Performance Aware Cloud-level Load Balancer for Connected and Autonomous Vehicles

Praveen Tammana, CSE Dept. IIT-Hyderabad

Proposal Summary: Vision-based autonomous ground vehicles with few radars are gaining more attention because of their low cost. However, camera-based sensing devices rely on a wide range of vision-based compute-heavy object detection algorithms like pedestrian detection, and obstacle detection etc., Due to constraints on available onboard computing and power constraints running accurate algorithms is challenging. Moreover, often the vehicles run out of date AI/ML models due to less frequent or zero software updates. To overcome these obstacles, in this work, we unlock the true potential of edge cloud and propose a performance-aware programmable onboard load balancer (LB) that intelligently routes video inference requests to either less accurate and lightweight onboard models or more accurate and heavyweight models in edge cloud based on the application requirements such as latency, accuracy, and available bandwidth between onboard and edge cloud. We characterize the performance of the proposed onboard LB in terms of request completion time, power consumption, and accuracy.

Deliverables:

1. A prototype of an "intelligent onboard load balancer" that splits video frames from an autonomous vehicle (e.g., warehouse rover) cameras between onboard and edge cloud.
2. Test the prototype using TiHAN testbed's outdoor WiFi and Edge cloud lab.
3. Evaluate the performance regarding scalability, request completion time, resource overheads, and accuracy.

A publication covering the problem statement, different design aspects of the system, and evaluation results.

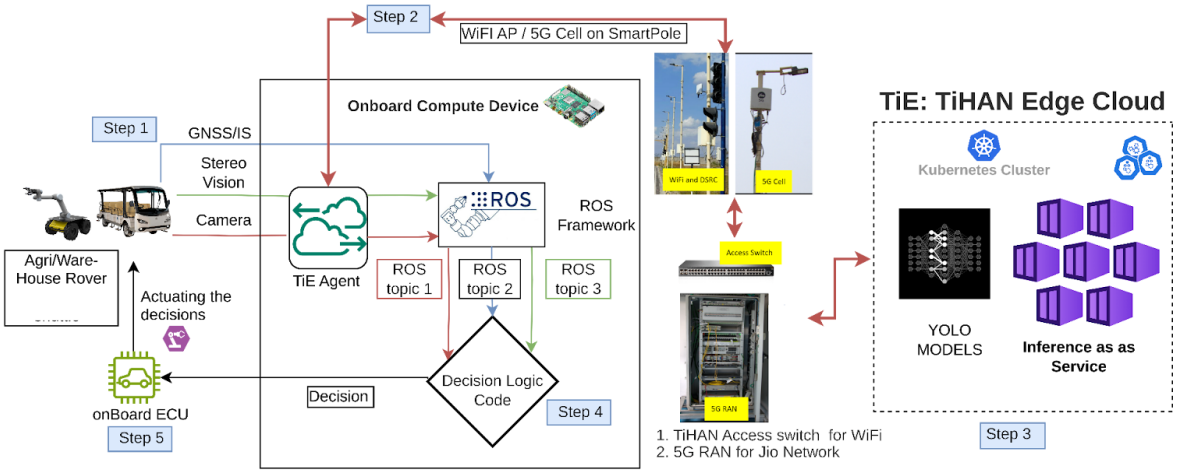
1. Problem statement

An Autonomous Vehicle (AV) would require a satellite positioning system and actively learn about its surroundings to avoid obstacles and navigate safely such as the detection of vehicles, pedestrians, speed bumps, traffic lights, traffic signs, etc. Similarly, an autonomous agricultural rover engaged in various tasks like harvesting, sowing, pruning, weeding, spraying, and monitoring, requires the detection of weeds, pests, objects, etc. Likewise, autonomous vehicles in warehouses involved in activities such as picking, packing, and transporting goods would require to detect items, empty slots to place items, and objects, etc.

