Algorithms and Architectures: A Case Study in When, Where and How to Connect Vehicles

Article *in* IEEE Intelligent Transportation Systems Magazine · January 2017

DOI: 10.1109/MITS.2017.2776142

Josh Siegel
Michigan State University

Sanjay E. Sarma
Massachusetts Institute of Technology

Algorithms and Architectures: A Case Study in When, Where and How to Connect Vehicles

Joshua E. Siegel, Dylan C. Erb, and Sanjay E. Sarma
Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Connected vehicle technologies and applications must coevolve. Existing applications are often suboptimally implemented, stemming from a dearth of design approaches considering resource use, sensor selection, and communication method. Such design tools are necessary to optimize connected vehicle implementations, as applications may have varied impact on individual vehicle and fleet performance, efficiency, and comfort depending on their technical and algorithmic implementation. This paper first introduces several key considerations in connected vehicle design, then explores the implications of varying input richness, connectivity methods, and data availability on an example application predicting automotive idle times. We illustrate common design tradeoffs by evaluating the predictor's accuracy for different implementations, providing developers a look at a design approach that will serve as a useful example framework for future application development. We close with a simplified cost/benefit analysis for our demonstration application to illustrate how feasible, cost-effective application implementations might be identified.

***Index Terms*—Intelligent Transportation Systems, Automotive Engineering, Automotive Applications, Automotive Electronics, Telematics, Connected Vehicles, Design.**

I. CHANGING DESIGN NEEDS

Since the introduction of automotive networks, vehicles have used sensing and computation for local optimization [1]–[3]. Today, Original Equipment Manufacturers (OEMs) rely on software to improve vehicle efficiency, reliability, and performance [4].

Consumer demand for smarter cars and government pushes for improved safety have recently led OEMs to interconnect vehicles and infrastructure, with applications from collaborative control, to data-informed efficiency improvements, to collision avoidance [5], [6].

More than ever, engineers must treat vehicles as nodes within larger networks. This paper explores common design considerations and tradeoffs in developing connected applications.

II. DEVELOPMENT CONSIDERATIONS

A changing technological landscape has made vehicular applications that take advantage of connectivity, sensing, and computation increasingly practical. These advances do facilitate new functionality, but little attention has been paid to ensuring optimal hardware and software implementations.

We argue that this is due to the lack of common design criteria during algorithm design and when selecting sensing, computation, and radio technologies. This section identifies common design considerations, and examines how application performance changes with in-car technologies, remote server architectures, and communication methods.

A. Computation

Selecting computing architectures for durable goods like vehicles is difficult. Engineers must carefully balance the capabilities and location of storage and processing resources to create an affordable, resilient, and scalable platform.

Manuscript received November 12, 2016; revised March 20, 2017. Corresponding author: J. Siegel (email: j_siegel@mit.edu).

TABLE I
LOCAL STORAGE CAN BOTH SUPPORT REALTIME AND DELAY-TOLERANT APPLICATIONS. WITH CURRENT RADIO TECHNOLOGIES, REMOTE STORAGE LATENCY IS TOO HIGH FOR REALTIME APPLICATIONS.

	Realtime	**Delay-Tolerant**
Local Data	Feasible	Feasible
Remote Data	*Infeasible*	Feasible

1) Storage

Storage refers to the memory used to record data within a vehicle or to store information at a remote location. The amount of storage necessary depends on an application's needs and the duration of record keeping.

Applications using only real-time data may require no in-vehicle storage. Others, like those using maps or historical data, may require gigabytes of local or remote storage. Streaming technology blurs the line between local and remote storage. For example, local storage can be decreased and remote storage can be increased. Streaming remote data increases bandwidth cost and latency, but simplifies sharing data among multiple vehicles.

Local storage has fixed installation costs and negligible upkeep; remote storage has initial and recurring costs based on use, and can be expanded on-demand for a fee.

The life of the vehicle, volume of data stored, bandwidth costs, and latency and data sharing constraints are significant factors in determining an application's storage needs.

Storage design is impacted by key trends: the decreasing cost of providing cellular data [7], [8] and flash memory, and the proliferation of scalable web storage platforms.

As a general rule, remote storage supplies aggregate, delay-tolerant data, while applications using local data for local control will preferentially store information in-vehicle. There are also limitations based on the need for real-time performance, as shown in Table I.

2) Processing Power

Processing power determines how quickly in vehicle computers can solve mathematical problems.

In-car computation is limited by cost and legacy practices. We see these constraints manifest in the cost-driven insecurity of modern vehicles – heightened system complexity to implement encryption or credentialing is economically unattractive [9]. An opposite pull indicates consumer willingness to pay for connected and autonomous vehicles may soon drive computation improvements [10].

Like storage, processing may take place locally or remotely to a vehicle. Generally, in-vehicle computation is used for realtime applications with limited processing needs. Remote computation provides scalability for intensive calculations and delay-tolerant applications.

Just as streaming blurs the lines between local and remote processing, differed computing architectures change how much computation takes place, and where. “Thick clients,” for example, process data locally and transmit precomputed values to a remote server, requiring more in-car computation but limiting bandwidth costs. “Thin clients” require minimal onboard processing, but transmit raw, costly data [11].

Computation has fixed costs for in-car modules and use-based variable pricing for remote computation.

B. Communications

By definition, connected vehicles required extra-vehicular communication. Multiple connectivity technologies offer varied latency, bandwidth, cost, and reliability to support differing application requirements.

1) Latency and Bandwidth

Latency refers to the transit time of data. Many applications, including those related to safety, require low latency to function. Delays may result in data starvation, causing application performance to suffer or fail – leading to potential catastrophe.

Radio technologies have different latency characteristics that vary based on network loading and signal strength. Additionally, data-management techniques like priority queueing may be used to speed message transit. Even algorithmic improvements can speed computation, lowering effective data latency.

Bandwidth (throughput) refers to the rate at which bytes travel across a network. Some applications, like streaming video, require radios capable of high bandwidth. These tend to be costly to install, and by virtue of the fact that these systems transmit more data, may have higher usage costs.

The feasible latency/bandwidth spectrum is explored for several common radio technologies in Figure 1. Green markers are peer reviewed sources; red markers are approximations based on the authors’ own work with the CloudThink platform [12]. In the case of disagreeing values, conservative figures are plotted. Similar feasibility plots are useful tools for other aspects of development. For example, plotting node density versus communication range may help identify networks useful for consensus applications, while a latency versus range plot could illustrate feasible technologies for over-the-horizon awareness applications.

Note the convergence of vehicle mesh networks and future cellular technologies. With broader coverage, falling costs, and decreasing latency, cellular connectivity will become increasingly attractive and may ultimately supplant DSRC as the de facto communication standard for connected vehicle applications.

2) Transmission Reliability

Applications may prioritize communications as being essential or “best effort.” An application that can operate in the absence of connected data, or that can safely cease to function, tends to use best effort communication. If connectivity must be assured, this can impact design choices and significantly raise implementation costs.

