Platoon Dynamics
A Cloud Platform for the Coordination of Vehicular Platoons

Antonio Menarde, Patrick Taggart, Stephanie Tang, Tristrum Tuttle, Monica Vyavahare

University of Pennsylvania
School of Engineering and Applied Science
Under the advisement of Dr. Insup Lee and Dr. Deepak Gangadharan

Abstract

Recent developments in semi-autonomous and autonomous technology for large trucks have enabled them to drastically cut fuel costs by platooning. Platooning is a strategy by which heavy duty vehicles (HDVs) closely follow one another to coordinate acceleration and decrease the negative effects of aerodynamic drag. As autonomous HDVs roll out in coming years, it will be difficult for them to find natural opportunities to platoon on highways. To solve this coordination problem, we built a cloud platform and designed algorithms that make and distribute platooning decisions for autonomous HDVs so that trucking companies can capitalize on the emergent fuel-savings. Our distributed algorithm leverages real-time routing data, speed profiles, and regional information to optimize platoons for fuel cost savings. Our solution demonstrates that a large-scale coordination system over US highways will lead to increased fuel-saving opportunities for the industry as platoons roll out over the next decade.

Motivation

Autonomous vehicle (AV) technology is approaching road-readiness and regulatory bodies have begun preparing for AV adoption in the form of controls, liability, and infrastructure regulations. Research on platooning has also increased dramatically in recent years. HDV platooning has theoretical benefits of up to 15% fuel savings, and these savings are mostly emergent due to smaller and smaller gaps between vehicles made possible by mature AV technology. Platoons enable a host of benefits beyond fuel savings, including decreased traffic congestion, less vehicular wear and tear, and more. Scholars in the AV space believe that considering these advancements, driverless platoons will soon be a reality.

However, as we move towards driverless platoons, there lies a gap between stakeholder needs and the current academic research. While academic research has generally focused on the lower-level mechanics of platoon formation, it often fails to address the game theoretic aspects of vehicle cooperation that a higher-level centralized coordination system on when platoons should form could offer. Which conditions should be considered and when is it worth forming/splitting a platoon? For example, given a horizon for analysis, a system could determine whether an HDV should merge with one platoon based on potential fuel savings (after costs of accelerating to merge) or by maximizing time in a platoon. Maximizing time in a platoon is a beneficial strategy as HDVs can enjoy the fuel saving benefits for extensive periods in their journeys. This coordination problem became the core of our technical work.

By introducing a centralized platform which addresses these concerns and facilitates the coordination of HDVs, trucking companies will have increased opportunity to cooperate (platoon) with other fleets of HDVs. This is a newfound opportunity for trucking companies with huge potential upside given that the trucking industry spends \$200B on fuel. Additionally, 90% of trucking companies have fewer than 6 HDVs, which restricts them from coordinating efficient platoons amongst themselves. This means that even though industry trends and research suggest that HDV platooning is the future of the trucking industry, most trucking companies in the United States will not be able to partake and enjoy the benefits without a centralized platform which coordinates with all HDVs irrespective of company affiliation.

A final need that our platform addresses is that of auditability for various stakeholders, including legislative bodies, insurance companies, and AV/HDV manufacturers. Regulators are in uncharted territory when it comes to AV technology. While companies, such as FedEx and Volvo, are conducting their own platooning research, there is great regulatory incentive for one unbiased, 3rd party player. This reduces the complexity and the need for integration from different manufacturers systems. A single 3rd party platform for coordination also introduces the ability to insert incentives and game theoretic guarantees into the network.

Our project addresses these three needs: the gap in platoon decision-making research, the need for a single and unbiased coordination agent, and the regulatory need for auditability.

Technical Approach

Our solution consists of three systems a traffic simulator, a cloud server hosting our decision algorithm, and a business dashboard which communicate with one another to deliver platooning coordination decisions to vehicles participating in our network. Our modular approach allows flexibility for future extension. Additionally, we built a model to represent road networks and vehicles in order to test and evaluate our decision platform.

Model Design

We modeled road networks with an intuitive graph structure that captures the inherent information of connections and routes in an existing network while also highlighting areas where platoons would likely form. Any given road network is translated into a directed graph. Edges in this graph consist of stretches of road that either join roads at an intersection, exit, or on-ramp, or a stretch of road with a maximal length specified by a hyperparameter. Nodes then represent connections between edges; in most cases, node will have an in-degree or out-degree that is greater than one and thus serve as a place to pool incoming edges. We chose this representation as it provides an easily digestible mechanism of looking at road networks and lends itself naturally to massively parallel algorithms that operate on the various nodes. Given the nature of our system a cloud network with the ability to scale being able to shard the decision making such that it could operate on individual nodes was crucial to ensuring real-time decisions could be delivered to participating vehicles. Considering the scope of this project, a number of abstractions and simplifications were made; simple traffic patterns, all vehicles are known, and all vehicles are of the same type. Relaxing these constraints would make for compelling future work.

