SLO-Aware Task Offloading within Collaborative Vehicle Platoons

TU Wien: Boris Sedlak, Andrea Morichetta, Schahram Dustdar
Tsinghua University: Yuhao Wang, Yang Fei, Liang Wang, Xiaobo Qu

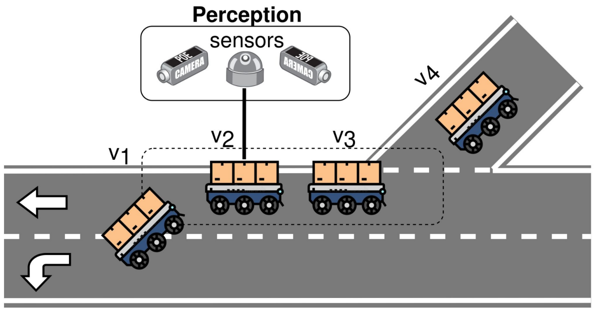
Problem Statement

Autonomous Vehicles (AVs) use services for perception and path planning, which pose **runtime requirements**

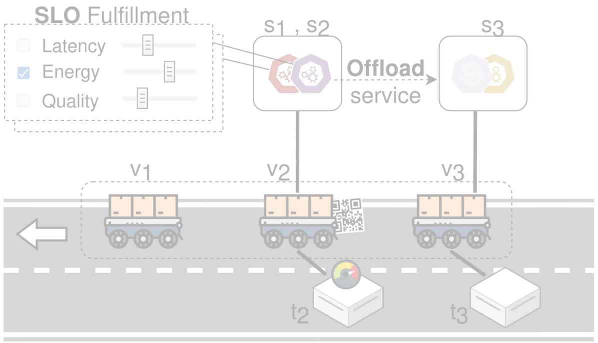
AV-enabling services have high computational demand, e.g., video or Lidar processing, but, AVs have **limited resources** available for scalability during runtime

Service offloading as an alternative; vehicle platoons can allocate services at **suitable** host and share results

When SLOs are violated, e.g., latency or quality, a service can be offloaded to a less utilized platoon member



Perception services executed in collaborative platoons



Offloading services to less utilized platoon members

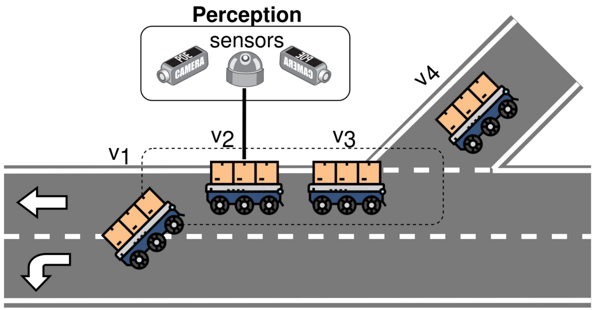
Problem Statement

Autonomous Vehicles (AVs) use services for perception and path planning, which pose **runtime requirements**

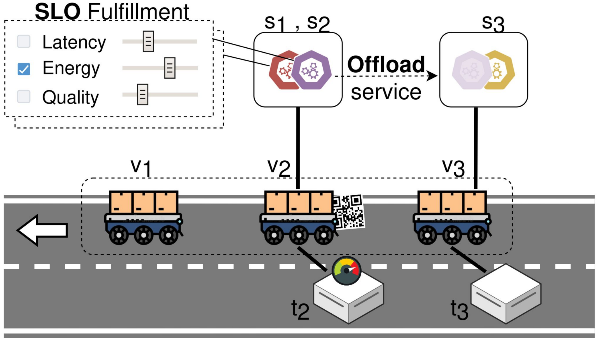
AV-enabling services have high computational demand, e.g., video or Lidar processing, but, AVs have **limited resources** available for scalability during runtime

Service offloading as an alternative; vehicle platoons can allocate services at **suitable** host and share results

When SLOs are violated, e.g., latency or quality, a service can be offloaded to a less utilized platoon member



Perception services executed in collaborative platoons



Offloading services to less utilized platoon members

Existing Shortcomings

Dynamic platoon disruptions

Vehicles can join or leave a platoon at any point, making it hard to train and apply an inference model in once consecutive session

⟶ Transfer decision models and tasks as devices join

Global impact of service offloading

Offloading a service to another device has implications on the resource availability of the remaining services on both sides