One approach to ensuring connectivity is to hybridize technologies. Fusing DSRC and 4G increases hardware costs but allows applications near-assured, low latency connectivity. This synergistic approach is especially useful for applications where mesh density may not be sufficiently high for DSRC-only operation, or in cases where 4G coverage is poor. An example might be a traffic application; a vehicle leaving a rural area but headed into a city might use 4G for congestion information until entering the range of the city’s VANET.

3) Relationship to Computation

Communications and computation must be optimized in concert. While in-car computation offers lower latency and bandwidth cost than remote computation (where bandwidth cost refers to a per-byte fee), platform considerations such as Fog vs Cloud backends can change the possibility space. Fog and edge computing, for example, can result in reduced latency to vehicles relative to the Cloud. Depending on the radio technology used, these may also offer reduced bandwidth cost. This is shown in Figure 2.

C. Extra-vehicular Inputs

Vehicular applications often require outside data. Data availability and freshness present challenges that must be addressed.

1) Data Availability and Freshness

Connected vehicles may require external data, but mesh technologies like DSRC may fail at long distances or high speeds. Sensors may fail in inclement weather. Applications must be able to operate during these and other information outages.

When connected to external data sources, applications must determine how frequently to query for information. High frequency updates drive costs and congest networks but may improve application responsiveness. Infrequent updates mean applications use and respond to outdated information. It is not only near-realtime data must be fresh: databases can also age.

Database freshness can be illustrated by considering in-car navigation systems. These leave the factory with a reference map, and the driver must determine whether to pay for updates or to let the database languish. Frequently, the costs outweigh the benefits.

It is possible to replace or update application databases. Selective updates based on likely use can minimize bandwidth consumption (if a vehicle has never driven outside of

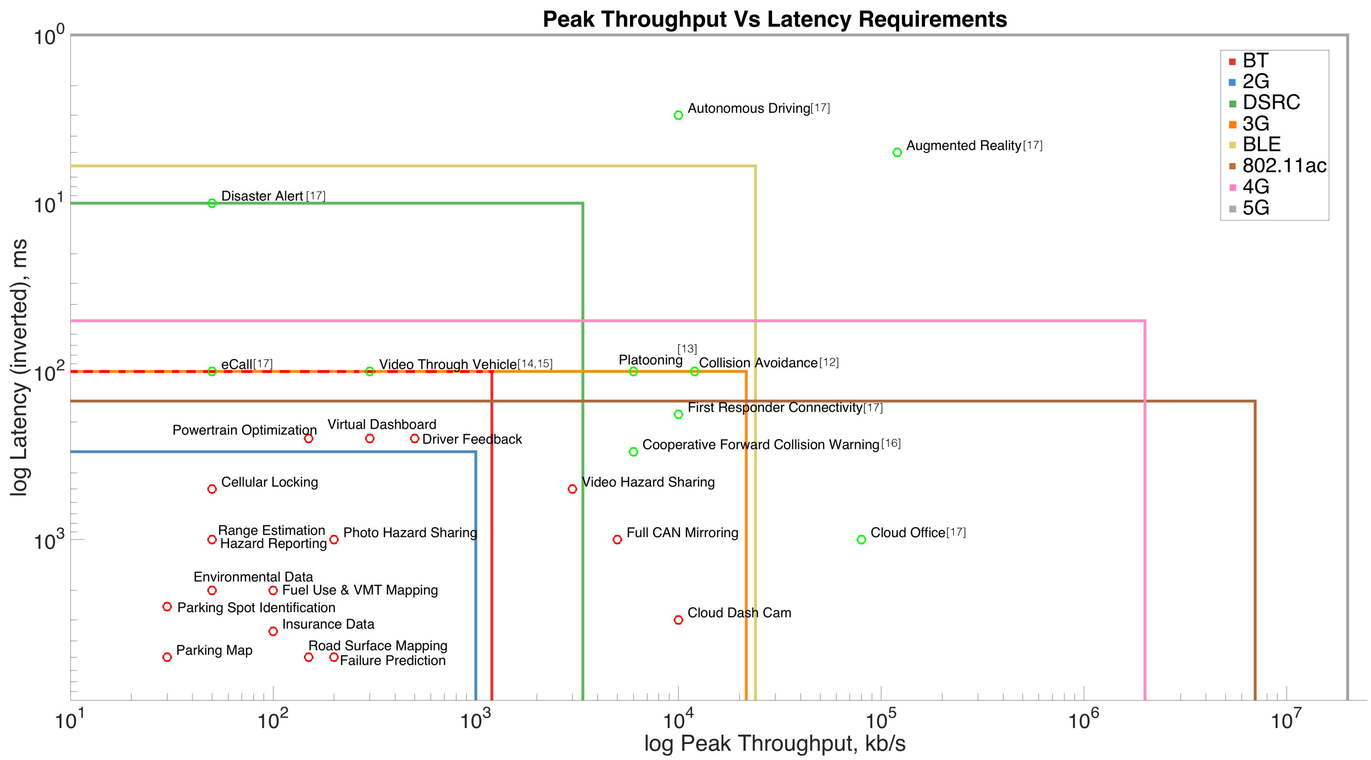

Fig. 1. This plot shows the latency and bandwidth requirements and technology limitations for common connected vehicle applications and enabling technologies. Applications inside a technology's bounding box are considered feasible. Green circles are applications drawn from reference literature [13]–[18] ; red circles are estimates based on the authors' own application development. Note that eCall and First Responder application needs differ – the higher bandwidth requirements suggest the First Responder application transmits additional incident information.

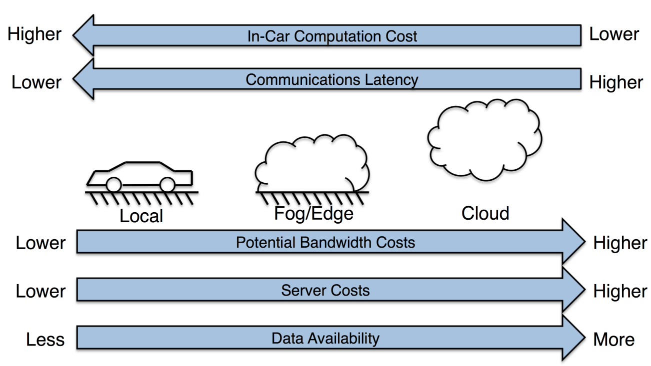

Fig. 2. This figure shows how differing communication architectures can impact the cost of computation, latency, data availability, and more.

Michigan, it wouldn't need California maps). Similarly, delta updates containing only new information (new and changed roads) can minimize bandwidth costs. These approaches may reduce the amount of in-vehicle storage necessary.

D. Solution Cost

Automotive OEMs strive to provide features at a low cost. As vehicles are increasingly differentiated by their networked applications, traditional cost/benefit models break down. Understanding the upfront and ongoing operating cost of vehicle technology can determine application feasibility.

1) Hardware Cost

Applications require computation, communication, sensing, and actuation. Each element contributes to the initial system cost and may have its own operating costs and consumer benefits.

Determining the true hardware cost is difficult, as each element may be support several applications – including applications that had not been envisioned at time of manufacture. OEMs must consider the cost and relative benefit of each hardware component included in a vehicle.

