Cloud Based Self Driving Cars

[1]Naveen S Yeshodara, [2]Nikhitha Kishore
[1]Software Consultant, [2]Software Engineer
Data Center Management
Novell Software Development Center
Bangalore, India
[1]ynaveen@novell.com, [2]knikhitha@novell.com

[3]Namratha S N
Assistant Professor, Instrumentation Department
B M S College of Engineering
Bangalore, India
[3]namrathans@gmail.com

Abstract—**This paper presents a novel idea for reducing the data storage problems in the self-driving cars. Self-driving cars is a technology that is observed by the modern word with most curiosity. However the vulnerability with the car is the growing data and the approach for handling such huge amount of data growth. This paper proposes a cloud based self-driving car which can optimize the data storage problems in such cars. The idea is to not store any data in the car, rather download everything from the cloud as per the need of the travel. This allows the car to not keep a huge amount of data and rely on a cloud infrastructure for the drive.**

Index Terms—**Self-driving car, public cloud, security, privacy**

I. Introduction

The most coveted project of the recent times had been the Google's Self-Driving cars of course. The ability to travel in a more safe way, reducing the accidents and obeying the traffic rules, all without manual intervention is something quite amazing and that's what the self-driving cars do!!. Google's fleet of robotic Toyota Priuses has now logged more than 190,000 miles (about 300,000 kilometers), driving in city traffic, busy highways, and mountainous roads with only occasional human intervention [1]. The Google's vision of reduced car accidents, fuel consumption and congestion doesn't seem to be quite far.

When the idea of a self-driving car itself poses to be so amazing what is not much researched is where should the huge data that the car generates go and sit. According to the reports the car is expected to generate one gigabyte of data per second [2] which is really huge. Big-data is trying to come up with solutions [2] but we don't have them in a fully read form now. It is true that the car needs roadmaps, code and lot of other data to optimize it's 'without-human' driving.

This paper proposes a solution for this huge data generation problem in the self-driving cars. The idea is to use a cloud infrastructure to store everything that is needed for the car to take its long drive. There is no need to now worry about the huge data generation, the data purging or anything, there'll be a cloud to handle all that. The idea is to have the whole code in the cloud and download it based on the user's usability. The cloud can consist of anything that needs to get the car onto the road: Traffic rules for each region, Road map for auto drive of all locations of that state or country, live traffic data, signal information and all other data which is required by the car for self-driving.

When the driver starts the car the authentication is done and after this he can input the region to where the car has to take him and the exact location. The idea is that only the data required to take the car to the specific location has to be loaded to the car thus avoiding the car to carry a huge amount of data on the go. This idea can be widely used for any kind of update to CAR software, Traffic Rules and can be even used to analyze how much a car has travelled and its wear and tear information.

The whole is based on the simple assumption that there is a persistent network connectivity to the cloud and sufficient storage is available in the car to back the data if it is well known ahead that the car has to drive to a location with limited network availability. This paper is organized into the following sections; Section II will give the overview of the Google's self-driving car, Section III will detail the existing challenges in terms of data growth in the existing system, Section IV suggests a solution to the existing challenges in the current system and Section V will have the usability of this solution. Section VI will finally conclude the paper.

II. Self-Driving Car

This section will brief about how the Google's self-driving car works in roads where not every time things work fine. The very heart of the system is a laser range finder mounted on the roof of the car. This is what generated the 3D view of the road lying ahead the car. After generating the 3D map of its environment the self-driving car combines the view with the various world maps, generate different data models so that finally the car can run without hitting any obstacles, any humans crossing the road violating the traffic rules and driving observing the traffic rules in that country.

There are four sensors on the car: four radars, mounted on the front and rear bumpers, that allow the car to "see" far enough to be able to deal with fast traffic on freeways; a camera, positioned near the rear-view mirror, that detects

*All the views put forward in the paper/publication are purely mine/ours and it has nothing to do with Novell/The Attachmate Group

traffic lights; and a GPS, inertial measurement unit, and wheel encoder, that determine the vehicle's location and keep track of its movements [1].

The car not just uses the GPS technology alone to detect where it is positioned but also relies maps from different terrains so that the data is accurate. Just relying the GPS will not get the co-ordinates accurate [1]. Another significant point is that the car compares the data that it is collecting with the data that was pre-recorded so that the pedestrians can be differentiated from other obstacles like poles, mailboxes etc.