⟶ Estimate how offloading affects both sides

Existing Shortcomings

Dynamic platoon disruptions

Vehicles can join or leave a platoon at any point, making it hard to train and apply an inference model in once consecutive session

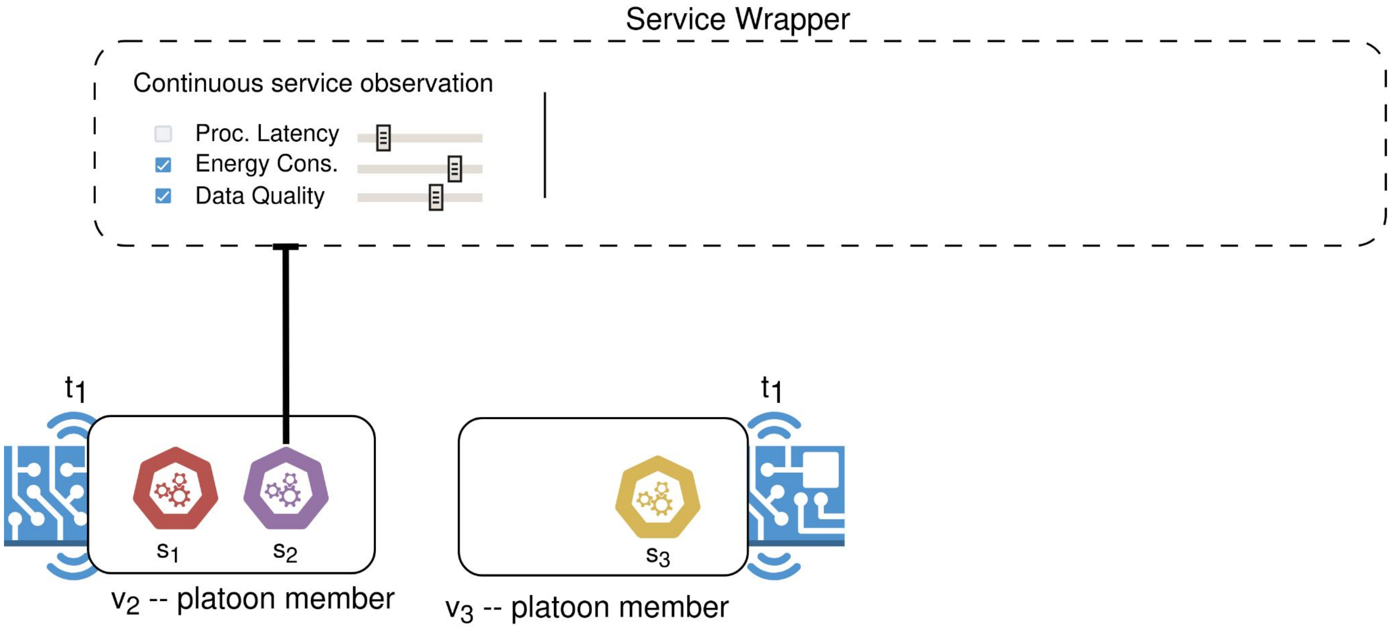
⟶ Transfer decision models and tasks as devices join