2) Operating Cost

Applications requiring connectivity have ongoing operating costs for bandwidth, remote storage and computation.

Bandwidth and line fee costs depend on technology and are negotiated with telecommunications companies based installed base, data volume, and contract duration. Technologies like DSRC have no bandwidth cost, but may require cellular augmentation to function when peer density is low.

Remote computation and storage pricing depends on data volume and who owns and operates the server (developer, OEM, managed provider). If an application is to be deployed across a large fleet of vehicles, it may become cost-effective to operate one's own servers. Similarly, an OEM may operate servers for multiple applications.

An application's operating cost may also include data access fees. The cost of licensing maps or other data sources can be significant, so some OEMs prefer to generate data in-house.

E. Algorithms

Connected applications must contend with network unavailability, inaccurate data, and sensor outages. Algorithm robustness is especially important for applications relating to safety, where a failure could have grave consequences.

To meet these needs, developers must maintain multiple algorithms and switch as inputs vary. This could be as simple as a routing application using historic traffic data rather than realtime, or could be as involved as an autonomous vehicle handing over control to a human. Developers must also identify the point at which inputs are too sparse, and the application performance will no longer be acceptable.

Therefore, application performance must be quantified. In a simple case, a classification application's performance might simply be its accuracy. In other cases, quantification is more difficult – for example, minimizing driver annoyance. Developers must define an objective function describing an application's utility.

III. Designing for a Case Study: Idle Time Prediction

The most perceivably-beneficial connected applications change how vehicles operate in realtime. To illustrate the earlier-mentioned design considerations, we examine a realtime application controlling an Automatic Engine Start-Stop (AESS) system. The approach and considerations presented are applicable to all connected vehicle applications for which performance is quantifiable and costs are measurable, and for which the solution space is appropriately constrained.

AESS reduces fuel consumption by shutting down the engine at idle. When the driver releases the brake and signals an imminent restart, the engine restarts. This system has become popular, as it helps OEMs meet stringent Corporate Average Fuel Economy targets and gain EPA off-cycle credits [19].

In real-world testing, AESS saves up to 10% of fuel consumed [20]. However, drivers are frequently annoyed by the system's intrusion during short idle events, such as those occurring at a stop sign or a crosswalk [21]. These systems annoy up to 11% of drivers to the point of disabling the feature [19], resulting in a fleet-wide increase in fuel consumption of 3.3%.

We consider the design process for a connected application predicting idle duration and eliminate short shutoffs, improving driver compliance and reducing emissions. Note that while applications vary, this section explores a canonical process for optimal application design and implementation.

A. Predicting Idle Time

We must estimate idle times as soon as the car stops to reduce driver annoyance. From experience, we noted that our location, proximity, and relative velocity differences to lead vehicles provide stop duration cues – a car arriving to a newly-green light might notice accelerating cars at the light and choose to slow down rather than stop, anticipating a short delay prior to moving. This driver may exhibit the same behavior near a crosswalk.

TABLE II
IDM Parameter Values

Parameter	Value
A_i	2 $[ms^{-2}]$
V_{i_0}	20 $[ms^{-1}]$
δ	4
s_{i_0}	2 $[m]$
L_i	4 $[m]$
T	1 $[s]$
B_i	2 $[ms^{-2}]$
$B_{i_{max}}$	6 $[ms^{-2}]$

We simulated and captured real data for the parameters in Figure 3. The simulation used MATLAB models based on the Intelligent Driver Model (IDM) described below [22]. This model predicts the acceleration of a given vehicle based on driver parameters and the relative distance/speed to the next vehicle in front.

For a given vehicle (i), the 1-dimensional position of that vehicle is denoted as x_i, and the velocity of vehicle i is represented by the ordinary differential equation

$$v_i = \dot{x_i} = \frac{dx_i}{dt}. \tag{1}$$

and acceleration is

$$a_i = \dot{v_i} = \frac{dv_i}{dt}, \tag{2}$$

which is modeled as

$$\frac{dv_i}{dt} = A_i \left[1 - \left(\frac{v_i}{V_{i_0}} \right)^{\delta} - \left(\frac{s^*(v_i, \Delta v_i)}{s_i} \right)^2 \right], \tag{3}$$

In (3), A_i is desired acceleration, V_{i_0} is the target speed, δ is the acceleration exponent and Δv_i is the relative speed to the next car ($v_{i-1} - v_i$). s_i is defined as

$$s_i = x_{i-1} - x_i - L_{i-1} \tag{4}$$

where x_{i-1} is the position of the car in front of car i, L_{i-1} is the length of the car in front. Finally, s^* is defined as

$$s^*(v_i, \Delta v_i) = s_{i_0} + max \left[0, \left(v_i T + \frac{v_i \Delta v_i}{2\sqrt{A_i B_i}} \right) \right] \tag{5}$$

where s_{i_0} is the minimum target bumper-to-bumper distance for car i, T is the desired safety time to the car in front, B_i is the target braking deceleration, and $B_{i_{max}}$ is the maximum allowed braking deceleration.

For this simulation, the driver parameters were selected to be representative of a typical city driver. The IDM parameter values utilized in this study are listed in Table II and are based off the values used by Kesting (2010) with changes made to better reflect real-world driving within Boston's city limits (increased desired acceleration, marginally decreased desire for speed and preference for shorter following times).

To validate the results of the simulation, real data were collected using GPS, on-board diagnostics (OBD), ultrasonic sensors, and LIDAR on a test vehicle. GPS was used to record

TABLE III
CONFUSION MATRIX

Actual Class	Classified Short	Classified Long
Short	# True Positive	# False Negative
Long	# False Positive	# True Negative

the vehicle's position and heading, OBD recorded the vehicle's speed, and ultrasonic sensors and LIDAR captured the intra-vehicle distance, velocity, and relative acceleration (on production vehicles, RADAR systems can provide these data). We drove two instrumented vehicles along a fixed trajectory in series to ensure geospatial data density and availability of self and lead-car data at all times.

All of these contextual and historic data were then used for "kth Nearest Neighbor" (kNN) matching to classify an idle as short (less than two seconds) or long (greater than or equal to two seconds). Two seconds was selected as the short/long cutoff because eliminating idles under two seconds results in increased fuel consumption due to higher startup fuel requirements [23].

B. Classification and Accuracy Methods

The kNN prediction model is described in depth in Erb, 2016 [21]. This simple but effective method finds the k nearest neighbors based on Euclidean distance in the training set to the test vector for the idle in question. The data set being classified is assigned to the class most common among its k nearest neighbors, so in the case where the simple majority of the k neighbors are short, then the classifier predicts a short idle, and vice versa.

To evaluate the accuracy of the classifier of predicting short idles, we calculate the true positive rate (TPR) as follows

$$TPR = \frac{\sum True\ Positive}{\sum True\ Positive + \sum False\ Negative}, \tag{6}$$

where the confusion matrix is defined in Table III.