However the car will turn aggressive in situation where other vehicles are not going to reciprocate the same way how it obeys the traffic rules. The current solution for such real world situation is that the car will advance a bit to such disobedient cars to show what it really intends to do.

Figure 1 shows how the on-board computer in a self-driving car sees the pedestrians and other objects in its environment using the laser range finder.

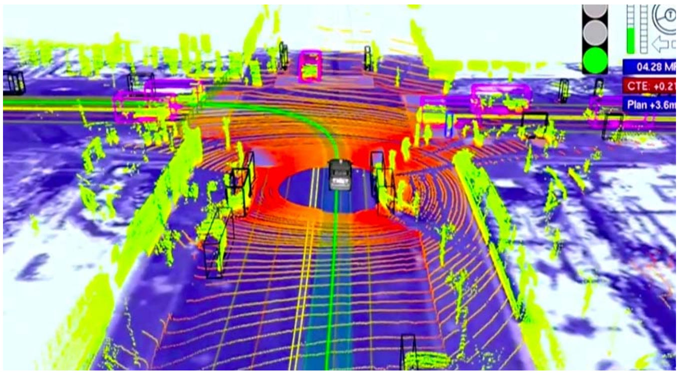

Figure 1: The image generated by the laser range finder in Google's self-driving car

However, the self-driving car cannot be just described as that of Google's. A lot of companies have predicted their own version of the autonomous cars in the new future. A list of official predictions are mentioned below [3]:

- Late 2014, Volvo will feature Adaptive Cruise Control with steer assist which will automatically follow the vehicle ahead in queues.
- Late 2014, The National Telecommunications and Information Administration is expected to set recommendations for setting aside broadband spectrum for autonomous cars.
- By 2015, Audi plans to market vehicles that can autonomously steer, accelerate and brake at lower speeds, such as in traffic jams.
- By 2015, Nissan expects to sell vehicles with autonomous steering, braking, lane guidance, throttle, gear shifting, and, as permitted by law, unoccupied self-parking after passengers exit.
- By Mid-2010's, Toyota plans to roll out near-autonomous vehicles dubbed Automated Highway Driving Assist with Lane Trace Control and Cooperative-adaptive Cruise Control.
- By 2016, Mobileye expects to release fully autonomous car technology.
- January 1, 2017 The National Highway Traffic Safety Administration hopes to mandate the adoption of Vehicle-to-Vehicle technology on all new automobiles.
- By 2017, Tesla plans an "autopilot" feature that handles 90% of miles driven. Tesla has partnered with Mobileye.
- By 2018, Google expects to release their autonomous car technology.
- By 2020, Volvo envisages having cars in which passengers would be immune from injuries.
- By 2020, GM, Mercedes-Benz, Audi, Nissan, BMW and Renault all expect to sell vehicles that can drive themselves at least part of the times.
- By 2025, Daimler and Ford expect autonomous vehicles on the market.
- 2035. IHS Automotive report says will be the year most self-driving vehicles will be operated completely independent from a human occupant's control.

The list underlines how much the idea of self-driving cars are gaining attention in the current scenarios. The autonomous cars are expected to reduce the traffic collisions, reduce the traffic congestion, eliminate the age-constraints for driving etc. Section III discuss in detail the challenges faced by the autonomous cars.

III. Potential Challenges For the Autonomous Cars

While there is a huge buzz going around the coming of self-driving cars the challenges that they pose cannot be ignored. Huge discussions are going around about the liability for damage, loss of privacy, cyber security etc. In this paper, we have focused on one of the challenges that is related to the data storage of the self-driving cars.

According to Mark van Rijmenam, founder of BigData-startups.com, self-driving cars will create 1 gigabyte of data per second. Multiply that by thousands, even millions, of cars, and the amount of data created daily comes to a staggering number [2].

When it comes to data storage this number is huge. To store this amount of data in the car on the drive will be an overhead. Google has already declared that while the car is on the track a set of complex algorithms will set on work to combine the various maps, GPS data and various other sources. Mobileye, which is a Dutch company that specializes in offering inexpensive cameras assisting self-driving cars, has witnessed a business raise to $400 million last year. So, this proves that the trend of self- driving cars is catching up fast and so in coming days we need to get ready to tackle the big data generated from these vehicles [4].

