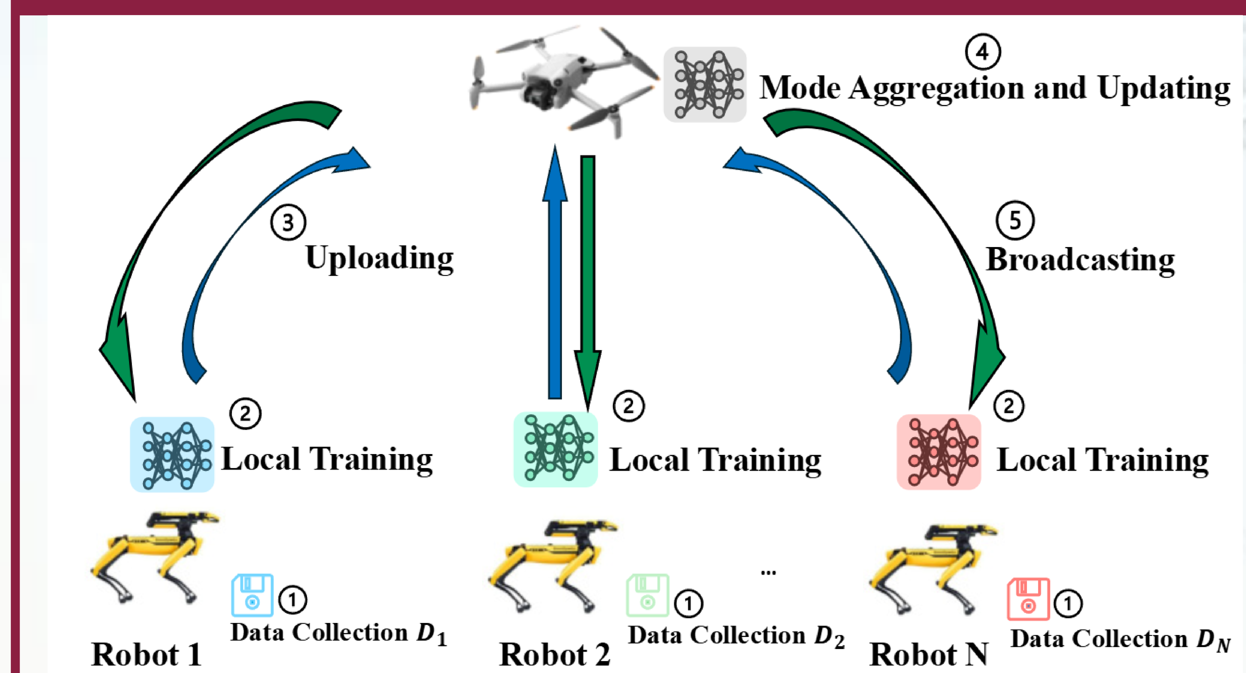
WOS: An Optimized Scheduling Scheme for Wireless Federated Learning

Fangtong Zhou∗, Yi Shi†, Yubo Wu∗, Shiva Acharya∗, Luiz DaSilva†, Sastry Kompella‡, Wenjing Lou∗, Y. Thomas Hou∗
∗ Virginia Tech, † Commonwealth Cyber Initiative, ‡ Nexcepta Inc.

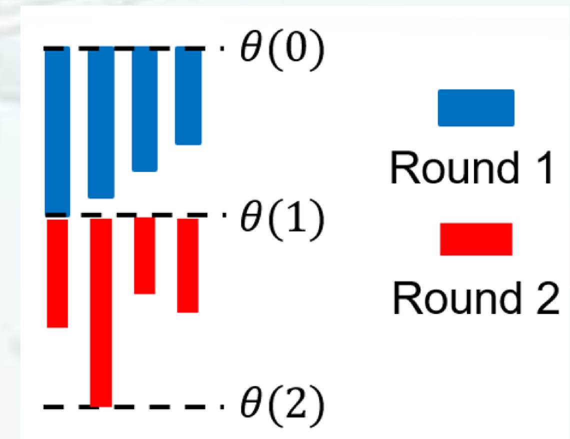
Background



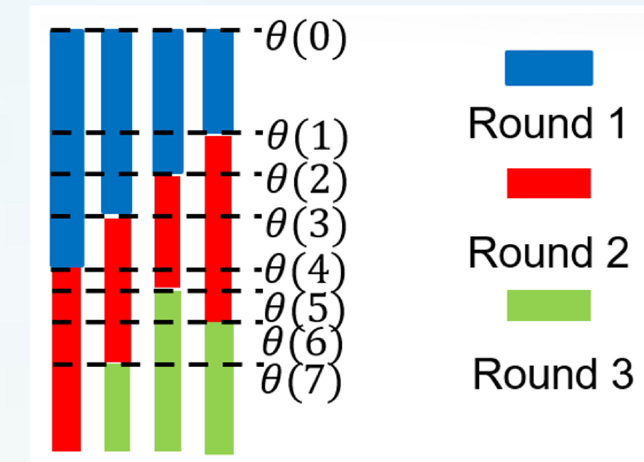
- Use Case
  - Area reconnaissance
  - Search and rescue
- Key Challenges
  - Bandwidth-constrained
  - Dynamic channels
  - Varying local computation time