Global impact of service offloading

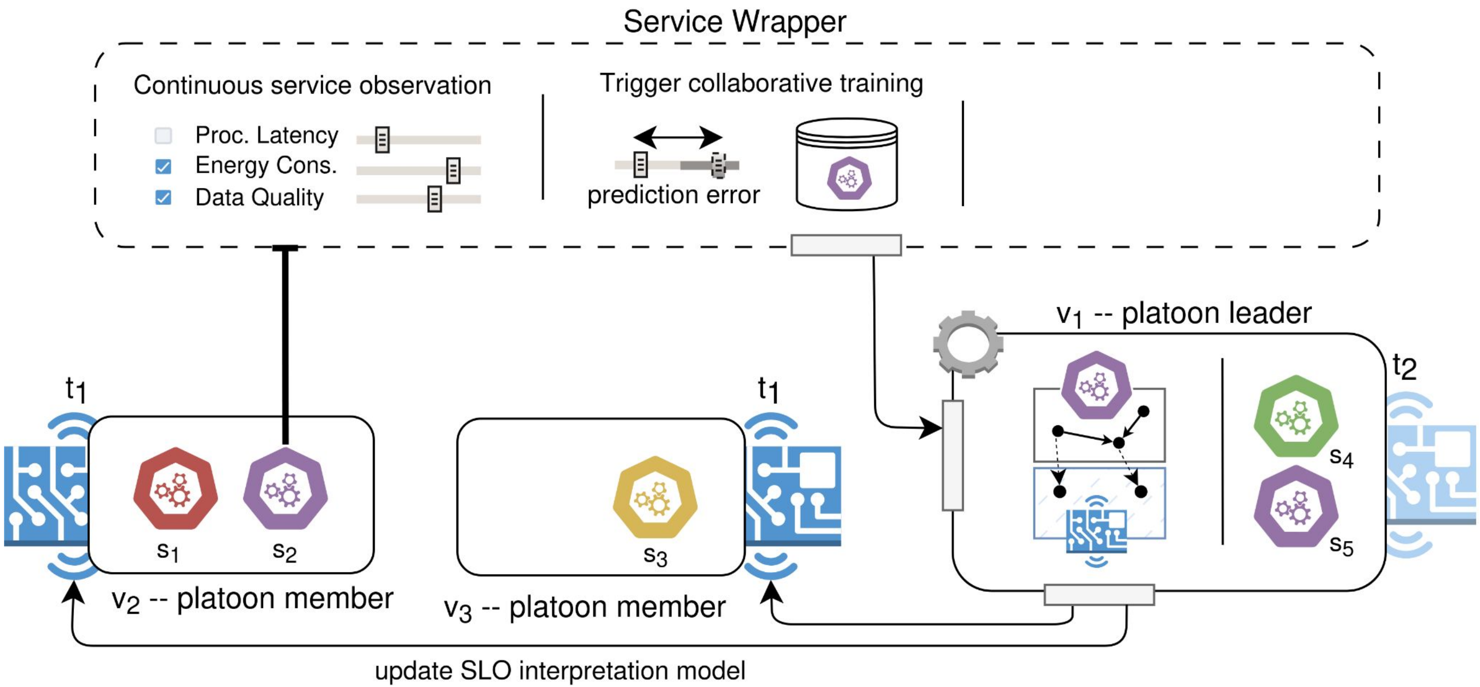
Offloading a service to another device has implications on the resource availability of the remaining services on both sides