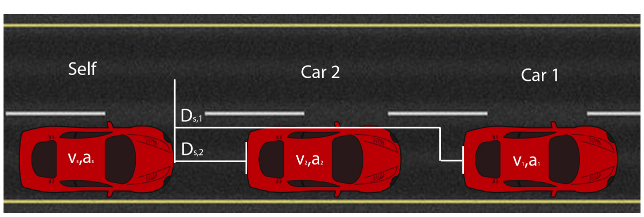

Fig. 3. Parameters measured to inform the Gap Cycle and idle time estimator include the relative position, velocity, and acceleration of three vehicles. The Gap Cycle is an extension of the conventional drive cycle concept, but incorporates an intelligent driver model informed by contextual data such as the distance and relative speed of leading vehicles.

Real and simulated data demonstrated similar prediction accuracy, surpassing 90% short idle prediction accuracy. Therefore, we used additional simulations to expedite data collection used in evaluating the application's sensitivity to design parameters.

IV. OPTIMIZING APPLICATION DESIGN

This section considers the implications of selecting different input data, architectures, and communication methods on start/stop prediction accuracy using the kNN model.

From Figure 1, we determined that all current radio technologies support idle time prediction. We considered range, and eliminate Bluetooth and WiFi from the possibility space. Selecting among the remaining technologies therefore comes down to implementation cost and and benefit (prediction accuracy).

Examined costs include sensing, storage, and computation with variable lifetime bandwidth costs. Input type, timeliness, and availability determine performance. In the following sections, we consider the cost/benefit tradeoffs, using a simple cost model and simulating each permutation 100 times to calculate average accuracy.

Note that while we focus exclusively on financial costs and benefits due to their ease of modeling, factors such as human emotions may have significant impact. We make the simplifying assumption that a reduction in stops over business as usual provides an increase in satisfaction and a reduction in net fuel consumption. More exhaustive models could take these and additional factors into consideration.

A. Cost Modeling

Our cost model includes fixed and variable elements for sensing, storage, computation, and communication for both the vehicle and remote infrastructure. These elements are shown in Figure 4.

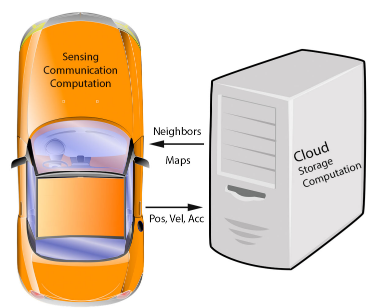

Fig. 4. This figure represents the typical fixed and variable costs for implementing idle time estimation.

This model includes the communication cost for sharing sensor data and receiving map updates, the sensor and in-car storage cost attributable to the application, the cost of the server's bandwidth, and the cost of storing data in the cloud. Variable costs may change with use and over time, as a function of other inputs like map update frequency, database growth, and service life:

$c_{total} = (d_{xmit} + d_{maps})(c_{cell} + c_{server} + c_{cloudstorage}) + c_{sensors} + c_{carstorage}$.

In this equation: d_{xmit} is the data transmitted to enable the application,

d_{maps} is the data required for map updates,

c_{cell} is the variable cost for cellular or other wireless data communication,

TABLE IV
COSTS AND TECHNOLOGY ASSESSMENT ESTIMATES USED IN DETERMINING OPTIMAL APPLICATION ALGORITHM AND ARCHITECTURE.

Supported vehicle lifespan	12 years
Potential idles checked	12,000 / year
Map efficacy loss	5% / year
Map replacement rate	6% / year
Database size	100 MB
Latency (DSRC, 5G, 4G, 3G)	0.5, 1, 2, 3 s
Transceiver cost (DSRC, 5G, 4G, 3G, WiFi)	$800, 600, 400, 200, 30
Data cost (DSRC, 5G, 4G, 3G, WiFi)	0, 0.5, 0.4, 0.2, 0 cents/kb
Sensor cost (GPS, ultrasonic, RADAR)	$50, 100, 500
Cloud bandwidth	$0.09 / GB
Cloud storage	$0.36 GB / year
Car storage	$0.40 / GB

c_{server} is the variable cost of bandwidth at the server side,
$c_{sensors}$ is the portion of the sensor (and computation) cost attributable to the application,
$c_{cloudstorage}$ is the cost of storage in the cloud,
and
$c_{carstorage}$ is the cost of providing the in-car storage necessary for the application to operate.

This model considers assumes 100% of the hardware cost; in practice, it should be allocated pro-rata to each dependent application.

The assumptions supporting cost calculations appear in Table IV. These values are based on government data for average vehicle age and miles traveled as well as the authors' experience in building and deploying connected vehicle hardware and software.

These inputs will generate a lifetime application cost, which will help developers determine whether the benefits warrant the deployment cost.

B. Algorithm Input Data

The kNN idle time classifier supports varied sensor inputs. Using only those with the most discriminating power reduces cost while retaining accuracy.

To illustrate how varied inputs change predictor performance, we modeled three representative cases using information available locally (GPS), from the nearby environment (ultrasonic parking sensors), and from a wider area (RADAR/DSRC). Each was simulated for a range of typical latency and data availability values, with non-availability simulated by eliminating data from the reference set at random.

Figure 5 plots the accuracy as surfaces of varying latency and data availability (sparsity), for input vectors consisting of local data (self position and time), nearby data (self and lead car position, velocity and acceleration), and wide area data (self and leading two cars' position, velocity, and acceleration).

The two-car model is the most accurate overall, with the one car model matching its accuracy at low latencies. The self-only model has overall poorer accuracy, but is robust against latency increases. The self-only model beats the one-car model at intermediate latencies. With one-car prediction, stale data leads to inaccurate predictions (e.g. relying on data from a car traveling through a yellow light).

If cost were not a factor, this plot could be used to determine the best available algorithm for given data inputs and technologies.

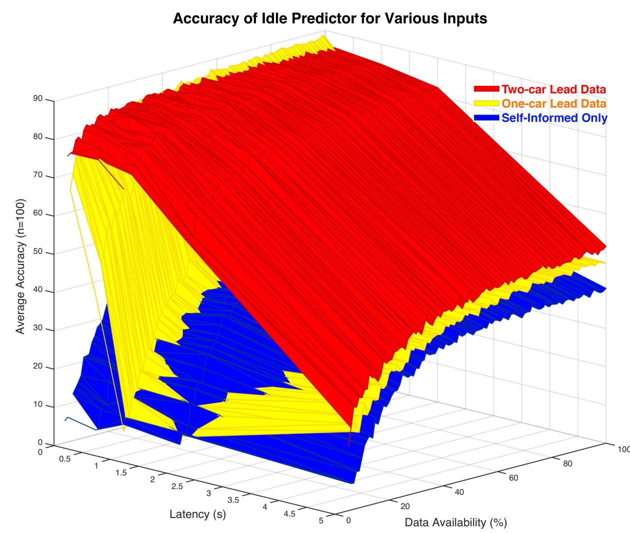

Fig. 5. Accuracy for three simulated idle predictors: self data only, data from one car ahead, data from two leading cars.

C. Other Considerations

Real-world data are dirty, and cost is a real factor. Other design choices may limit the technologies available for a developer's use. Addressing these needs, application accuracy must be evaluated against cost, sensitivity to data availability, and other metrics.