While Big-data is getting ready to provide some solid solutions to tackle this huge data storage problem, this paper proposes an idea which can optimize the amount of data that is used for driving and allow for much better autonomous driving.

IV. LIVE AUTOMATIC SELF-DRIVING CAR

The idea presented in this paper begins with the assumption that the network connectivity available should be 24/7. We have seen the potential advantages and disadvantages that the autonomous driving can introduce. Raw data is the essence behind the concept of a self-driving car. The data from various sources, GPS, maps of various terrains, laser range finder and sensors all pile up the data that is used by the software to turn to information to direct the car through real-life traffic.

In a Live automatic self-driving car, nothing is going to be stored in the car. So there is no fear of that car running and generating data and searching space to store it efficiently. SO where do the data go? Cloud is the solution that we propose. It is a known fact that the IT world is busy in expanding the cloud world to use just what is needed rather than worry about the infrastructure and hardware behind it. Same goes with the autonomous car also. From the perspective of a user, he need not worry about the data that is being dealt while the car is on a drive. The data storage issues should not bother a normal user who is unaware of such terminologies. In the idea that we propose here, every data is stored in the cloud. Whenever the data is required the software downloads it and make use of this. The set of complex algorithms, the pre-recorded GPS data and all that stuff are now going to be inside the cloud and the car downloads it whenever it requires. The only thing that is to be taken care of now is some external storage. This external storage is required because the car may need to locations where the network connectivity might fail. In such cases, the software downloads the area map and such details before the network fails so that it can work offline also.

The working of the proposed 'Live automatic self-driving car' is described below.

In the Google's self-driving car all the data resides inside the car. Switch on the car and the car takes the code, gets the maps ready, combine them with the live images and then start the drive. The only difference here is that all the data is pulled from the cloud, and yes that means the code also. When the driver starts the car, the on-board computer gets connected to the network, whichever is available, be it Wi-Fi, 4G or 3G. This starts the web console and asks for the authentication from the user. The user can sign in with the username and password registered with the cloud service provider. The user can then input in detail the location to where he needs to drive. The software then downloads the code to process the data, takes the pre-recorded data from the cloud that is just necessary to go to that locations, the world maps and start processing. Everything will be still in the control of user as how it is when the data is inside the car. A best advantage is that since nowadays V2V is gaining popularity the clouds can interact and make the self-driving safer.

The software will record the user's habits, where is goes often, where he stops, which are his favorite routes and all that and can use this data if the user wishes so to drive next time. The fact is that most users may need privacy and so the software needs to respect it. So when the driver is on a leisure trip such patterns can be put to use. The next important thing is sometimes the user may need to travel to a destination where the network connectivity is not available. In such cases the user can select a mode in which the software can download the data in advance and store in an external storage in the car. The software checks in intervals about the network connectivity and whenever it seems to have a problem it either asks the user to take up for a manual drive or downloads the data in advance to the external storage. Once the location is reached the software wipes off all the data from the car thus enhancing the security. But in case the driver wishes to keep the data the data will be encrypted so that even if the car is hacked also the data cannot be corrupted.

Figure 2 shows in detail how work flow of the Live automatic self-driving car.