Simulator

Using the above, we created a python simulator with accompanying visualizer to serve as a mechanism to replace real world data and a testing ground for vetting our algorithm. Our simulator leverages the above defined model to simulate timesteps of actions in a real road network (Appendix: Technical Approach—Simulator Snapshots). Given a specification of a road network and a set of vehicles with relevant routing information (both specified via JSON files), a corresponding graph is created and populated with vehicles. The simulator assigns to each edge in this graph a maximum and minimum speed limit, a flow rate, and an ID. Vehicles initialized in the simulator each have an immutable route, a current speed, a platoon allocation, and an ID. Each timestep (which is configurable, but for our purposes was set to 10 milliseconds) all vehicles in the simulation have their state updated. Updates consist of a change to a vehicles position in based on current speed, a fuel cost calculation for that change in position (Appendix: Technical Approach—Cost Function), a ping to the cloud server with updated information, and a request to the cloud server for updated routing and platoon allocations. If a vehicle moves to a new road, speed information is also updated to take into account speed limits (by default, the speed of the vehicle will be set to the flow rate for the new edge it is traversing). Included with the simulator is the ability to connect to a server for decisions regarding platooning allocations. The simulator both encodes and decodes JSON messages (Appendix: Technical Approach—JSON Simulator Vehicle Specification, JSON Simulator Network Configuration), which it uses to update its current state. Failure to connect to a server will have a simulation run without any platooning decisions. Platoons are handled through the tracking of specified ID sequences which are shared between the simulator and server (it is left to the server to assign IDs). Platoons, extending the same model as HDVs, follow the same behaviors as individual vehicles in terms of speed assignments.

Coupled with the simulator is a visualizer for depicting the behavior of the underlying model in a user-friendly manner. The graph structure is represented as a real road network. The visualizer depicts platoon behavior by offering visual cues to the different parts of the platooning process. Vehicles traveling independently are assigned to be red, vehicles in or maneuvering to merge into a platoon assigned to be yellow and converge on the road network.

Platoon Decision Algorithm

Our platooning decision algorithm was informed by the various constraints and opportunities afforded by the structure of our overall solution to the problem. Given that our algorithm is purposed to deliver real-time decisions to vehicles actively on the road, our chief concerns in designing a solution were computation time for processing platoon decisions, stability of decisions and optimality. Unfortunately, deciding optimal allocations of platoons for vehicles in a network at a global level is an NP-Hard problem. Thus, our approach looked to optimize over localities – in our case defined by nodes. Given that our algorithm runs on a cloud platform, we make use of the massively parallel structure of our graph to define an approach that can be run independently on each node – thus allowing as many instances as needed to run parallel versions of the algorithm (Appendix: Technical Approach—Algorithm Specification). The decision making process is split into server/cloud and node-level decisions. The algorithm lives on a cloud system operating in real time. First, the algorithm opens a polling period where vehicles register their information with the system. Once the polling period closes, the algorithm aggregates the registered vehicles at relevant nodes. The aggregation process involves an algorithm called k-lookback (Appendix: Technical Approach—Algorithm Specification-Algorithm 6) – for the purposes of this paper all results listed are arrived at with a k-lookback of 1. Once vehicles are aggregated, the algorithm is called on each node in the network and decisions are stored. Platoon allocations are distributed as per request of each individual client. If a client requests an allocation before the algorithm has been run with that clients new information, the algorithm is run to generate the new allocation and the given cache is updated.

Dashboard

Lastly, the dashboard serves as the business portal. The dashboard connects to the cloud platform to access information on the performance of a particular fleet. In addition, it also provides a number of different visualizations over varying metrics a fleet owner would be interested in. Stressed in the reports are the amount of CO2, the amount of time spent in a platoon, and the total cost savings experienced for this month and the previous 12 months. Since the dashboard has a real-time connection to the current state of the server, fleet owners can also track their vehicles, see route progress, and see which vehicles are platooning. Serving as the direct link to the customer, the dashboard is the portal through which accounts are managed and subscriptions renewed (Appendix: Client Dashboard).

Evaluation and Discussion of Findings

Our projects success is measured by the ability of our algorithm to form cost-saving platoons. To evaluate our project, we ran several simulations over multiple example highway systems and compared the fuel and energy usage of HDVs with and without platooning.

First, we used VENTOS, a research-grade traffic simulation tool, to evaluate the cost savings of autonomous vehicle platooning compared to Cooperative Adaptive Cruise Control (CACC), the best currently feasible alternative to platooning where each vehicle matches the speed of the vehicle in front of it using a computer controlled system. As you can see in the CACC vs Platooning table (Appendix: Results—VENTOS Validation), autonomous vehicle platooning created 47.5% more fuel savings and 31.9% less CO_2 emissions than CACC over a straight stretch of highway. Autonomous vehicle platooning also reduced the average acceleration profile of each vehicle by 31.9% relative to CACC, indicating that autonomous vehicle platooning cuts down on the variance in acceleration of each vehicle. These results validated our approach to build our platform and savings estimation function around autonomous vehicle platooning as opposed to CACC. Even though we believe that our high-level platooning algorithm could provide cost savings for trucks using CACC as well as autonomous vehicles, we wanted to find the best solution to a problem in order to generate the highest return on investment for potential clients.