These considerations are explored for the idle predictor in the following sections, where we explore the sensitivity to system changes one element at a time.

1) Cost vs Benefit

While it is good to understand the theoretical capabilities of an application, one must understand the cost/performance relationship as well. Cost is calculated from the model in Section IV-A, and we use accuracy as a surrogate for benefit.

For these evaluations, we generated accuracy performance measures from the simulation possessing location and position data for a vehicle and its two leading cars. The results for systems with current and degraded maps are shown in Figure 6. Degradation and information gain was modeled by randomly eliminating or incorporating a percentage of data tuples to or from a baseline set.

These figures illustrate the cost for our system and demonstrate a) the high cost of hardware for communication and sensing, and b) the cost and benefit of updated maps in improving performance. The top plots demonstrate the cost of the application and all related hardware; the bottom plots show the cost of the application's variable resources. On the left, maps are left to languish while the right-side maps are replaced at a rate beating simulated data atrophy.

One sees that the cost of deploying an application is small relative to the cost of the enabling hardware, and that the cost

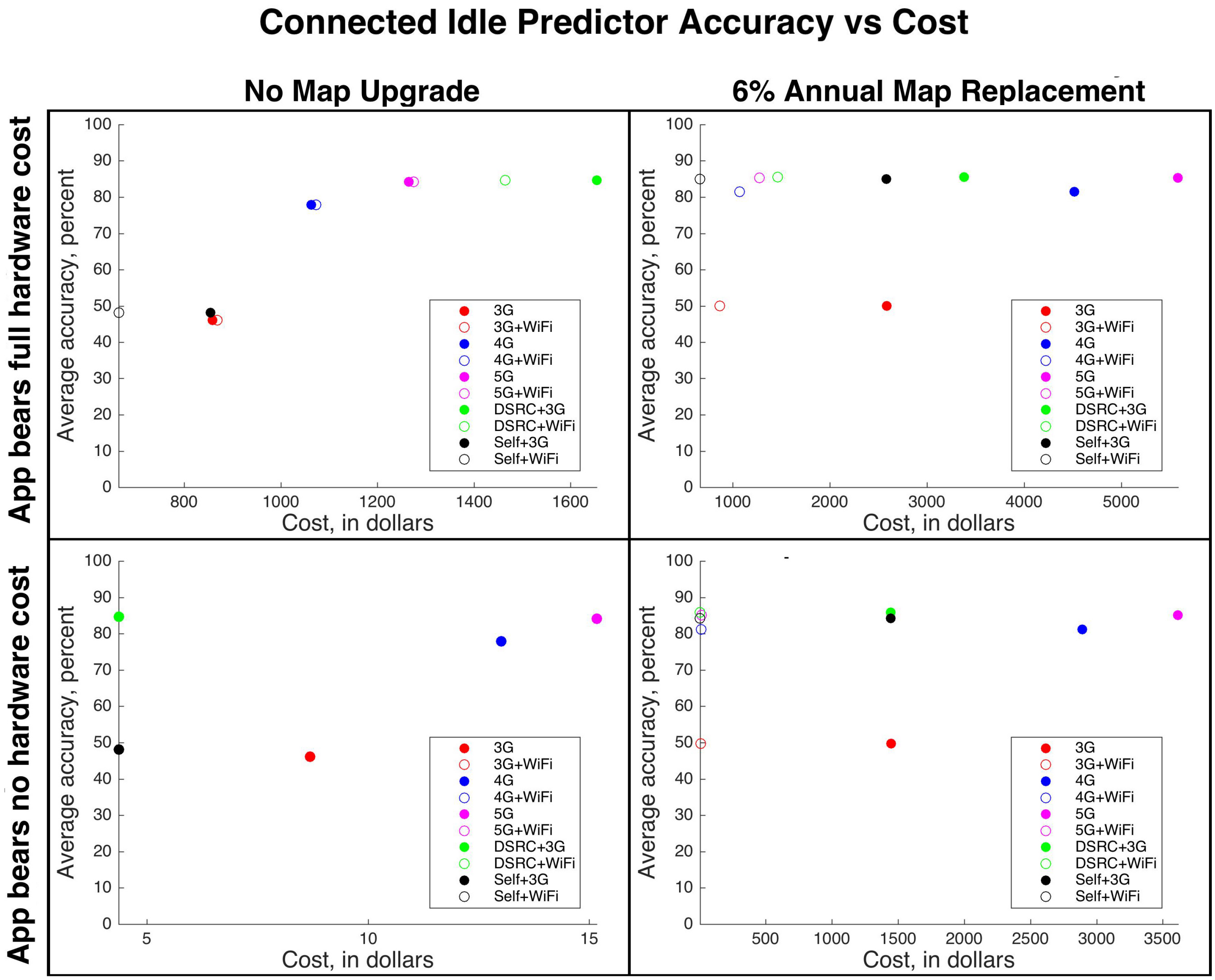

Fig. 6. This figure shows the cost and accuracy of two predictors: one with current maps, and one with a degraded map database. Note that these plots are not comprehensive; for example, DSRC with 3G has a high accuracy and low variable cost but, to attain this accuracy requires the presence of nearby, connected vehicles.

of updating maps cellularly dominates other communication costs. Clearly, WiFi hybridization or other map updating approaches could significantly reduce costs.

2) Sensitivity to Latency

Idle estimation and traffic prediction are time-sensitive applications. High latency is similar to a leading distracted driver missing a traffic light turning green, and could cause a ripple effect of congestion.

Figure 7 shows the performance changes with increased simulated data receipt latency. Sensitivity to latency grows as latency increases, quickly driving the prediction rate below 50% and rendering the application ineffective.

3) Sensitivity to Bandwidth

Beyond map updates, shared sensor data drives bandwidth use. To model the cost of data transmission, we provided each parameter with a fixed packet size and we calculated their update rates based on stop frequency estimates. Each transmission's three-byte packet assumes each input data parameter is 16 bits in resolution, and that the messaging protocol and retransmission adds 50% overhead.

In Figure 8, we see that additional data inputs increase the cost and accuracy of running an application. Depending on the cost/reward function, additional data transmission may be worthwhile.

Bandwidth reduction techniques such as estimator-based mirror may be used to further maximize the data encoded in each byte, reducing bandwidth costs [24].

4) Sensitivity to Data Availability

Whether data come from on-board sensors or a training set, dense data are required to improve classification accuracy. The impact of sparse data depends on several factors including network technology, latency and bandwidth limits.

The knee in Figure 9 indicates that input richness reaches a critical point and then offers diminishing returns.