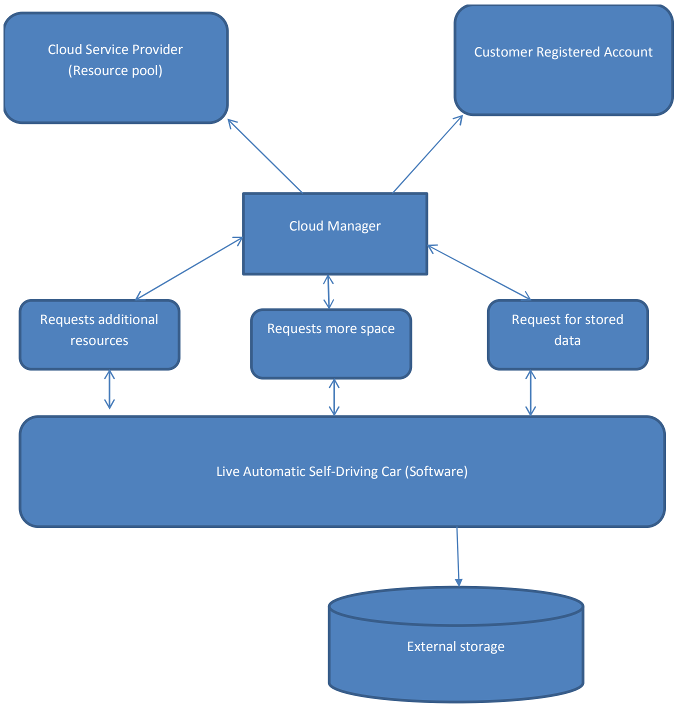

Fig. 1. Flowchart for a Live Automatic Self-driving car

A. The Web Console

When the user gets into the car and once he switch on the car, the web console appears in the on-board computer screen. It asks the user for the credentials with which he has registered with the cloud service provider. Once the authentication is done the web console should ask the user to input the destination to where he has to go. Once the user enters the destination, the web console should show options like whether the user is in an emergency or not. If the user is not in an emergency the web console pop-up will ask whether the user would prefer to use the pre-recorded behavioral patterns to be used while driving. This means that if the driver has stopped by an ice-cream parlor two-three times while going to the same

*All the views put forward in the paper/publication are purely mine/ours and it has nothing to do with Novell/The Attachmate Group

destination earlier the console would alert the user whether it should stop again in this drive a while before reaching the same ice-cream parlor.

The web console is an abstraction of what the self-driving car is all about. The user is not exposed to any of the technical details of how the car is going out in the street all alone. It is a friendly interface with which the user can interact on how to take the car to streets. The console also have other options like internet browsing, the much hyped vehicle to vehicle connectivity options, which not only makes the driving easier but allows the user to be connected with the world even while he is in the car.

Fig. 2. Data to be selected after authentication

B. The Network Selector

The users selections on the web console must now be send to the cloud and this requires a network connectivity. There are a lot of options for getting the car connected to the cloud, either the user can use tethering/hotspot, embedded, public wi-fi, 4G/3G services etc. Since a private Wi-Fi is still not available, the car will have to switch between public Wi-Fi or 3G/4G which is available at the time in a speed that the connectivity remains contiguous and the user need not interfere in the driving.

In such cases what is needed is a network selector. Cisco has already come up with such a client for V2V (Vehicle to Vehicle) solutions [4]. It claims that the Cisco on-board software solution can seamlessly switch between available 3G, 4G and other wireless networks based on cost and quality of service preferences. Such a client is what we call the network selector here. This will always ensure that the car is connected 24/7.

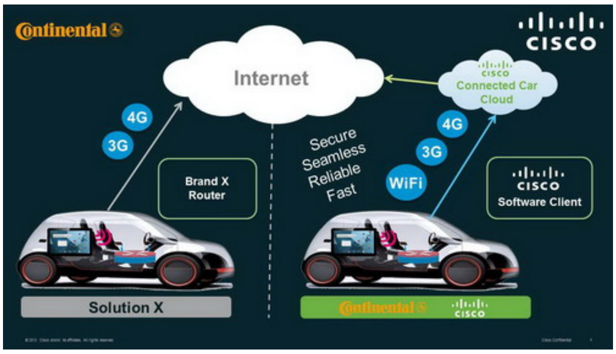

Fig. 3. Cisco software client to switch between available networks [9].

C. Authentication module

The authentication module is required for the car to authenticate the user of the car. This module is started once the car gets connected to the cloud as the user has to be registered with the cloud. The authentication module is not just a piece of software that remains local to the car, but after getting connected to the car it can authenticate other cars and send them to other locations also. This means that if the user wants to send his another car to a location to get is child from school or so, he can authenticate from his car and download the details to the other car also. But for this the requirement is that the other car has to be in the network at the same time.

D. External storage

It is quite normal that the user may decide to change the location to where he had decided to travel. In such cases there may be chances that the newly entered location has troubles with the network connection. This is processed using the self-driving car's on-board computer. In such cases a back-up is required. The software identifies the new location, send the details to the cloud manager and the cloud manager downloads the new data to the external storage. Once the network connection experiences any troubles the software does not fail, rather it uses the data that is now stored inside the car.