Achieving levels 3 and 4 of automation requires deploying top-of-the-line hardware such as GPUs and sensors in the vehicle, which can be quite expensive. For example, LIDAR, a remote sensing device that uses light pulses to generate precise 3D maps of a vehicle's surroundings, is prohibitively costly at USD 75,000 and is therefore not widely available commercially. The need for additional high-data sensing sensors and computing also means increased power consumption due to the higher demands on cooling and storage. This also leads to a rise in power constraints, as well as increased battery costs.

To address this, vision-based autonomous ground vehicles with few radars are gaining more attention because of their low cost. However, camera-based sensing devices rely on a wide range of vision-based compute-heavy object detection algorithms like pedestrian detection, obstacle detection etc., Some of the vehicles have power constraints due to limited battery power. Often the vehicles run out of date AI/ML models due to less frequent or zero software updates.

2. Our approach: Offload compute-heavy vision-based object detection jobs to the nearest cloud

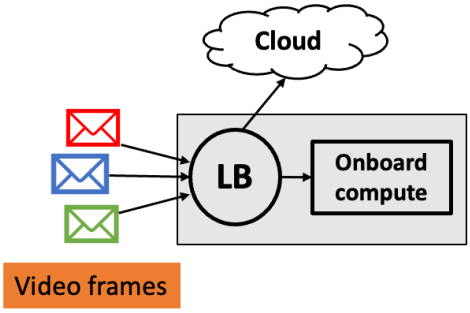


Offloading these compute-heavy object detection algorithms to a cloud is one option. However, public clouds are typically located in far-off places increasing the network latency from the source. This latency is directly related to the distance of the cloud from the source application and will only increase in case of a congested network.

As an alternative, the edge cloud keeps the processing closer to the data source. Our goal is to leverage compute in the edge cloud for vision-based autonomous ground vehicles, such as rovers and campus shuttles, in controlled environments such as warehouses, agricultural farms, small to large-scale industries, institutions, etc. This approach mitigates the risk associated with dynamic uncontrolled environments and also reduces the need for tedious and repetitive human labor. Leveraging edge cloud, we also envision reducing the reliance on expensive onboard computing, GPUs, and batteries and enabling low-cost alternatives. On-board software can be programmed to switch between less accurate algorithms deployed on-board and more accurate algorithms deployed at the edge cloud. Especially when the network conditions are good, we can invoke more accurate and compute-heavy detection algorithms running at the edge cloud via an API call. This choice is particularly beneficial as we aim to deploy simple, low-cost, and power-constrained compute boards on autonomous vehicles, such as campus shuttles and rovers.

3. Solution: Performance-aware load balancer

In this proposal, we work on building a novel onboard load balancer (LB) for AVs connected to an edge cloud. The LB routes video inference requests between onboard and cloud. Onboard has models with low accuracy and resource efficiency. Whereas Edge Cloud runs models with high accuracy. If the vision-based application is delay-sensitive, video frames are forwarded to both the onboard and edge cloud and the cloud's response is picked only if the response arrives on time (e.g., < ~100 ms). If the application is delay tolerant, then forward frames to one of the two locations based on the network conditions.



The proposal's goal is to explore and build an onboard LB that considers the application's delay tolerance, network delays, edge cloud availability, application accuracy level (high to low), and onboard load. More specifically, we design and develop an onboard intelligent and programmable load balancer for a vision-based object detection application to be deployed on autonomous vehicles. The features of proposed LB are:

1. Aware of application requirements (e.g., delay tolerance, accuracy)
2. Reroute requests based on network performance
3. Adapt fast to changes (100's of microsecs)