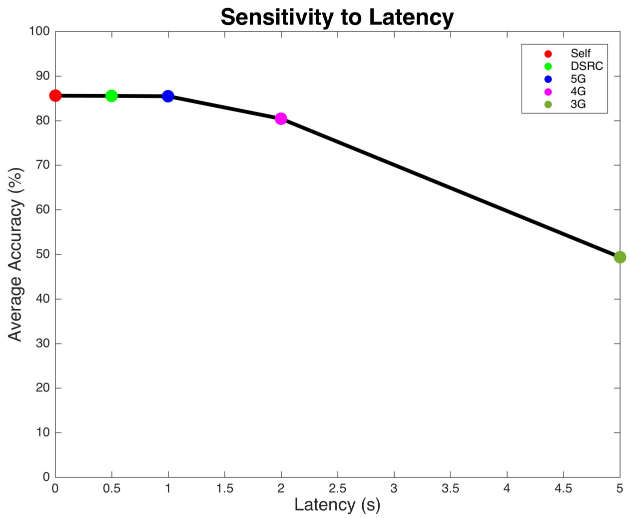

Fig. 7. Accuracy is not impacted at low latency; as latency increases, prediction accuracy falls off sharply. This illustrates that slow wireless technologies are not enabling.

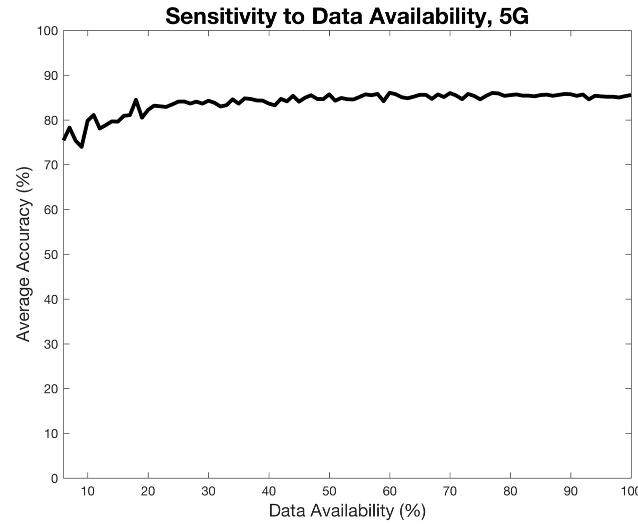

Fig. 9. Prediction accuracy decreases with increased input sparsity.

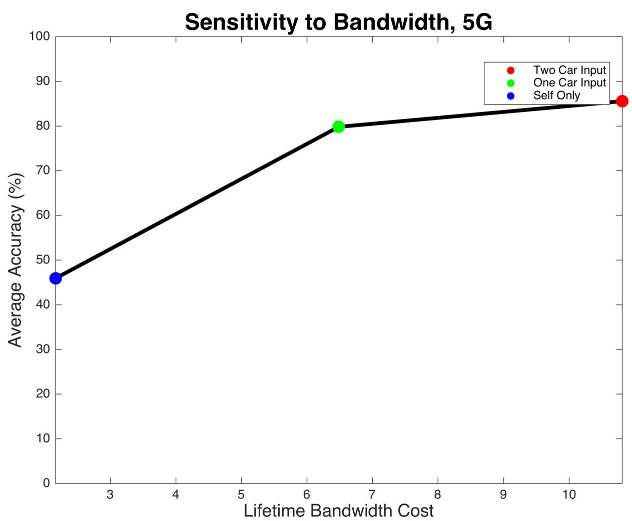

Fig. 8. This plot shows the impact of sensor bandwidth use relative to application accuracy.

5) Sensitivity to Freshness

While local databases work well for applications where reference data are not time-variant, other applications need fresh information. Databases can be replaced or replenished, with new data added and/or a subset of the old data removed.

For predicting idle times, the neighbor database must be kept updated to deal with geospatial changes (road construction), engineering changes (traffic light timing) and social changes (driver behaviors).

We modeled data freshness by linearly varying reference neighbor sparsity. Data loss was held constant, while replenishment rates were varied to show the impact of losing or gaining reference data.

Figure 10 shows that replenishment may keep this application working well for the lifetime of a vehicle depending on rate of atrophy and replacement. If the maps are allowed to degrade by a net 3% year over year, the application will stop functioning before the end of the vehicle's anticipated lifetime. On the other hand, map replenishment of more than 2% annually offers insignificant additional insight and due to the cost is not advisable unless it is achieved without incurring bandwidth cost (via mailed installation media or updates over WiFi).

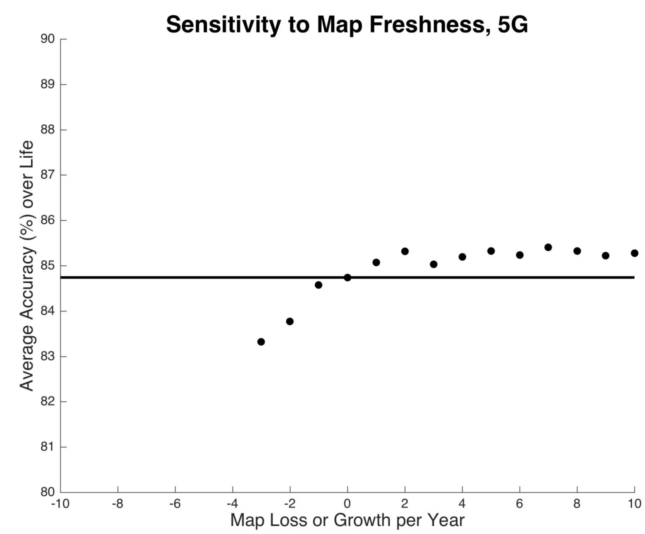

Fig. 10. Lifetime average accuracy for idle time prediction with reference set data loss, stagnation, or growth.

The impact of data freshness must be considered over the vehicle's life, so developers must determine when to stop supporting an application. Figure 11 shows data atrophy over time and how technology choice changes our application's robustness to this decay.

Here, we see that 3G and other "slow" technologies are most sensitive to data loss.

6) Car/Cloud Storage Split versus Bandwidth

The split between in car and remote storage and computation determines our application's bandwidth consumption.

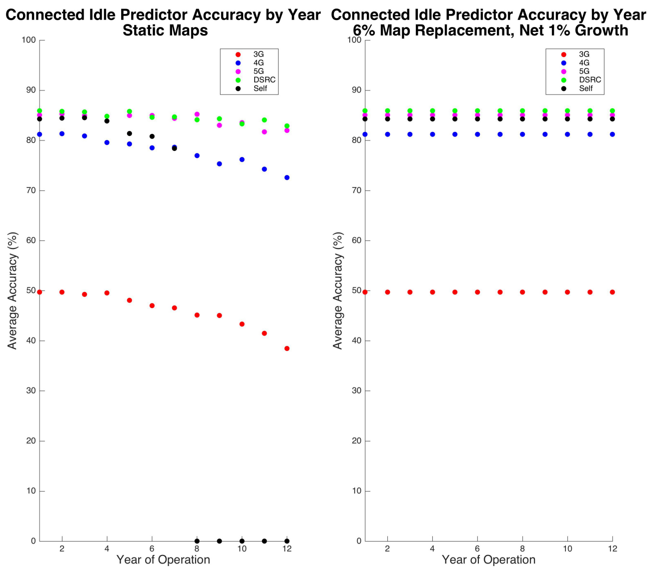

Fig. 11. Databases lose utility over time. Replenishment can keep applications running well for longer, at possibly at significant expense (depending on the chosen connectivity technology).