E. Encryption module

This module comes into play if the user wants the data to be stored in the car instead of wiping it off once the car is switched off. There may be cases when the driver has to go to the same destination the next day also. For example, the driver goes to office all five days in a week. He may think why to pull the data every day from the cloud. Rather keep the data in the car itself till Friday and wipe it off on Saturday. But the data has to be secured once it is inside the car and therefore it should be encrypted. This means that a hacker even if he gets the data also should not be able to make any use out of this.

The encryption is very important in the cases where the user is trying to drive another user profile. In this case both authentication and encryption has to go hand-in hand. Even if a wrong user can authenticate also the encryption/decryption module has to be strong enough that both the cars can easily acknowledge each other without any misunderstanding.

When a car is self-driving then it is very important how it talks to other cars. There must be a mechanism in which one car has to tell that 'I am Car A and I want your authentication that you are Car B'. Once the authentication is done between both of them the communication should be encrypted. No hacker must be able to easily tap the communication and misguide the cars. For example, if car A tell car B that I will first turn and then you can take the turn and a hacker is misguiding the negotiation like, 'I am car A and you can take turn first', this will mean huge accidents are expected to happen on roads in the near future. A strict cryptographic policy is required for self-driving cars to move safely on the roads.

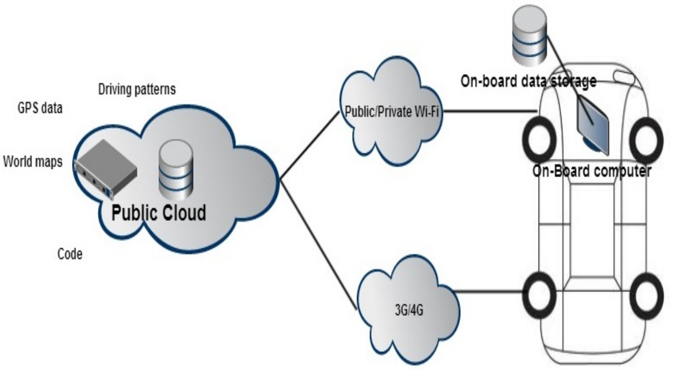

Fig. 4. Network Diagram for the proposed Self-driving car

V. WORKFLOW FOR THE PROPOSED MODEL

In this section, the workflow of the proposed cloud based self-driving car is explained.

1. The user gets into the car and switches on the car. Upon this step the login screen for authentication pops-up on the on-board computer screen.
2. The network connecter in the car will get connected to the internet using 3G/4G/Wi-Fi whichever is the best option available according to the user configurations.
3. Once the car is connected to the internet, the user will be given a number of profiles to select from. He can even login to the profile of his wife's car and send that car to a particular location.
4. The car connects to the cloud and authenticates the user details entered.
5. If the authentication is successful the screen to enter the location details are shown. The user enters the location/destination to where he has to reach.
6. The software pulls the details that are needed to drive to this location, like the code, the GPS data, the pre-recorded data, the user driving patters (if he wishes to), world maps etc to the storage in the car.
7. The car starts driving, the data is combined with the live map got from the laser-ranger finder for more accuracy.
8. In between if the user decides to change the location the software verifies whether the new location have data connectivity in a contiguous manner. Otherwise, the data for self-driving is downloaded to the car's external storage.
9. Once the car reaches a location, the software uploads all the data back to the cloud and get disconnected from the cloud.
10. If the user requires the data to stay in the car the data is encrypted and stored in the external storage.

This workflow is depicted as a diagram in Figure 5.