After building our own HDV simulator that could communicate with our algorithm through a server, we ran several simulations over different highway networks to determine average fuel savings of our platform. We ran several simulations over multiple road networks with various numbers of vehicles. For the purpose of this report, we include results from two networks at opposite extremes: a very simple network with just two intersections, and a more complicated road network with multiple intersections. For each network, we ran our simulation with the algorithm turned on, then ran it with the algorithm turned off and compared the fuel usage of each vehicle to its counterpart. For our simple network, the average HDV used 15.96% less fuel than the equivalent HDV used without platooning, according to our cost model. For our complex model, the average HDV used 27.32% less fuel than the equivalent, non-platooning HDV. These results confirm our hypothesis that our algorithm coordinates effective platoons, and corroborates our findings on the estimated fuel savings for HDV platoons. We have included two charts depicting the fuel savings from each of these simulations as a function of percentage of route completed by each vehicle (Appendix: Results—Simulator Results). As you can see, the average fuel savings difference increases the most for the middle of the route (20-80% route completion) which is when a majority of the platoons were formed.

Based on our results, we were able to successfully meet the needs of our users and produce platooning decisions that lead to 15-25% in average fuel savings, in broad variety of scenarios. The user is able to access and view these savings for their fleet of vehicles using our business dashboard (Appendix: Client Dashboard). Additionally, because the algorithm is a modular, self-contained component that runs independently on its own server, it will be easy to swap out the simulator for real autonomous vehicles when the opportunity presents itself.

Ethical Considerations and Societal Impact

As with most developing technologies in the sphere of autonomous vehicles, safety is the first and foremost ethical consideration in building our product. As we work with large 18-wheeler trucks, changes we make to vehicular speeds and platoon groupings should be validated with safety precautions. We addressed this issue in our algorithm by ensuring that vehicles maintain a level of speed stability by ruling out any potential decisions that involve speed changes that are too sudden, greater than a maximum threshold of flow speed, or less than a minimum threshold of flow speed. Given these built-in safety precautions, our algorithm is able to produce decisions and speed profiles that promote road stability and prevent collisions. Furthermore, our space is one-step-removed from the mechanical and control systems research that currently dominates this field, the extremely time-sensitive safety hazards are out of our scope. As our platform will be used by autonomous vehicles that have these time-critical safety systems in place such as object sensing, localization, and redundant systems, our safety concerns involve mostly deferrable decisions that are made long before the resulting action is actually carried out. This is addressed by designing our algorithm such that platoon groupings for a certain road in a road network are generated on the edge before they must maneuver to platoon. The algorithm is then repeatedly rerun with updated road conditions over time, ensuring that our decisions continue to be safe throughout the vehicles trip.

Beyond safety, another ethical concern is the disturbance to other vehicles in the system that may not use our platform or may not be autonomous. We address this concern in multiple ways: maintaining or increasing road stability and the use of platooning-specific lanes. As discussed above, we designed our algorithm to maintain road stability by thresholding against sudden changes of speed or final speeds that are much greater or lower than the current traffic flow speed when a vehicle is trying to catch up to a platoon. In addition, once the vehicles catch up to each other to form a platoon, there are fewer sudden acceleration and deceleration events within the platoon. As such, road stability is increased, thereby minimizing disturbances to other vehicles in the network. Furthermore, with increased regulatory buy-in over the past couple of years, new platoon-only lanes are currently opening up for testing and future adoption in states such as Pennsylvania, California, and Georgia. As more of these platoon-only lanes are introduced, we believe that the disturbance to other vehicles will be further decreased.

Our platform offers many secondary societal effects that can positively impact our highways and our environment in the future. By optimizing for reduced fuel costs, we are also optimizing for reduced greenhouse emissions. The trucking industry currently accounts for 1.6 billion metric tons of carbon dioxide a year worldwide (5.75% of total greenhouse gas emissions). With conservative estimates of around 10-15% of reduced fuel costs, platooning can enable a much greener trucking industry. Furthermore, platooning also allows for less congestion by increasing road capacity and decreasing the distance between trucks. Given that congestion continues to be a worsening problem, platooning offers a way to improve current conditions without having to rebuild and redesign current highways and disrupt day-to-day travelers.