⟶ Estimate how offloading affects both sides

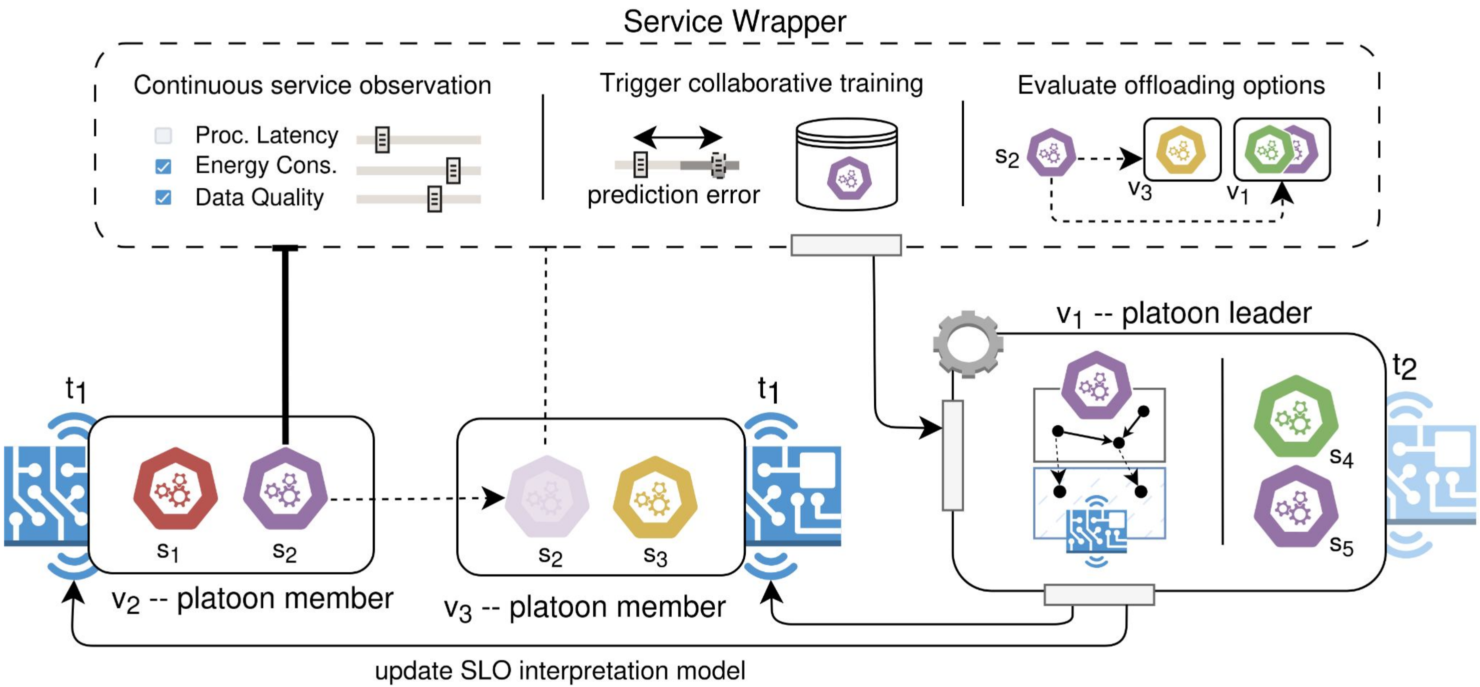
Methodology – Overview



Methodology – Overview

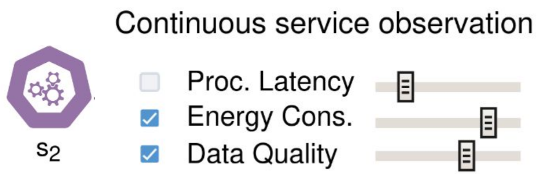


Methodology – Overview

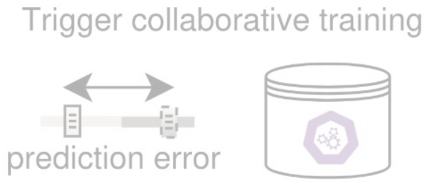


Methodology – Phases

Service Observation
(**1**) Track metrics of processing services in a local buffer;
(**2**) use metrics to evaluate local SLO-F(ulfillment);
(**3**) compare actual SLO-F with historical prediction using a Bayesian Network (BN) ⟶ SLO-I(nterpretation) model;
(**4**) get retrain factor from prediction error and buffer size



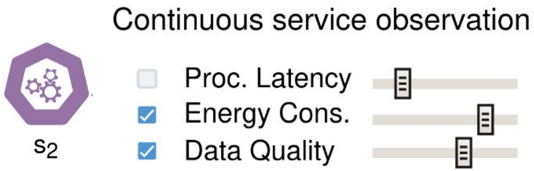
Collaborative Training
(**1**) When retraining threshold is met, send buffer to a mutable platoon leader; (**2**) leader computes an updated SLO-I model for the source device type; (**3**) leader distributes update model to all vehicles with device matching type; (**4**) substitute BN during service runtime



Methodology – Phases

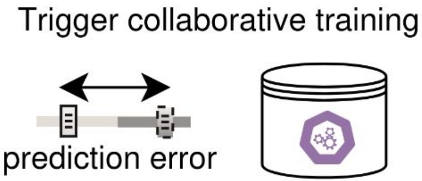
Service Observation

(**1**) Track metrics of processing services in a local buffer;
(**2**) use metrics to evaluate local SLO-F(ulfillment);
(**3**) compare actual SLO-F with historical prediction using a Bayesian Network (BN) ⟶ SLO-I(nterpretation) model;
(**4**) get retrain factor from prediction error and buffer size



Collaborative Training

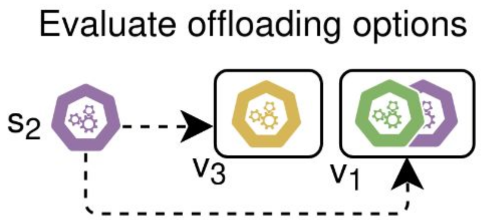
(**1**) When retraining threshold is met, send buffer to a mutable platoon leader; (**2**) leader computes an updated SLO-I model for the source device type; (**3**) leader distributes update model to all vehicles with device matching type; (**4**) substitute BN during service runtime



Methodology – Phases (cont.)