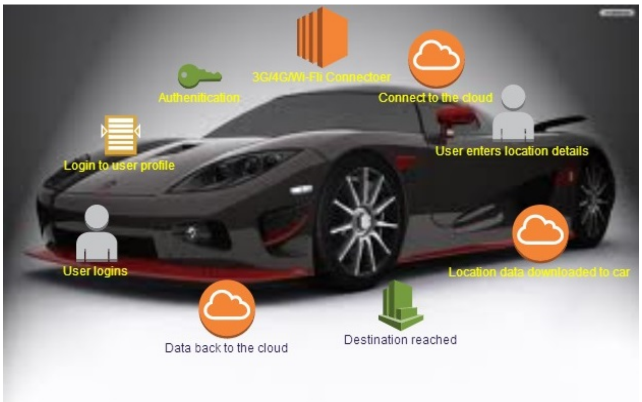

Fig. 5. Workflow Diagram for the proposed Self-driving car

VI. CONNECTIVITY CHALLENGES

For the successful implementation of the proposed idea the network connectivity is a major challenge. The co-operation between the mobile network operators and automotive business is a key player here. However the structure between both the sectors are quite different for them to go hand-in-hand. The biggest gap will be the scope that is defined for both the business. The scope of business for an automotive company is worldwide whereas for a mobile network it is just limited to a particular country. For the same reason, a network that operates in India may not work in another region. So the car company cannot just make deal with a fixed mobile operator or so. Even in a country there may be local broadband network providers and some network provides who go everywhere within the country. Even the workflow for both the businesses are different. The development cycles are normally longer for the car companies and rapid for the network operators. A car may come with a 3G supporting software and the network operator may suddenly upgrade to the 4G support. Such rapid changes in the network industry has to be taken care when the self-driving cars are going to get connected with the cloud.

*All the views put forward in the paper/publication are purely mine/ours and it has nothing to do with Novell/The Attachmate Group

A very good option can be tethering. This means that the user just needs to have a smartphone with internet connectivity and the car can easily use this to get connected to the cloud. Since smartphones can easily switch between networks, it can be a good way to go. But can tethering help when the car is downloading huge amount of code, data, GPS records and all that? This is yet another challenge.

Another idea is the network connector that is now introduced by Cisco. Such a client will take the part of switching between the networks. Thus the changes in the networks can be easily handled and will be no more an issue. Any changes can be accommodated easily by upgrading the client whenever needed rather than changing the whole software provided with the car.

Since we also propose that a car when connected to the cloud can use the user profile of another car to drive it to another place the network connectivity is the heart beat of the proposed idea. The user at any time can authenticate the profile of another car once he know the details and can drive it to a particular location if the car is still in the network. One challenge here maybe how to get the car in the network. It is a huge waste if the user has to always put his car in the network.

VII. Usability and Challenges

In this section we present some use-cases for the proposed system and some challenges. Reducing the data storage trouble is the greatest advantage that the newly proposed system have over other solutions. Rather than optimizing the data generated the cloud-based system just moves the unused data out of the car and keeps it in the cloud. The data is downloaded only on-demand and therefore the amount of optimization done is huge. Some of the real-time use-cases for the proposed system is given below.

1. If driver wants to go to other location which is not selected during the startup, It should get connected to cloud and get the data downloaded some hours before he travels that way or it should inform him to switch to Manual mode.
2. This idea can be widely used for any kind of update to CAR software, Traffic Rules and even use to analyses how much a car has travelled and its wear and tear information
3. This idea can be extended to a level where, it drives to the nearby service center if it feels that there is some issue in the car or contact the nearest service station by itself and ask for help.
4. Also we can avoid traffic jams and even accidents if we know what’s happening 1 KM or some distance ahead.
5. Once the car is stopped, the data is completely wiped out from car so that no one can steal or use the car without permission.
6. If driver wants to drive to same location on next day, he can mention that in application. Now the application will not wipe the data, rather will encrypt it and once driver authenticate it, the data will be decrypted.

Now we evaluate some of the potential challenges faced by the self-driving cars and see how the proposed solution can tackle them.

A. Cyber-security

One of the most potential threats to the autonomous cars is indeed the cyber-security. Most of the data will be inside the car and a hacker can tamper with the data to cause accidents. This a great threat as it can easy for terrorists or robbers to hack the data and cause huge mishaps.

According to the reports many security experts has already warned the security concerns that they have for the self-driving cars. According to most experts when there are self-driving cars that means they need a lot of negotiation between each other. What if this negotiation is not proper, instead of reducing road accidents, it will increase the number [5]. That means there needs to be a way to authenticate each other and convey who should move first or stop first etc. As long as security principles are not applied behind the

In the newly proposed system the data does not reside inside the database at all. Since the car downloads the data that is required only on the go and used up data is completely wiped off after the usage there is now no fear of mishandling of the data from outsiders. Even the data that is kept inside the external storage is also encrypted so that it is not easy for the hacker to quickly tamper the data.