While local computation offers low latency and improved reliability, remote servers support complex computation and near-limitless reference data. Here, we consider the cost impact of running our estimator locally versus remotely in Figure 12. A similar figure may be created for the cost of computation.

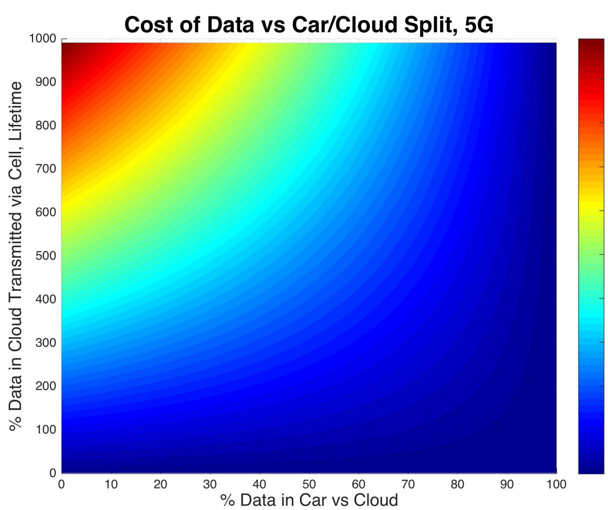

Fig. 12. This figure shows how the cost of bandwidth and data storage depends on the car/Cloud split.

V. Example Value/Tradeoff Calculations

We now examine this application from a simplified profit/loss standpoint. First, we create a sales.

We assume that our example car company sells one million units per year and makes a net profit of $\$2,500$ per car sold. This car company conducts a market survey and determines that if this system works perfectly (100% prediction accuracy), it stands to gain 4% in annual sales volume, while if this system works poorly (0% accuracy), it stands to lose 4% in sales (both these gains and losses ignore network effects and brand reputation changes). With 50% accuracy, sales remain unchanged, and we will assume that this model scales linearly from 0 to 100%. We assume a development cost of $\$5,000,000$ and to simplify the cost/benefit model, assume this cost is invariant for all application embodiments.

Using these simplifying assumptions, we conduct a quick analysis to determine how this company's profitability would change when implementing predictors with differing accuracies and costs. The relative changes in profit for several prediction accuracies are shown in Table V.

One sees that due to the significant $5 million development cost and relatively small 4% sales increase for this feature, the predictor's accuracy must be at least 52.5% to break even. With 100% accuracy, the system generates a net increase in profit of $\$91.35$ per car sold. Therefore, to be cost-effective, the system must cost $\$91.35$ or less over the operational life of the vehicle assuming a constant selling price.

From our models, we see that implementing a predictor is infeasible if we hold the application directly accountable for the cost of radios, sensors, and bandwidth. It is possible, however, that these models may be feasible if we are able to justify the hardware and map update costs as being supported entirely by another value-add application (e.g. traffic-aware navigation).

If we instead assume that the car already has appropriate sensing and communications, we see that some of our simulated solutions can deliver sufficient accuracy at low enough operating cost to be viable. For example, all of the radio-connected solutions that do not require map updates and do not support the hardware cost are feasible and result in a sales increase as well as a net profit relative to business as usual.

Consider the DSRC case as an example. Accuracy is 85% with an ongoing operational cost of $\$0$, resulting in a profit increase of $63.25 per vehicle sold. The 5G case, which yields 90% accuracy, supports an additional $72.67 and costs $15 in ongoing operational fees over the life of the vehicle to implement, resulting in a net profit of $57.67. Both technologies are feasible, though the OEM may select the implementation best representing the company's values (luxury/refinement, cost-consciousness, as examples).

Manufacturers will need to use variations of this simple cost/benefit analysis to determine whether applications are wise investments.

VI. Application and Technology Outlook

We have considered many factors impacting connected vehicle application design and performance.

Examining our sample application through the lens of these considerations illustrates the impact of radio technology choice, data availability, and architecture on estimator accuracy and implementation cost. Though this document explored the design factors for a single application, our approach is extensible and generalizable toward other application types.

First, the application should be evaluated for technical feasibility and candidate technologies identified, using a process

Prediction accuracy (%)	Number of cars sold	Net profit (fleet-wide)	Profit increase/decrease	Allowable cost per car
0.00%	960K	$2,395M	-$105M	-$109.38
10.00%	968K	$2,415M	-$85M	-$87.80
20.00%	976K	$2,435M	-$65M	-$66.60
30.00%	984K	$2,455M	-$45M	-$45.73
40.00%	992K	$2,475M	-$25M	-$25.20
50.00%	1,000K	$2,495M	-$5M	-$5.00
52.50%	*1,002K*	*$2,500M*	*$0.00*	*$0.00*
70.00%	1,016K	$2,535M	$35M	$34.44
85.00%	1,028K	$2,565M	$65M	$63.23
90.00%	1,032K	$2,575M	$75M	$72.67
100.00%	1,040K	$2,595M	$95M	$91.35

TABLE V
THIS TABLE SHOWS THE RELATIVE PROFIT INCREASES FOR INCLUDING THE INTELLIGENT ENGINE STOP/START SYSTEM, BASED ON OUR SIMPLIFIED PROFIT AND SALES MODEL.

to evaluate contemporary technology and determine those capable of supporting the target application, just as we did in examining feasible regions of latency and bandwidth for various technologies. It is important to look at emergent enabling technologies, as the landscape is growing rapidly even today. With technologies such as 5G becoming increasingly convergent with best-available mesh technologies in terms of bandwidth, latency, and cost, developers must increasingly consider this and other nascent technologies.

Upon identifying appropriate technology candidates, the algorithm's hypothesis should be validated and tested with different input sets to determine how different data sources impact algorithm performance and cost.

Finally, a cost model and objective function must be created so that this function may be optimized. Our models were simplified; others may be more complex. For any models, the plots we created showing costs and benefits for differing technology implementations are effective tools to visualize how an application might be implemented.

It is worth noting that formalizing optimality for connected vehicle applications is not possible, because application benefit depends on subjective qualitative value assessments. In the end, the optimal application implementation will often need to strike a balance between local and remote computation. The specifics of how much computation and storage takes place in which location will depend on the application and its value to consumers, the brand image, and the manufacturer.

Following this type of framework will help developers create efficient applications capable of improving vehicle design and use. Importantly, these developers will be able to rationalize their design choices and ensure that applications are feasible and worthwhile.

REFERENCES

[1] M. Barth, G. Scora, and T. Younglove, "Intelligent off-board management of vehicle operating parameters," in *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, vol. 1, Oct 2003, pp. 352–357.

[2] R. W. Cox, "Local area network technology applied to automotive electronic communications," *IEEE Transactions on Industrial Electronics*, vol. IE-32, no. 4, pp. 327–333, Nov 1985.

[3] R. K. Jurgen, "Coming from detroit: Networks on wheels: Multiplexed wiring in new models promises cost-effective control and accurate diagnostics, as well as fewer wires and less redundancy," *IEEE Spectrum*, vol. 23, no. 6, pp. 53–59, Jun 1986.