Service Offloading

(**1**) Calculate evidence to load off according to prediction error and absolute SLO violation; (**2**) if offloading threshold is met, estimate for each platoon member how offloading a service there would impact SLO-F at source and target device; (**3**) choose best option and orchestrate service relocation



- ❑ Estimated SLO-F dependent on resource availability
- ❑ Agency to offload lies with the processing service itself
- ❑ Frequency of wrapper execution decides reactivity

Experimental Environment

→ Build a physical vehicle platoon from Edge devices; use multiple instances of NVidia Jetson equipped with GPUs

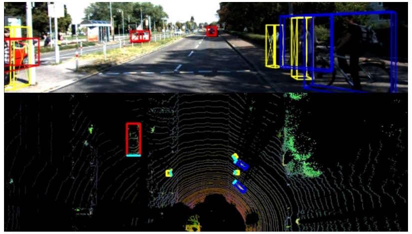
Full Device Name	ID	Price[8]	CPU	RAM	GPU	CUDA
Jetson Orin NX (3)	*NX*	450 €	ARM Cortex 8C	8 GB	Volta 1k	11.4
Jetson Orin AGX	*AGX*	800 €	ARM Cortex 12C	64 GB	Volta 2k	12.2

→ Provide three perception services for AVs; process streaming data (video / point cloud) according to SLOs

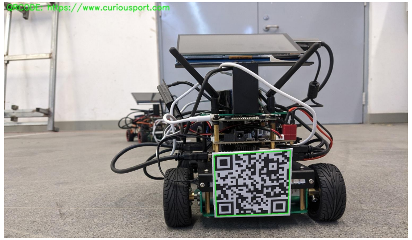
ID	Service Description	CUDA	Parameters	SLOs
CV	Object Detection with Yolov8 [26]	Yes	*pixel*, *fps*	**time**, **energy**, **rate**
LI	Lidar Point Cloud Processing [6]	Yes	*mode*, *fps*	**time**, **energy**
QR	Detect QR Code w/ OpenCV [21]	No	*pixel*, *fps*	**time**, **energy**

CV

LI



QR



[1] https://github.com/borissedlak/intelligentVehicle/

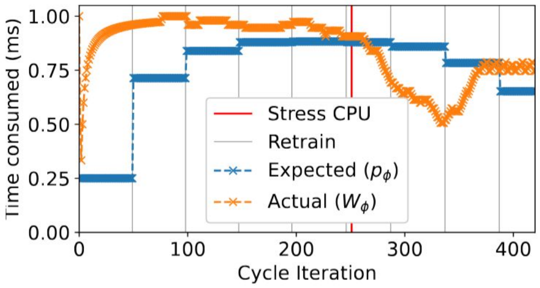
Evaluation: SLO-dependent Retraining

→ Evaluate if SLO-dependent retraining (i.e., evidence to retrain > t) improves SLO-F after SLO violations

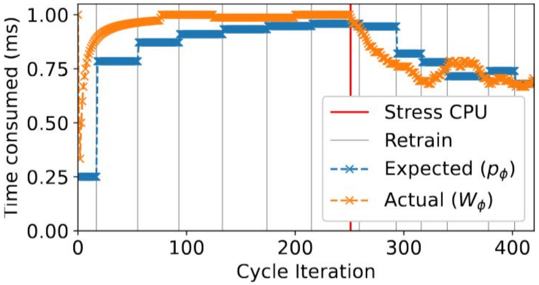
Setup: Execute *LI* service on Jetson *Orin*; start without prior understanding, i.e., blank SLO-I model; let the agent retrain its SLO-I model until it has a good prediction accuracy; after 125s, disturb its predictions by introducing 40% CPU stress

Result: Using SLO-dependent retraining, the agent reported improved prediction accuracy both before and after stress

Implication: Useful for accelerating model updates during phases with dynamic behavior; short cycles improve accuracy