FL supports many distributed applications such as finance, healthcare, autonomous driving, by enabling collaborative model training without sharing local data.

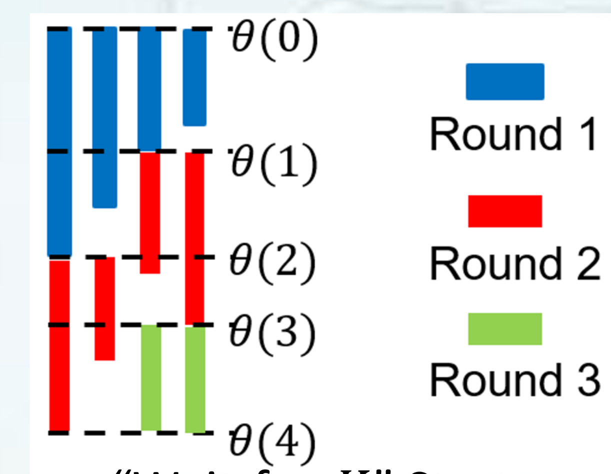
State-of-the-art



- "Wait-for-all" Strategy
  - Server must wait for all clients to finish training
  - Each client waits for the new global model to train its next model
- Pros
  - Incorporate all local models in aggregation
  - Clients always work with freshest global model
- Cons
  - Straggler effect (long wait)
    - Fewer learning rounds under given time
  - Single robot failure (wait forever)



- "Wait-for-one" Strategy
  - At server, upon receiving a local model, immediate update global model
    - Broadcast new global model immediately
  - At local client
    - Completes its local training AND uploads its local model
    - Then work with the latest version of global model for next round training
- Pros
  - No straggler effect
    - Offer most learning
- Cons
  - Wide disparity in model versions being trained by local robots (intractable contribution from local models)



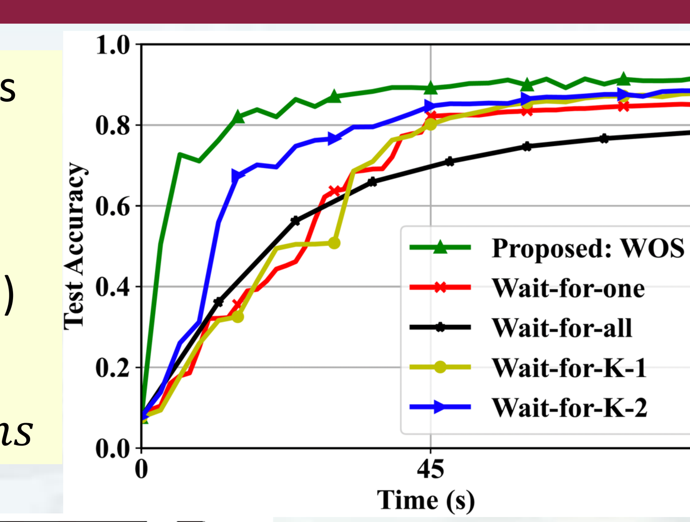
- "Wait-for-K" Strategy
  - A generalization for "Wait-for-all" and "Wait-for-one"
  - UAV waits to receive K local models for aggregation
- "Wait-for-K" Case 1
  - Select K clients, prioritize to those not selected in the previous round
- "Wait-for-K" Case 2
  - Select top-K robots with best channel states