[4] S. Arthur, H. N. Breed, and C. Schmitt-Luehmann, "Shifting car makeup shakes up OEM status quo: Software strength is critical," 2003. [Online]. Available: https://www-935.ibm.com/services/th/igs/pdf/g510-1692-00-shifting-car-makeup-shakes-up-oem-status-quo.pdf

[5] G. Bilchev, D. Marston, N. Hristov, E. Peytchev, and N. Wall, "Traffimatics — intelligent co-operative vehicle highway systems," in *Intelligent Spaces: The Application of Pervasive ICT*, A. Steventon and S. Wright, Eds. London: Springer London, 2006, pp. 93–108. [Online]. Available: http://dx.doi.org/10.1007/978-1-84628-429-8_7

[6] S. C. Albulescu, A. A. Pucãrea, and O. Dascãlu, "The automotive industry in a new tehnological era," *SEA-Practical Application of Science*, no. 5, pp. 99–106, 2014.

[7] "4G LTE: Cost efficiency infographic," Alcatel-Lucent. [Online]. Available: https://www.alcatel-lucent.com/solutions/4g-lte/cost-efficiency-infographic

[8] Z. Epstein, "As data gets cheaper for verizon to transmit, customers are paying more," 2013. [Online]. Available: http://bgr.com/2013/01/23/verizon-shared-data-plans-analysis-303240/

[9] K. Han, A. Weimerskirch, and K. G. Shin, "Automotive cybersecurity for in-vehicle communication," *IQT Quarterly*, vol. 6, no. 1, 2014. [Online]. Available: https://kabru.eecs.umich.edu/papers/publications/2014/IQT%20Quarterly_Summer%202014_Han%20et%20al.pdf

[10] B. Schoettle and M. Sivak, "A survey of public opinion about autonomous and self-driving vehicles in the U.S., the U.K., and Australia," University of Michigan, Ann Arbor, Transportation Research Institute, Tech. Rep., 2014.

[11] J. E. Siegel, "CloudThink and the Avacar: embedded design to create virtual vehicles for cloud-based informatics, telematics, and infotainment," Ph.D. dissertation, Massachusetts Institute of Technology, 2013. [Online]. Available: https://dspace.mit.edu/handle/1721.1/92230

[12] E. Wilhelm, J. Siegel, S. Mayer, L. Sadamori, S. Dsouza, C.-K. Chau, and S. Sarma, "Cloudthink: a scalable secure platform for mirroring transportation systems in the cloud," *Transport*, vol. 30, no. 3, pp. 320–329, 2015. [Online]. Available: http://dx.doi.org/10.3846/16484142.2015.1079237

[13] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty, "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," in *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, ser. VANET '04. New York, NY, USA: ACM, 2004, pp. 1–9. [Online]. Available: http://doi.acm.org/10.1145/1023875.1023877

[14] L. Xu, L. Y. Wang, G. Yin, and H. Zhang, "Coordinated control and communication for enhanced safety of highway vehicle platoons," in *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, Dec 2013, pp. 43–47.

[15] P. Gomes, C. Olaverri-Monreal, and M. Ferreira, "Making vehicles transparent through V2V video streaming," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 930–938, Jun 2012.

[16] C. H. Lee, C. M. Huang, C. C. Yang, and T. H. Wang, "A cooperative video streaming system over the integrated cellular and DSRC networks," in *2011 IEEE Vehicular Technology Conference (VTC Fall)*. IEEE, Sep 2011, pp. 1–5.

[17] T. ElBatt, S. K. Goel, G. Holland, H. Krishnan, and J. Parikh, "Cooperative collision warning using dedicated short range wireless communications," in *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, ser. VANET '06. New York, NY, USA: ACM, 2006, pp. 1–9. [Online]. Available: http://doi.acm.org/10.1145/1161064.1161066

[18] GSMA Intelligence, "Understanding 5G: Perspectives on future technological advancements in mobile," *GSMA Intelligence Understanding 5G*, pp. 3–15, 2014. [Online]. Available: https://www.gsmaintelligence.com/research/?file=141208-5g.pdf

[19] U. S. E. P. Agency, "EPA decision document: Mercedes-benz off-cycle credits for MYs 2012-2016," United States Environmental Protection Agency, Tech. Rep., Sep 2014. [Online]. Available: http://www3.epa.gov/otaq/regs/ld-hwy/greenhouse/documents/420r14025.pdf

[20] D. Edmunds. (2014) Edmunds. [Online]. Available: http://www.edmunds.com/car-reviews/features/do-stop-start-systems-really-save-fuel.html

[21] D. Erb, "Optimizing hybrid vehicles: Battery pack design, energy management, and collaborative learning," Ph.D. dissertation, Massachusetts Institute of Technology, Jun 2016.

[22] A. Kesting, M. Treiber, and D. Helbing, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4585–4605, 2010.

[23] T. Fesefeldt and S. Műller, "Optimization and comparison of quick and hybrid start," SAE International, Tech. Rep., Apr 2009. [Online]. Available: http://dx.doi.org/10.4271/2009-01-1340

[24] J. Siegel, "Data proxies, the cognitive layer, and application locality: Enablers of cloud-connected vehicles and next-generation internet of things," Ph.D. dissertation, Massachusetts Institute of Technology, Jun 2016.

Josh Siegel is a Research Scientist in the Mechanical Engineering department at the Massachusetts Institute of Technology. He received a bachelor's degree, master's degree, and Ph.D. from the Massachusetts Institute of Technology in 2011, 2013, and 2016, respectively. His research interests include connected vehicles, pervasive sensing, data-informed design and secure and efficient architectures for the Internet of Things. Josh received the Lemelson-MIT National Collegiate Student Prize in 2014.

Dylan Erb is a Research Affiliate in the Mechanical Engineering department at the Massachusetts Institute of Technology. He received a bachelor's degree from the University of Illinois Urbana-Champaign in 2011 and a master's degree and Ph.D. from the Massachusetts Institute of Technology in 2013 and 2016, respectively. He currently works at Ford Motor Company in Electrified Powertrain Engineering.

Sanjay Sarma received the bachelor's degree from the Indian Institute of Technology, the master's degree from Carnegie Mellon University, and the Ph.D. degree from the University of California at Berkeley. He was the Fred Fort Flowers (1941) and Daniel Fort Flowers (1941) Professor of Mechanical Engineering at MIT. He is the first Dean of Digital Learning at MIT. He co-founded the Auto-ID Center at MIT and developed many of the key technologies behind the EPC suite of RFID standards now used worldwide. He was also the Founder and CTO of OATSystems, which was acquired by Checkpoint Systems (NYSE: CKP) in 2008. He serves on the Boards of GS1, EPCglobal, and several startup companies, including Senaya and ESSESS. He also worked with Schlumberger Oilfield Services, Aberdeen, U.K., and the Lawrence Berkeley Laboratories, Berkeley, CA. He has authored over 75 academic papers in computational geometry, sensing, RFID, automation, and CAD, and is a recipient of numerous awards for teaching and research, including the MacVicar Fellowship, the Business Week eBiz Award, and the Informationweek's Innovators and Influencers Award. He advises several national governments and global companies.