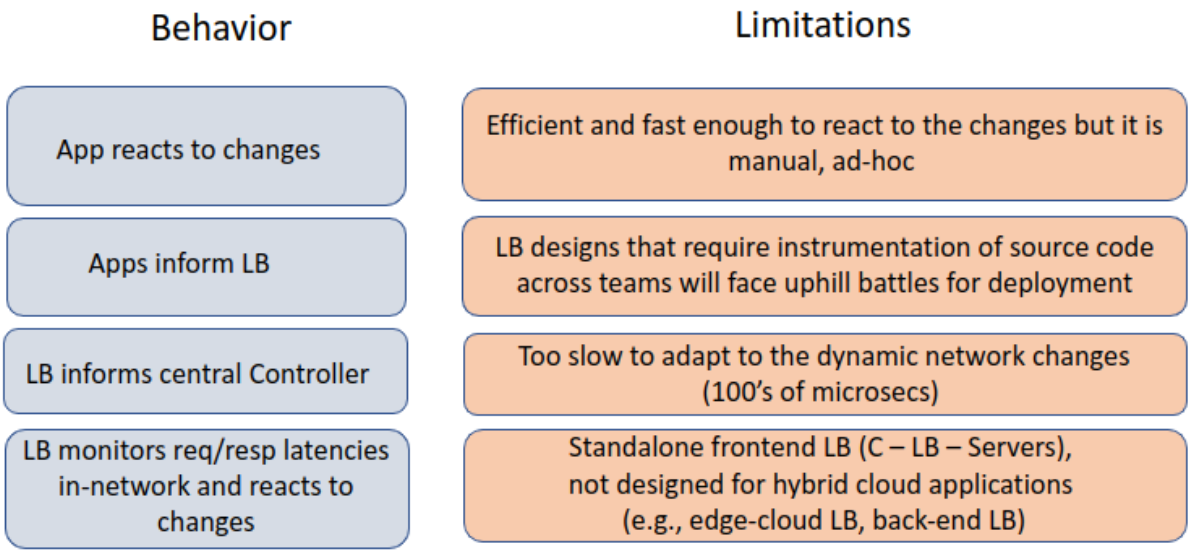
4.Prototype using TiHAN Edge Cloud

Implementation: Implement a working prototype of the proposed onboard load balancer on an autonomous vehicle (e.g., warehouse rover) and object detection application running on cloud platforms such as envoy proxy, Istio, Linux networking, and Kubernetes.

Testing and evaluation: Deploy the prototype on an autonomous vehicle (e.g., warehouse rover) at the TiHAN testbed and test by sending vision-based workloads. Benchmark the performance in terms of request completion time and requests per second with and without the proposed LB. This helps to compare the performance and demonstrate the benefits of the proposed LB.

5. Related work

Industry: The current load balancing systems in practice such as envoy proxy have a few strategies such as RR, random. Ring hash, weighted least request, etc. But still all these strategies cannot efficiently split the traffic based on the app requirements and there is no dynamic adaptation to choose the best next hop which could considerably affect the performance of the application.



Literature: In the literature, there has been a considerable amount of work done in the domain of load balancing systems. Applications themselves do the work of splitting the traffic by reacting to the changes [1] This solution is fast and efficient to react to the changes in the network and system. But, these solutions are manual and ad-hoc and if any changes need to be made, the whole code of the application has to be changed hence not developer friendly. Applications informing a load balancer and the load balancer deciding splitting of traffic [2]. This solution is also manual and ad-hoc and the application developers have to write the code for both the load balancers and the application for this kind of configuration which is very difficult. The load balancer informs a Central Controller and the controller decides to route requests [3]. This solution is very slow to adapt to the dynamic network changes. Finally, Load balancers themselves react to the changes [2]. This solution is fast and efficient but it is not designed for hybrid cloud applications to choose the best cloud among the currently available clouds and is not designed to choose the best pod in the backend within the edge.

References:

[1] Francis Y. Yan, Hudson Ayers, Chenzhi Zhu, Sadjad Fouladi, James Hong, Keyi Zhang, et al., "Learning in situ: a randomized experiment in video streaming", Proceedings of the 18th USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2020.

[2] Bhavana Vannarth Shobhana, Srinivas Narayana, and Badri Nath. 2022. Load balancers need in-band feedback control. In Proceedings of the 21st ACM Workshop on Hot Topics in Networks (HotNets '22). Association for Computing Machinery, New York, NY, USA, 76–84. https://doi.org/10.1145/3563766.3564094

[3] 2013. HAProxy load balancer feedback agent check. [Online,Retrieved Oct 14, 2022.] https://www.loadbalancer.org/blog/open-source-windows-service-for reportingserver-load-back-to-haproxy-load-balancer-feedback-agent/. (2013)