Business Plan

The Market for HDVs

Much of this project is built on the assumption that there will be widespread roll-out of autonomous 18-wheelers in the next decade. We hold that this is a safe assumption because of the state of research on autonomous HDVs, regulatory buy-in, and a competitive market need. The state-of-the-art on autonomous HDV research largely indicates that the hardest problems (controls, communication, and sensing technology) have been solved, and the real challenge is making them road-ready. To this, regulatory bodies have opened their roads enough that these technologies are being actively tested today. Lastly, we see a strong market draw from the trucking industry. The revenue model in the trucking industry can be characterized by its low margins and high operating costs. Fuel represents 39% of operating costs, and labor another 26%. Autonomous HDVs directly impact labor costs, and since the technology needed to platoon is not vastly more complicated than the baseline technology needed to self-drive, it is likely that vehicle platooning ability will be a competitive feature of these vehicles, as it can help reduce high fuel usage and prices. In this paper, we do not discuss the electric vehicle market, though platooning in this market can still have significance as a reduction in amount of refueling (charging) needed, and hence the cost of charging and time-to-destination.

The trucking industry is not characterized by oligarchic companies, but rather by many small independent contractors. There are over 500,000 trucking companies in the United States, and 90% of these companies have less than 6 drivers. This fact is the most important thing that drives our value proposition. It is unlikely that trucks within a single company, not even the largest carriers in the country, will be able to platoon among themselves without fundamentally redesigning their routes to keep their vehicles together. Dave Jackson, CEO of the nations largest for-hire carrier (Knight-Swift Transportation Holdings Inc.), said, Platooning expects an inconsistent world to act consistently. To get two drivers, two loads going to the same location at the exact same time, it just doesnt happen very often. This is true platooning is inherently a cooperative behavior, yet the individuals in this network are largely operating independently. Any kind of intelligent platooning (such as choosing the best vehicle to platoon with and looking for opportunities outside of a vehicles immediate vicinity) require sharing of information that might be considered business-critical (destinations, routes, fuel information), and require talking to some platform which can communicate to vehicles far enough from each other that they cannot communicate directly. This indicates the need for a large-scale platform vehicles can communicate to which will assist them in finding viable platoons.

Value Proposition of a Large-scale, Independent Coordination Platform

The market analysis above indicates that when autonomous HDVs launch nationwide, for platooning to be viable, there will already need to be a large-scale coordination system in place. We think that by being the first mover, there will be a couple fundamental mechanisms that allow us to capture a meaningful margin of the savings trucking companies experience due to platooning:

1. The more vehicles on the platform, the more and better the platooning opportunities that can be found.
2. Without participation in a platform, a vehicle cannot independently capitalize on meaningful platoon savings.
3. Partnerships with truck manufacturers and design dominance can increase switching costs for trucking companies.

These mechanisms together can work to secure us from competition by competing platforms, and help us maintain market dominance. At the same time, because we enable the platoons at a fundamental level, we can directly charge for the expected benefits when we form platoons. If a carrier switched to a platform with a smaller network, their platoons would generally not be as cost-effective as ours.

We project revenues based on our 2030 estimates for market opportunity. Its important to remember that there are currently no autonomous HDVs operating on todays highways. This means our revenue timeline is highly dependent on the timing of the AV adoption curve, which is hard to predict, but we believe will be reach a critical mass by 2030. We expect $20MM revenue in 2030 (Appendix: Revenue Model). Revenue growth beyond this point will be tightly coupled to autonomous HDV adoption, which is likely to see an inflection point in the mid 2030s. In 2040, our revenue may have grown by a magnitude as most new trucks purchased are autonomous and have had time to replace old vehicles. Overall, the trucking market as a whole is expected to grow 20% in the next decade, but because of the incredible size and maturity of the market today ($730Bn yearly revenue), market growth in our sector is not an important consideration point for our product.

Revenue Model

Our revenue model will be a monthly per-vehicle subscription with a per-allocation payment structure. The monthly subscription will be a low fee that gives access to the platform to our customers and gives them access to our suite of dashboards. Our business will provide these dashboards as part of the information we already have for our main product. These include fleet tracking, detailed information regarding fuel usage and cost and saving projections regarding fuel usage. While we do not talk about these dashboards in detail in this paper, they exist to make our platform a more complete solution for our customers and to increase switching costs.

The largest driver of revenue is the per-allocation payment structure. How this works is essentially that whenever we give an allocation, we charge the company (or pay them), based on the specifics of that allocation. This billing will of course be automatic, but the reason this billing structure is fundamental to the model is because of how incentives work in the algorithmic design. The number of platooning opportunities is much higher when considering all platoons with net positive benefit, instead of only the platooning opportunities with positive benefit for every vehicle in the allocation. Since our algorithm finds the former, it may give an allocation that will cause a vehicle to use more fuel. In this case, the algorithm will pay them to join the allocation. This per-allocation rule gives a game-theoretic guarantee that whenever we give an allocation, it is in a companys best interest to accept the offer and join the platoon.

