Distributed Implementation of a Learning-Enabled Demand-Response Mechanism over Cefore
（学習エージェントを用いた需要応答メカニズムのCefore上での分散実装）

Xun Shao[1] · **Ryoichi Inoue**[1] · **Shinken Takekawa**[1] · **Go Hasegawa**[2]

[1] Toyohashi University of Technology [2] Tohoku University

https://doi.org/10.48550/arXiv.2608.17469

SS-Lab-TUT

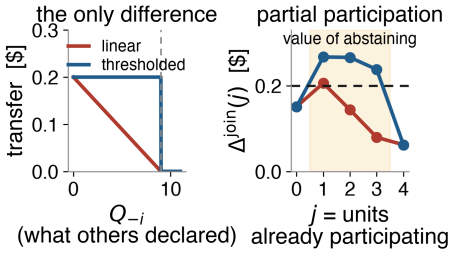
Xun Shao — Senior Member, IEEE · Editor-in-Chief, IJPCC
Symposium Co-Chair, IoT & Sensor Networks Symposium, IEEE ICC 2027

Static equivalence does not imply learning equivalence — and distribution must preserve who declared what, in which round.

1 Why the static checks leave the outcome undetermined

Aggregators recruit distributed energy resources with settlement rules and participation payments. Such designs are validated at fixed points: zero participation must cease to be an equilibrium, truthful reporting must remain a best reply.

We hold everything else fixed and vary only how the transfer $R_i = 1\{\text{participate}\} \cdot s_0 f(Q_{-i})$ decays with what the others declared. Linear and thresholded agree on all five static criteria — allocation and payments at the intended profile, every contract-selection margin, $s_0^{elim} = 0.003104$, $s_0^{entry} = 0.199928$ — and by Theorem 1 the transfer vanishes there, so no s_0 disturbs the calibration.



Linear — the transfer falls off as the others declare more.
Thresholded — held at full value until the capability target is met, then zero.

STATICALLY EQUIVALENT ≠ LEARNING EQUIVALENT	
96 / 96	**0 / 96**
thresholded reach full participation	linear — none do

Collapse initialization, $s_0 = 0.199928$, $\beta = 4$, 8 000 rounds; 95 % Wilson intervals [0.962, 1.000] vs [0.000, 0.038]. From a random start every condition reaches 1.00.

$\min_j \Delta^{join}(j)$ **OVER THE PARTIAL-PARTICIPATION STATES [$ per event-day]**

linear		incumbent estimate $\hat{u}^{(0)}$		thresholded
0.0795	<	**0.20**	<	**0.2378**

The static criteria are evaluated only at the two ends, where the structures agree.

THE THRESHOLD ORDERING REVERSES
Thresholded carries the population at $s_0 = 0.12$; linear needs 0.35, well past the entry threshold.

MONOTONE IN THE INTERMEDIATE PAYOFF
Across nine members of the decay family, convergence is monotone in the minimum partial-participation payoff.

IT IS A PROPERTY OF BANDIT FEEDBACK
Give the counterfactual payoffs and the separation closes; at 96 000 rounds every condition converges.

A unit's settlement is a function of what the others declared, and is absorbed into state that shapes its later declarations.
One inconsistent declaration therefore moves several units' settlements in that round, and their behaviour in every round after it.

2 From equations to a running CCN system

A simulator may assume the correct unit–round–declaration association; a distributed implementation must preserve it.

each unit: settlement received → own realized payoff → update its private estimate $\hat{u}_i$, n_i → next declaration

AGGREGATOR
collects declarations · assembles the joint profile · computes every settlement

round state ↓ declaration (abstain / conservative / aggressive) ↑ ↓ settlement $w_i = U_i + R_i$

CEFORE (NICT)
aggregator and units as independent OS processes in separate namespaces · v0.12.0 under CeforeEmu (Mininet) · six forwarders · every crossing a real Interest/Data exchange

p0 | **p1** | **p2** | **p3** | **p4**

FIVE INDEPENDENT LEARNING UNITS — each holds its own estimate $\hat{u}_i$ and visit count n_i, which never leave it, and its own cefnetd

Cefore names round state `ccnx:/<app>/agg/round/{r}/state` declaration `ccnx:/<app>/p{i}/decision/{r}` settlement `ccnx:/<app>/agg/settle/{r}/p{i}`
one cefnetd per namespace · publisher identity is the routing prefix · static FIB, N + 1 = 6 prefixes · no routing protocol · no non-CCN fallback

Exactly three object classes cross a boundary: the round state · each declaration · each unit's own settlement. The payoff table, calibration and assembled profile stay at the aggregator.

3 Did the mechanism survive distribution?

SEMANTIC FIDELITY (Def. 2): every round, every unit — declaration, joint profile, settlement and learner state equal to the centralized reference, exactly.

POSITIVE VALIDATION

DISTRIBUTED = CENTRALIZED, ALL 8 000 ROUNDS
N = 5 · seed 12345 · profile-mean estimator · no tolerance anywhere

L1	declaration (vs. centralized · vs. aggregator)	0 / 40 000 · 0 / 40 000
L2	assembled joint profile	0 / 8 000
L3	settlement vector (max abs. diff 0.0)	0 / 40 000
L4	learner estimate and count (max abs. diff 0.0 and 0)	0 / 240 000 · 0 / 240 000
L5	coarse level m(r) · first passage r = 7538 in both	0 / 8 000

STILL EXACT UNDER IMPAIRMENT
200 ms delay, 40 ms jitter, 15 % loss, 0.5 Mbit/s: round time up to 3.45×, 371 timeouts — all 750 fetches succeeded, every quantity still equal.

Canonical run: 40 000 fetches · 0 retries · 0 aborts · 6 of 6 forwarders alive · 15.76 h at 7.09 s per round.

NEGATIVE CONTROL

EVERY MESSAGE DELIVERED
450 retrievals · 330 publications · 0 retries · 0 timeouts · 0 aborts

ONE PREREGISTERED ATTRIBUTION SWAP
at round 10, two already-fetched declarations swap attribution

MECHANISM STATE DIVERGED
joint profile 1 / 30 · settlement 4 / 150 · unit-vs-aggregator declaration 2 / 150
learner state 80 / 150 — and it persisted to the end of the horizon

WHAT DID NOT DIVERGE, AND WHY IT MATTERS
Unit-side declarations 0 / 150: no unit's own behaviour changed — an attribution error, not a behavioural one. Counts n 0 / 150 and level m 0 / 30 also agreed.

DELIVERY SUCCESS ≠ MECHANISM CORRECTNESS
Not a Cefore failure — the fault was injected after successful retrieval.