OST-Assisted Node Selection in Federated Learning: A DTN Perspective

Moncef Zarrouk
ENSIAS, Mohammed V University in Rabat, Morocco

Workshop on Empowering Connectivity: Bridging Space and Earth with DTN
22-26 September 2025

moncefzarrouk7@gmail.com

Hook — Common ground with DTN

- DTN experts: you deal with intermittent links, opportunistic contacts, and probabilistic delivery.
- Key message: Federated Learning (FL) in IoT faces the *same* operational constraints, but nodes also perform local computation.
- Goal: show how OST (used for timely decisions) is natural for client selection in FL.

Mapping: DTN $\leftrightarrow$ Federated Learning

DTN concept	**FL analogue**
Node / relay	IoT client (device)
Bundle (store–carry–forward)	Model update (store–compute–forward)
Contact opportunity	Client–server connection (RSS, availability)
Forward vs hold decision	Select now vs wait (OST)
Delivery probability	Probability of selecting exactly m useful clients

Rural IoT Example

Let's make this concrete. Imagine a rural IoT deployment where sensors are spread across a large area. Some sensors have very accurate local models but weak or intermittent links. Others have stable connections but poor data quality. If the server picks the wrong set, we waste the whole round — just like forwarding a bundle in DTN to a relay that drops it. This is the exact selection dilemma we face in federated learning.

Operational problem: pick the right relays (clients)

- At each training round the server observes n available clients and must pick exactly m.
- Tradeoff: Some clients have high local model quality (valuable "bundle"), others have strong links (high contact probability).
- Wrong selection = wasted round (long delay + poor global update) — analogous to forwarding to a bad relay in DTN.

Our idea: compute a score, then stop (OST)

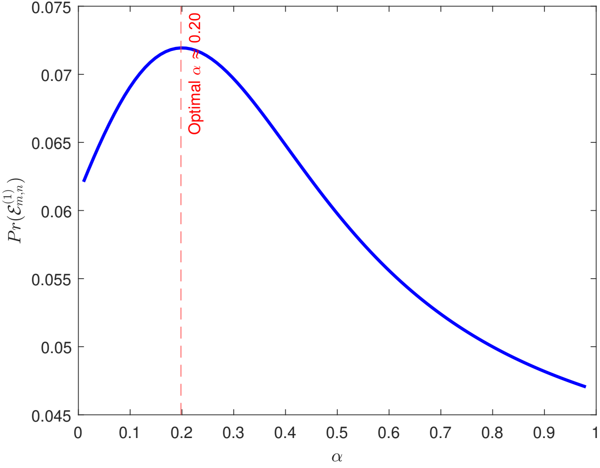
- Compute a single score per client that fuses data quality and link reliability.
- Tune the balance with one knob α (data vs. link).
- Scan clients sequentially and evaluate their scores.
- Apply an optimal-stopping rule: stop when we are confident we have m strong clients.
- Result: lightweight, low-complexity selection robust to intermittent contacts.

Unified score — a single knob α

$$A_i^{(u)} = \alpha A_i + (1-\alpha)\frac{Z_i}{T_\epsilon}$$

- A_i: local model accuracy (value of the bundle)
- Z_i: received signal strength (link reliability)
- $\alpha \in (0,1)$ tunes the balance ($\alpha \approx 1 \rightarrow$ prefer data; $\alpha \approx 0 \rightarrow$ prefer link)

Optimal α (selection probability vs. α)

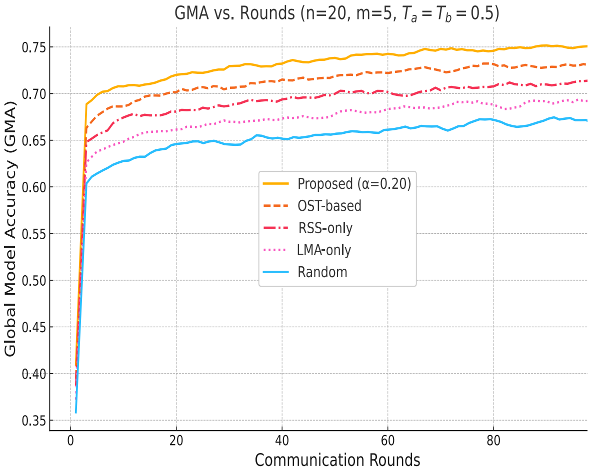


Takeaways

- Peak at $\alpha^* \approx 0.20$
- Maximizes $\mathcal{P}_{m,n}^{(1)}$ (exact m-selection)
- Confirms value of balancing LMA & RSS

Setup	
n, m	20, 5
,	0.5, 0.5
ϵ	4.54×10^{-5}
$P^{(I)}, \gamma$	0.2, 5

Global model accuracy over rounds



Summary

- Proposed ($\alpha^* = 0.2$) reaches $\sim$ 75%
- Outperforms:the classical OST-based, LMA-only, RSS-only, random
- Same conditions, lower runtime complexity

Protocol	
Dataset	MNIST
Rounds	100
LR (local)	10^{-5}
Clients / round	$m = 5$

Takeaways

- FL in IoT behaves like DTN with computation: intermittent contacts, opportunistic delivery, and probabilistic decisions.
- A single tunable knob α plus OST gives a lightweight, robust client selection protocol.
- Practical for DTN-like deployments (vehicular, rural, satellite) where contacts are scarce and costly.

- Thank you .