The Pathway to Market Dominance and Competitive Analysis

Today, there are no independent companies building this large-scale platform as their fundamental business model. The companies developing similar platforms are those companies which themselves are designing the autonomous vehicle technology. Peloton is the largest autonomous vehicle and platooning company which is also directly creating IP related to the coordination problem. The other AV companies are the truck manufacturers themselves, of which Volvo and Daimler are among the most active. There are two main reasons why, as an independent company, we can be an asset to the truck manufacturers and competitive to Peloton. The first is that the core competencies related to this platforms are largely different than those of building AVs; this platform requires competency primarily related to large scale distributed systems and algorithmic game-theory, which does not overlap much with autonomous vehicle development. The second is that when the company making the platooning technology is also the company creating the cooperation infrastructure, there are disincentives for other platooning technology companies to make their technology compatible with that cooperation infrastructure, as doing so supports a direct market competitor.

The takeaway from this is that: as an independent technology provider, we can work in collaboration with multiple autonomous vehicle manufacturers simultaneously, to ensure that their technologies are largely compatible. Each would like to cooperate with us, as we save them from having to do development outside of their main research areas, and give them a promise of the scale needed to make their platooning technologies, and hence vehicles, more attractive. In this way, vehicle manufacturers will become our partners, and Peloton will remain our competition. But it also competes with the vehicle manufacturers, whom develop their own AV and platooning technology. The pathway forward becomes:

1. Develop partnerships with vehicle manufacturers to guide their development
2. Partner with regulators to give them a key access point into monitoring AVs on our highways (an important part of becoming a dominant design)
3. As HDVs are sold, partner with the largest trucking carriers in order to get the scale needed to create cost-effective platoons
4. Onboard the hundreds of thousands of small carriers, who are the stakeholders who operate on the thinnest margins and would hence have the highest incentive to participate in our platform

Conclusion

In conclusion, our team successfully built and tested a scalable platform to solve the problem of platoon coordination. The algorithm we have developed, along with the simulation and server infrastructure, can provide insight into the advantages of autonomous vehicle platooning and has the potential to revolutionize the entire trucking industry. Our positive results from testing our algorithm on various road networks has encouraged us to continue developing the project in the near future. We are currently engaging in discussions with our advisor over how best to proceed, and a few of researchers and industry representatives have expressed interest in working with us to develop the project further. Regardless of the final state of our project, the code bases we have built and the innovations we have discovered will continue to benefit ongoing platooning research and any major players in this technological space. We are very excited to see where this project leads us as we continue our drive down the highway of progress.

Acknowledgements & Future Work

We would like to show our gratitude to Dr. Insup Lee and Dr. Deepak Gangadharan for their continuous support and technical guidance. A huge thank you to Dr. Ani Nenkova for all of her advice and help in addressing potential stakeholders' concerns. As the autonomous HDV industry matures, we expect to see projects in coordination that are less theoretical and take into account the regulatory environment, but will likely build on the baseline algorithms present in this project and other early research.

Appendix

Technical Approach

Algorithm Specification Note: HDV implements Vehicle, Platoon implements Vehicle

Algorithm 1 Overall Algorithm

```
procedure ALG( )
    for all V_i do                                              ▷ Parallel
        root ← MakePlatoonTree(V_i)
        root.vehicles = root.vehicles + CollapseTree(root)
        return FindPlatoons(root.vehicles)
    end for
end procedure
```

Algorithm 2 DFS Platoon Formation Subroutine

```
procedure COLLAPSETREE(vertex)
    if vertex.children.length is 0 then
        return vertex.vehicles
    end if
    for all c : vertex.children do
        c.vehicles = c.vehicles + CollapseTree(c.children))
        if c.vehicles.length is 0 then
            return null
        else if c.vehicles.length is 1 then
            return c.vehicles
        else
            return FindPlatoons(c.vehicles)
        end if
    end for
end procedure
```

Algorithm 3 Platooning Tree Creation

```
procedure MAKEPLATOONTREE(V_i as vertex)                 ▷ Create platoon tree from a vertex
    c_0 ← vertex.vehicles.get(0)                                  ▷ Represents C_i[0]
    iter ← 0
    traversed ← ∅
    for r: c_0.route do
        separate ← ∅
        for c: vertex.vehicles - c_0 do
            if c.route[iter] ≠ r then               ▷ Identify vehicles taking a different route
                vertex.vehicles ← vertex.vehicles - c
                separate ← separate.add(c)
            end if
        end for
        if separate ≠ ∅ then                                  ▷ If separated, split tree
            newPV = new PV(null, ∅, vertex.vehicles, c.route[0: iter])
            newPV.children.add(makePlatoonTree(vertex.vehicles, iter, newPV))
            newPV.children.add(makePlatoonTree(separate, iter, newPV))
        end if
    end for
    return newPV                                   ▷ Return the top level platoon vertex
end procedure
```