(a) Without SLO-dependent retraining

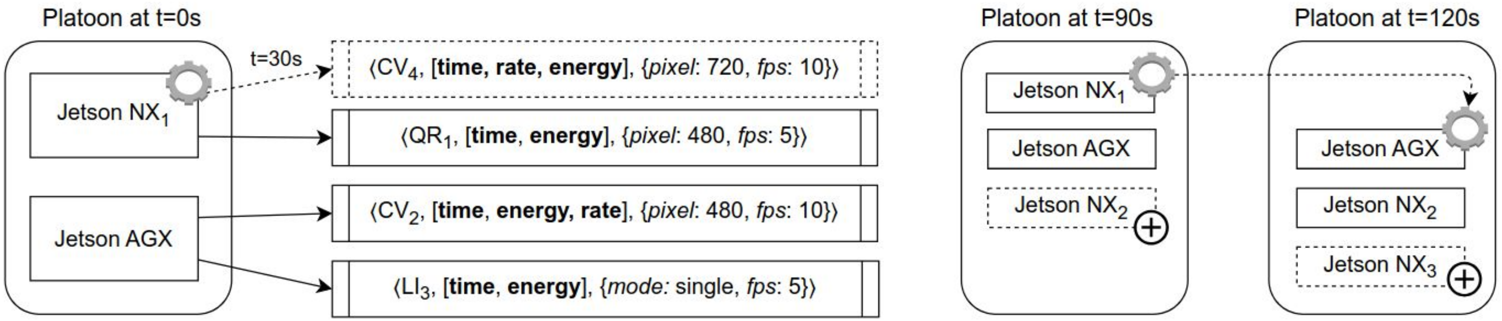


(b) With SLO-dependent retraining

Evaluation: Global SLO Fulfillment

→ Evaluate the global SLO-F of the methodology

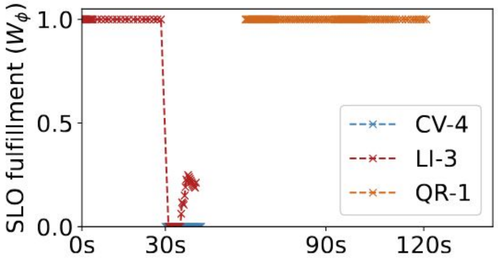
- ❑ **t=0**; start processing at two platoon members
- ❑ **t=30**; start a new service (i.e., *CV4*) at device NX1
- ❑ **t=90**; Add a new device (i.e., *NX2*) to the vehicle platoon
- ❑ **t=120**; Add one vehicle, and remove the existing leader



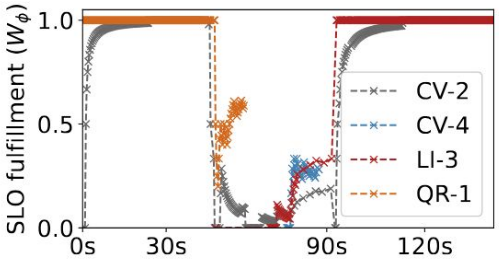
Evaluation: Global SLO Fulfillment (cont.)

Result: Initially, *NX1* and *AGX* had an SLO-F of 100%; at **30s**, *NX1* starts a new service, which crushes the SLOs for all its services; the devices shift services and improve SLO-F slightly; at **90s**, adding a new device to the platoon recovers SLO-F; at **120s** changing platoon leader had no negative impact

Implication: Services can be offloaded dynamically within the AV platoon to find the optimal assignment given available resources; also, we could show that the platoon leader is mutable and cannot become a single point of failure



(a) Orin NX_1



(b) Orin AGX

Conclusion

Perception services have high computational requirements, that **constrain** the deployment, i.e., where and how to execute

Forming platoons of AVs allows to **share resources** between vehicles by offloading services according to members’ capacity

---

Services estimate the expected SLO-F according to **historical data**, retrain their SLO-I model, and decide to offload to another platoon member if this promises to improve global SLO-F

Methodology evaluated in a physical testbed; we could show that smaller platoons could shuffle services according to available resources so that a list of SLOs was fulfilled

Let’s **discuss**!

Please come forward with any **question** you have

@

boris.sedlak@dsg.tuwien.ac.at

Evaluation: Platoon Scalability

⟶ Evaluate the scalability of increasing platoons

Setup: Start a platoon with a single vehicle that executes a CV service; every 25s (= 50 iterations) add a vehicle to the platoon and measure wrapper execution time

Result: Linear impact of platoon size on the wrapper execution time, exception for |P| = 0, which is obsolete

Implication: Given an evaluation interval of 500ms, the services might struggle to evaluate larger platoos with |P| > 3; can structure platoon into smaller subgroups or adjust evaluation interval