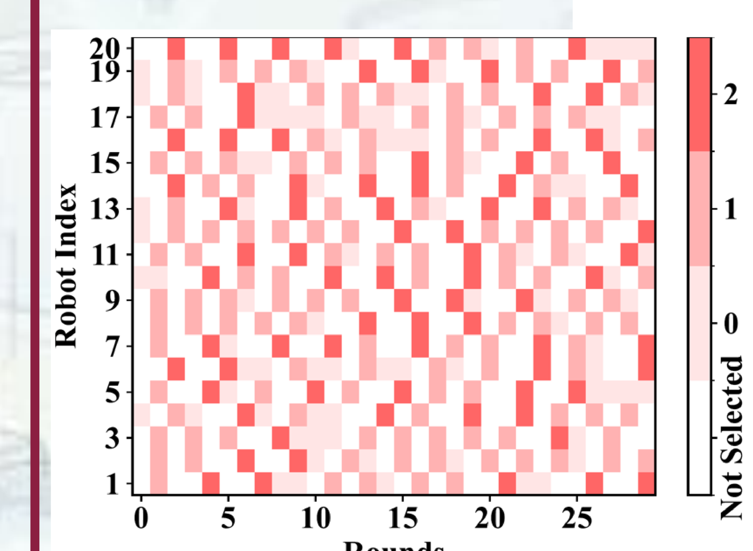
Open Questions

- None of given SOTA addresses the first question: How to set K?
- Observation: A fixed K is not plausible
  - Adaptive K
- Why is this question so challenging in wireless environment?
  - Ending time includes uploading time
    - Scheduling algorithm
    - Wireless channel condition

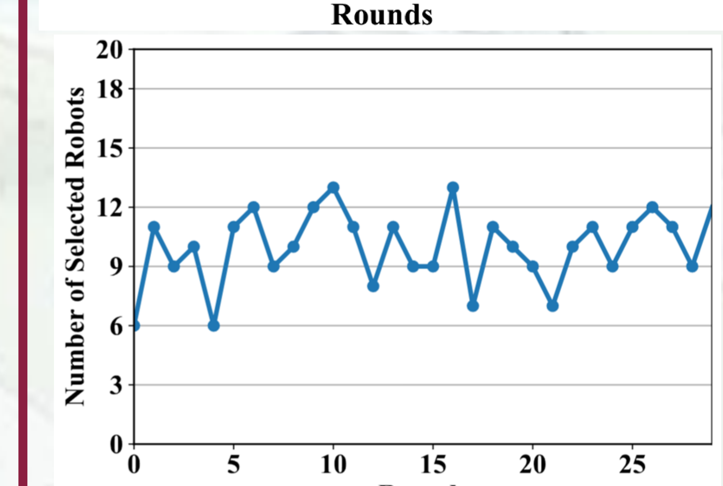
Results: Case Study

- Single UAV, 20 robots
- 40 RBs
- Dataset: MNIST
- Model: DNN ($314KB$)
- Total time: 90 s
- Scheduling slot: 50 ms





No robot is not selected in more than $R = 2$ continuous rounds



The number of selected robots in each round is varying

Proposed Approach: Wait-for-Optimal Set (WOS)

- Objective
  - Determine **the optimal subset** of clients to be selected in each round by selecting all clients completing their training before a **cut-off time**
- To improve global model accuracy, should consider gap in model-version for setting **cut-off time**
  - Make sure all clients complete their local models in no more than R rounds
  - Guarantee the model-version gap between a local and the global models is no greater than R
- Set cut-off time as $\Delta_{cutoff} = \max\left\{\frac{T^{max}_{gap=0}}{R}, \frac{T^{max}_{gap=1}}{R-1}, \frac{T^{max}_{gap=2}}{R-2}, \dots, T^{max}_{gap=k-1}\right\}$
  - $T^{max}_{gap=i}$ is the computation time of the slowest client with i model version gap
- Example when $R = 2$