Algorithm 4 Platooning Tree Creation Cont.

```
procedure MAKEPLATOONTREE(vehicles, timestep, root)    ▷ Continue using results from previous iterations
    c0 ← vehicles.get(0)
    iter ← 0
    traversed ← ∅
    for r: c0.route do
        separate ← ∅
        for c: vehicles - c0 do
            if c.route[iter] ≠ r then    ▷ Identify vehicles taking a different route
                vehicles ← vehicles - c
                separate ← separate.add(c)
            end if
        end for
        if separators ≠ ∅ then    ▷ If separated, split tree
            newPV = new PV(root, ∅, vehicles, c.route[timestep:timestep + iter])
            newPV.children.add(makePlatoonTree(vehicles, timestep + iter, newPV))
            newPV.children.add(makePlatoonTree(separate, timestep + iter, newPV))
        end if
    end for
    return newPV
end procedure
```

Algorithm 5 Platooning Decision Algorithm

```
procedure FINDPLATOONS(vehicles)    ▷ Finds valid platoons over a given set of edges
    largestBenefit = 1
    v1 = null
    v2 = null
    while largestBenefit ¿ 0 do
        largestBenefit = 0
        for c_i ∈ vehicles do
            for c_j ∈ vehicles do
                benefit = MaxSlowdownMinSpeedup(c_i.vehicles+c_j.vehicles) / (c_i.size+c_j.size)
                if largestBenefit ¡ benefit then
                    largestBenefit = benefit
                    v1 = c_i
                    v2 = c_j
                end if
            end for
        end for
        vehicles = vehicles - v1 - v2 + {v1, v2}
    end while
    return vehicles
end procedure
```

Platoons are discovered via an iterated approach. At each iteration, all $\binom{n}{2}$ combinations of two Vehicles are considered (which can be a platoon or HDV). The discovered potential platoon with the largest benefit per member is then formed. The vehicle list is updated to remove the atomic vehicles from the list and replaces them with the resultant platoon. For a set of n vehicles, this algorithm will find potential platoons in $\mathcal{O}(n^3)$.

Algorithm 6 K-Lookback

```
procedure KLOOKBACK(nodes, vehicles, k)        ▷ Finds vehicles within radius of node
    for n_j ∈ nodes do
        for v_j ∈ vehicles do
            for i = 0; i ¡ k; i++ do
                if v_j.route[i].endNode() == n_j then
                    n_j.incoming+ = v_j
                end if
            end for
        end for
    end for
end procedure
```

Cost Function

1. Cost Function for Vehicles in Platoons

$$Cost = 0.00028572 \times speed^2 + 0.325752 \times speed - 10.9569$$

2. Cost Function for Individual Vehicle

$$Cost = -12.346 + 0.374518 \times speed$$

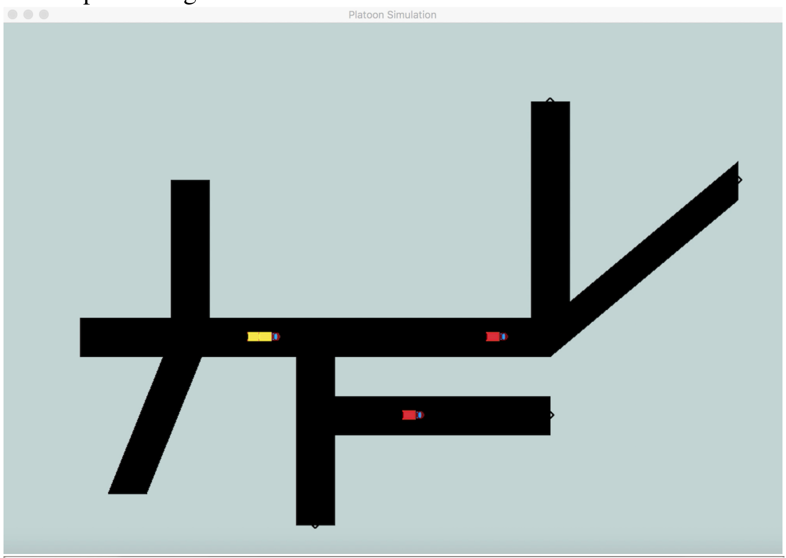
JSON Simulator Vehicle Specification

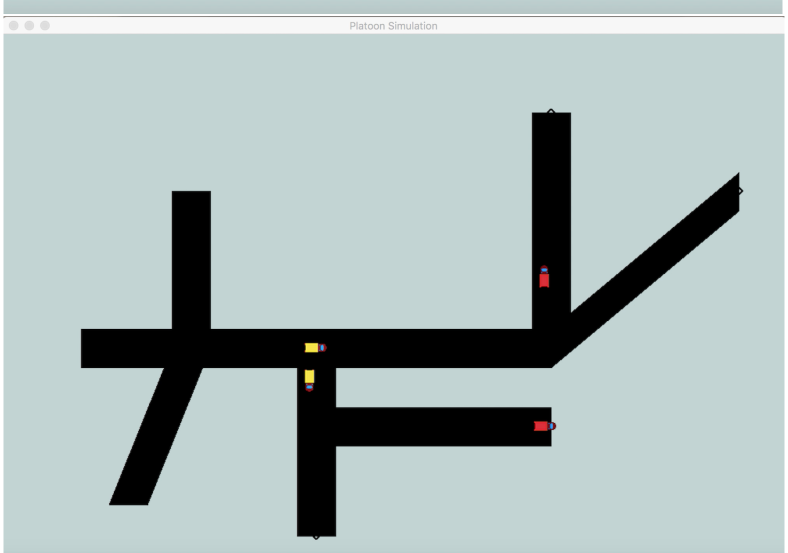
```
"$schema": {
    "id": "unique int",
    "route": "list of unique edge ids",
    "starttime": "time vehicle begins in seconds from simulation start"
}
```

