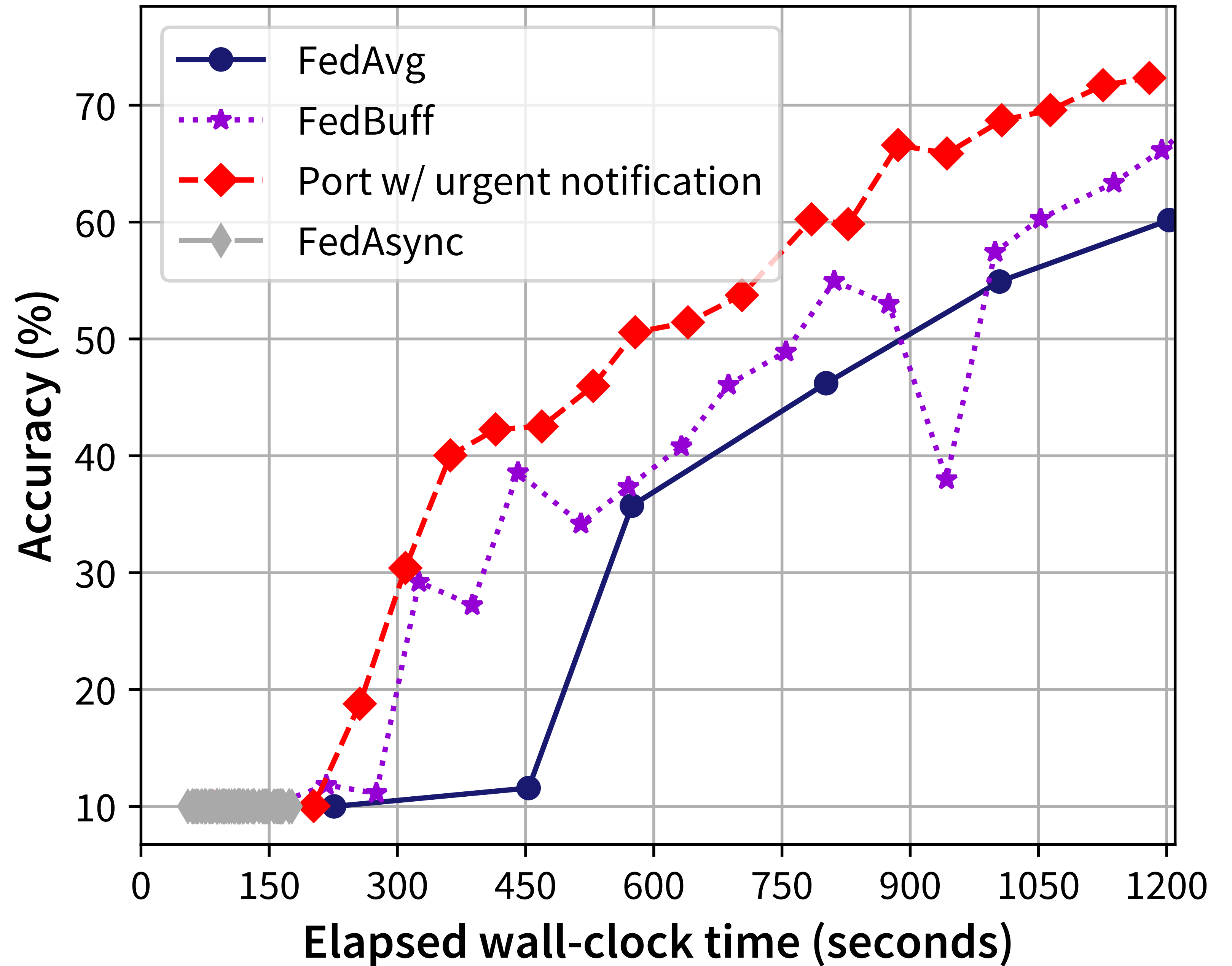


**A new idea: push urgent notifications  
to slow clients**

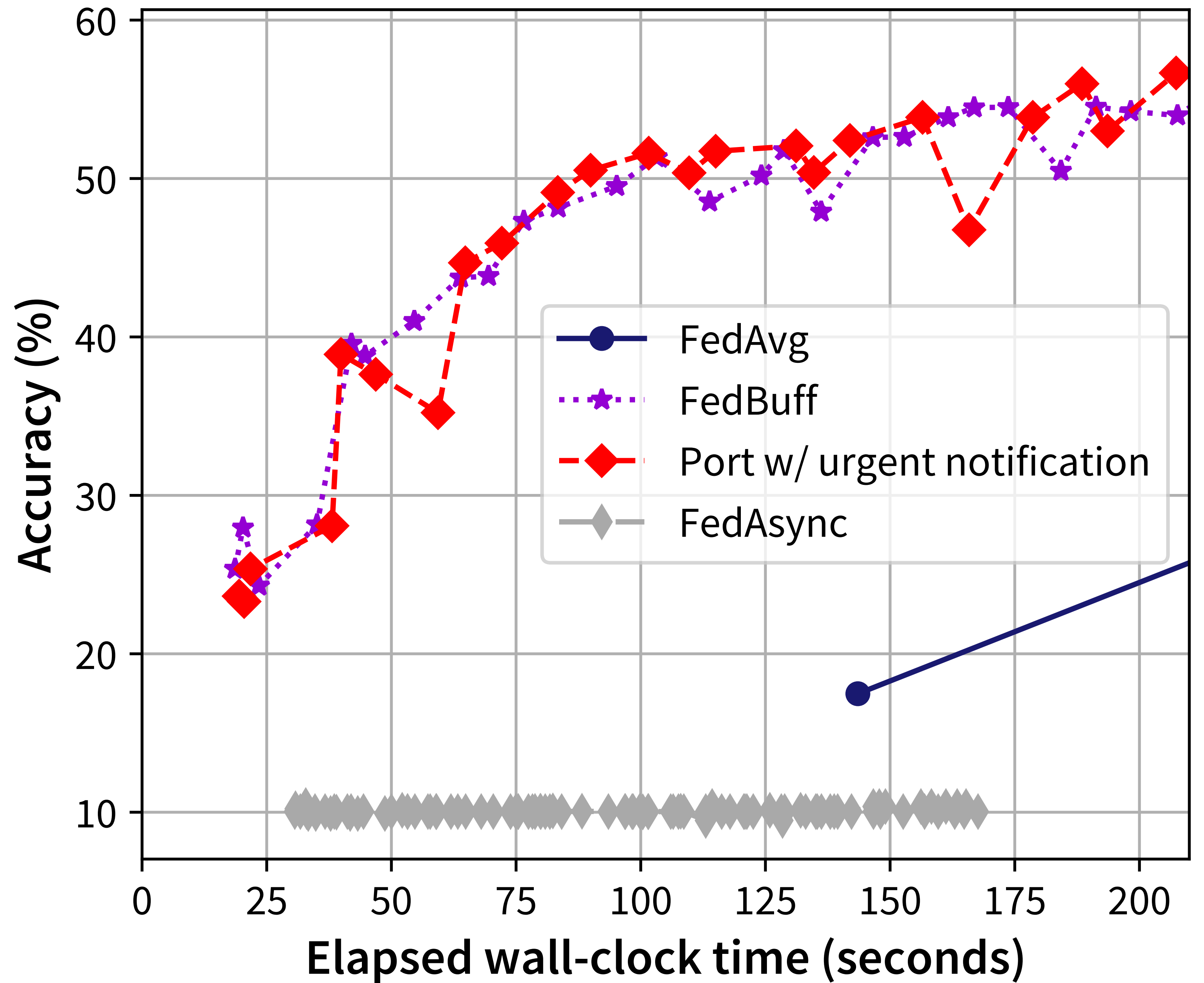


**push-pull** mechanism

# Port (with urgent notifications) vs. its competitors on CIFAR-10



# Port (with urgent notifications) vs. its competitors on CINIC-10



# Port: Our Contributions

**Adaptive** aggregation mechanism based on staleness and cosine similarity

**Urgent notifications** pushed to the slow clients

**Outperformed** the state-of-the-art — FedBuff — in some scenarios

**Thank you**