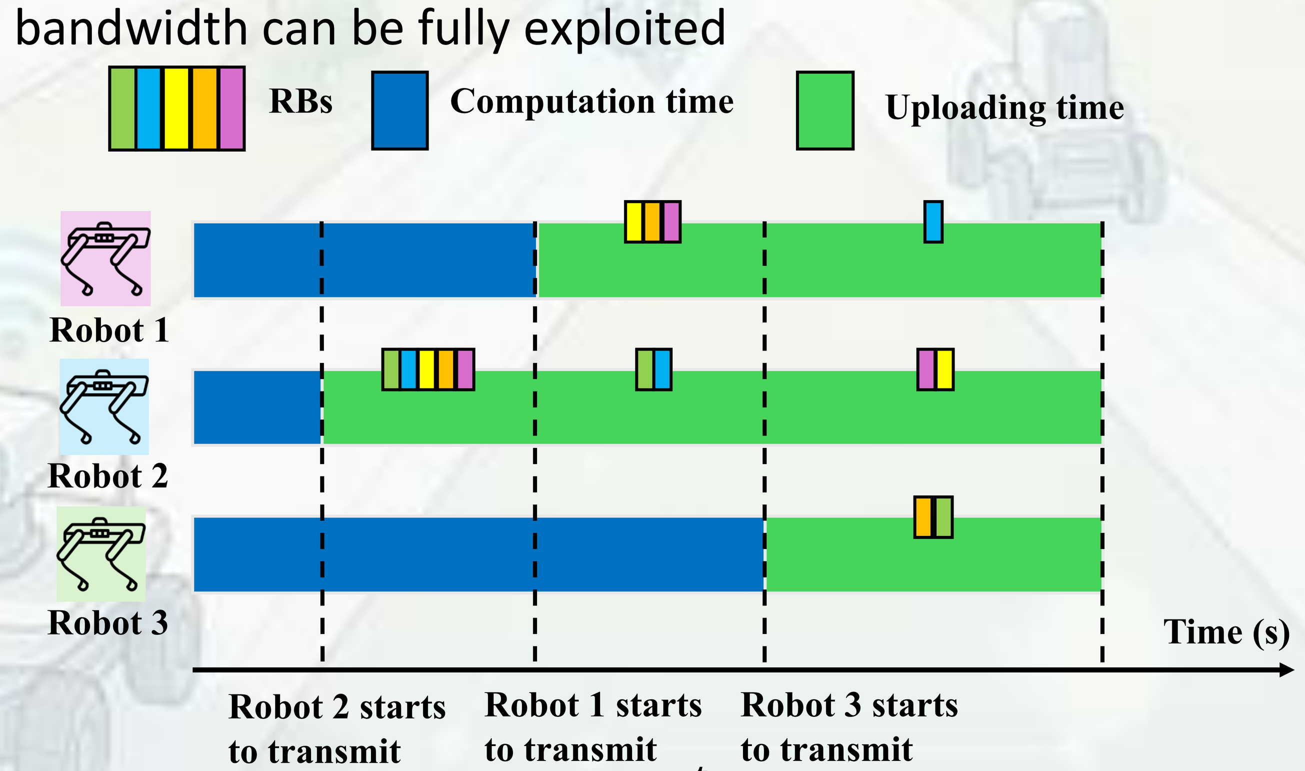
Round 1: $\Delta_{cutoff} = \frac{10}{2} = 5$	Model Ver. Gap	0	0	0	0	0	0
	Est. Comp. Time	(1)	(2)	(3)	6	8	10
Round 2: $\Delta_{cutoff} = \max\left\{\frac{16}{2}, 5\right\} = 8$	Model Ver. Gap	0	0	0	1	1	1
	Est. Comp. Time	(3)	10	16	(1)	(3)	(5)
Round 3: $\Delta_{cutoff} = \max\left\{\frac{12}{2}, 8\right\} = 8$	Model Ver. Gap	0	1	1	0	0	0
	Est. Comp. Time	(7)	(2)	8	(7)	12	10
Round 4: $\Delta_{cutoff} = \max\left\{\frac{10}{2}, 4\right\} = 5$	Model Ver. Gap	0	0	2	0	1	1
	Est. Comp. Time	8	10	(0)	(6)	(4)	(2)

Slowest robot with 0 Model Ver. Gap (shaded: Round 1 – 10; Round 2 – 16; Round 3 – 12; Round 4 – 10)
Slowest robot with 1 Model Ver. Gap (shaded: Round 2 – 5; Round 3 – 8; Round 4 – 4)
Selected Robots (circled values shown in parentheses)

- When $R = 2$, $\Delta_{cutoff} = \max(\frac{T^{max}_{gap=0}}{2}, T^{max}_{gap=1})$
- R is the Model-version gap threshold, which is a tuning parameter

WOS: Wireless Customization

- Clients start to upload immediately once they finish training
- WOS dynamically allocates RB in each TTI to minimize the maximum uploading latency among selected clients so the bandwidth can be fully exploited



- $\min \max_{i \in \mathcal{N}^t(\tau)} T_i(\tau), T_i(\tau) = \frac{Q_i^t(\tau)}{\sum_{j=1}^{M} x_{i,j}^t(\tau)\gamma_{i,j}^t}$
- $Q_i^t(\tau)$ is remaining upload size, $x_{i,j}^t(\tau)$ is RB assignment, $\gamma_{i,j}^t$ is rate on RB j
- Non-linear min-max integer programming problem
- Reformulated to linearize as $\max \min_{i \in \mathcal{N}^t(\tau)} \sum_{j=1}^{M} \frac{\gamma_{i,j}^t(\tau)}{Q_i^t(\tau)} x_{i,j}^t(\tau)$

Publication

F. Zhou, Y. Shi, Y. Wu, S. Acharya, L. DaSilva, S. Kompella, W. Lou, and Y. T. Hou, "WOS: An Optimized Scheduling Scheme for Federated Learning in Dynamic Wireless Networks," in Proc. IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, Oct. 2025

cyberinitiative.org