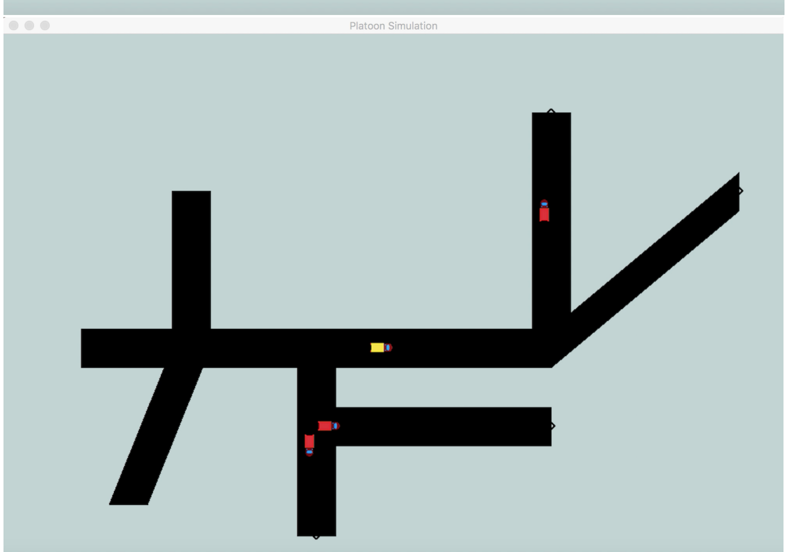
JSON Simulator Network Configuration

```
"$schema": {
    "id": "unique int",
    "max_speed": "double mph",
    "min_speed": "double mph",
    "flow_speed": "double mph",
    "length": "double miles",
    "outgoing": "list of ids",
    "x_position_start": "miles from left on visualizer",
    "y_position_start": "miles from top on visualizer",
    "x_position_stop": "miles from left on visualizer",
    "y_position_stop": "miles from top on visualizer"
}
```

Simulator Snapshots This is an example of a complex network. The yellow vehicles depict platooning and the red vehicles are not platooning.





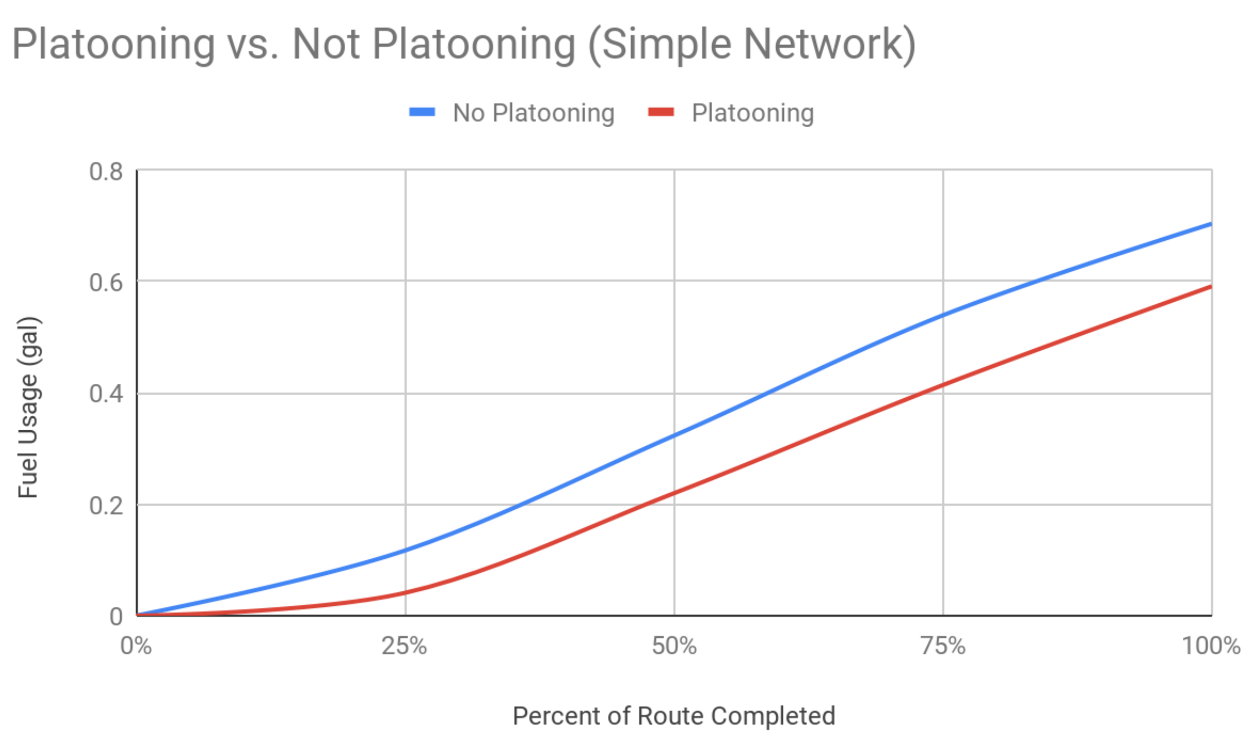


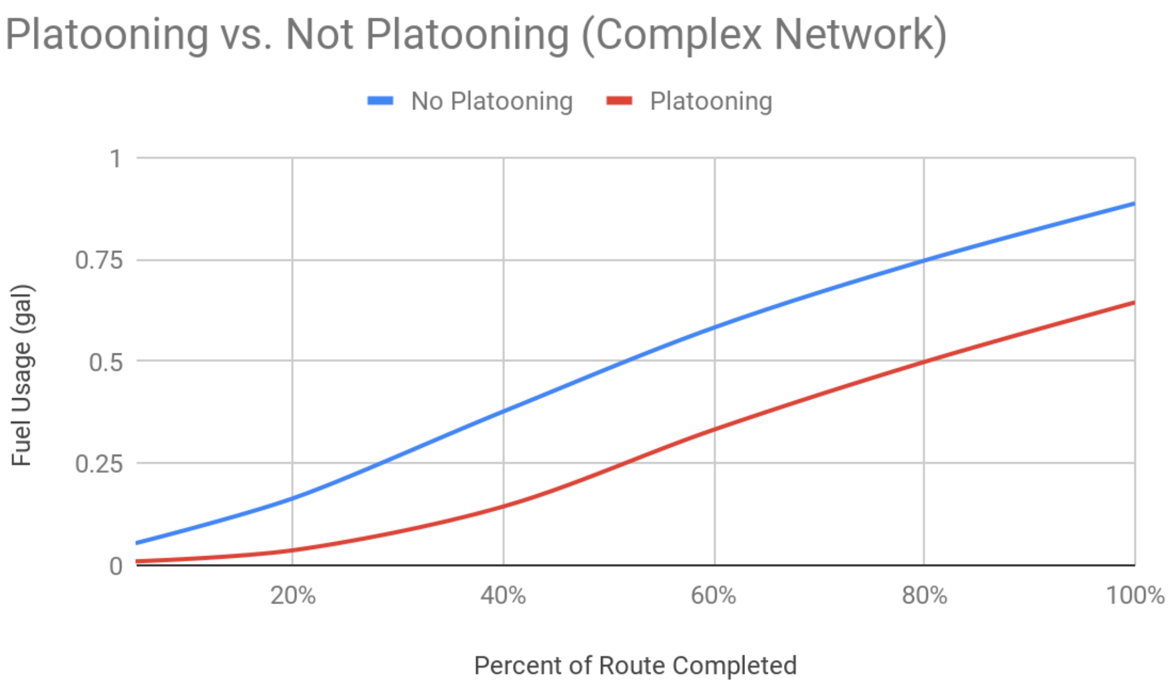
Results

VENTOS Validation: CACC vs Platooning These results were generated by a research-grade simulation tool, VENTOS, on HDVs using autonomous platooning or CACC over a straight, single lane highway. We believe the difference in fuel savings of this simulation (46.7%) is significantly lower than the generally accepted difference (15-25%) due to the ideal conditions of the highway model used.

	Cooperative Adaptive Cruise Control (CACC)	Autonomous Platooning	Percent Decrease (%) wrt CACC
Fuel Usage (L)	4.20	2.24	46.7
Speed (m/s)	11.7	11.3	3.42
Acceleration (m/s^2)	0.093	0.064	31.94
Front Space Gap (m)	9.14	8.15	10.83
CO2 Emissions (g)	9780	5200	46.83

Simulator Results The top image corresponds to a simple road network, and the bottom corresponds to a more complex road network.




Platooning vs. Not Platooning (Complex Network)
No Platooning
Platooning
Fuel Usage (gal)
1
0.75
0.5
0.25
0
20%
40%
60%
80%
100%
Percent of Route Completed

Revenue Model

Marginal fuel cost / mile (2030)	\$0.3
Miles driven in the U.S. in 2030	200 billion
Total HDVs with sufficient technology to participate (long depreciation times slow adoption)	5%
Percent time spent in an allocation	65%
Percent saving from platooning (on lower end of data)	10%
Market Penetration (other large competitors and non-cooperating autonomous HDVs)	50%
Margin	20%
Annual Projected Revenue	**\$19,500,000**

Client Dashboard