The challenge while going for implementation with the public cloud is indeed the security concern. Even though the hacker does not have the data in the car, the whole data is now available in the cloud. The authentication can be easily broken with a smart level of expertise and thus can easily get the data. Not only that, your data is not now in your hand, it is available in the public cloud. This is a challenge for the proposed system.

Nowadays most of the enterprises are trying to move to the public cloud due to the level of agility that it provides. With this move, most of the public cloud service providers have increased their security levels. It might take much time for a hacker to hack into the entire system rather than trying to hack the data if stored inside the car. So in this way, storage of data in the car leads to vulnerability than when it resides in the cloud.

B. Loss of privacy

This is one factor that may be a curse in disguise of a boon. Most of the autonomous car ideas offer a feature that studies the users driving habits and take actions according to that. For example: if the user always stop by his favorite ice-cream shop when he travels to a particular location, the autonomous car has this feature to ask whether the user wants to stop by the ice-cream corner next time it goes through the same location. This may be pleasing sometimes, but not always. Suppose we late for a meeting and going in the car and the car starts asking you about your favorite restaurant, ice-cream shop etc it can be a real nuisance. How to tackle the misuse of your private data is the biggest question.

In the newly proposed idea, all the data is residing in the cloud, including the users' private data. When going for an important meeting or any urgent situation the user can prevent the download of his private data so that the car straight away takes him to the location that waste time in analyzing the behavioral patterns of the user. This allows less data to be downloaded into the system saving more space. This portion of the data is now with the car, but also untouched which is even better.

C. Failures in network connectivity

The heart of the entire proposed system is in fact the network connectivity. If there is no network connectivity fails there should be ways in which such a situation can be handled peacefully. The switching between the networks need also to be very smoothly that it never affects the driving.

The car may be running in a public Wi-Fi connection and may enter a zone where Wi-Fi signals are not available. In such cases it is very necessary that the car immediately switched to a mobile network like 3G/4G. This hand-over needs to be taken very smoothly. Even a small second of failure in the road can cause a huge loss.

Imagine an autonomous car user deciding to go a place where he has not been before. In such cases if the new place does not have a persisting network connection and the user does not know how to drive it may be the ultimate challenge. In the newly proposed solution when the user enters the location details the software collects all the network details. Once it finds out that the area has problem with connectivity it downloads all the sufficient data to the external storage. In this case once the network fails also the car can drive without a driver with the code it had downloaded before.

As for the smoother hand-over between different networks the Cisco has introduced a new cloud client as discussed in the earlier section. In the proposed system we have included such a network selector than can smoothly conduct the transition between the networks so that the driving can be accurate and safe.

VIII. CONCLUSION

This paper proposed a solution for the data storage troubles that are encountered in a self-driving car. With many companies announcing dates for the official release of their self-driving cars, it has to be acknowledged as the technology of the future. Even though the challenges in building such a car is many the big question is where to store the huge amount of data that it generates. This paper suggests a cloud based self-driving car in which the data is completely taken away from the car. Everything is now stored in the cloud. The software talks with the cloud manager and gets the data that is required to go to a particular location. Only the details for that location is downloaded into the car. All the remaining data stays with the cloud. An external storage is also required in cases when the car has to travel to locations where the network connectivity is not contiguous.

REFERENCES

[1] 'How Google's self-driving car works', *IEEE Spectrum blog.*

[2] 'Self-driving cars to create IGB of data per second', *FierceBigData,* Pam Baker, July 22, 2013

[3] http://en.wikipedia.org/wiki/Self-driving_car

[4] 'Connected car industry report 2013', *Telefonica.*

[5] Siva R.K Narla, 'The Evolution of Connected Vehicle Technology: From Smart Drivers to Smart Cars to… Self-Driving Cars', *ITE Journal,* July 2013.

[6] 'Connected cars: Business Model Innovation', *GSMA Connected Living Programme: mAutomotive.*

[7] 'Vehicle Information Exchange Needs for Mobility Applications: Version 2.0', U.S Department of transportation, Research and Innovative Technology administration.

[8] 'Exploring the connected car', *Cognizant 20-20 insights,* Nov 2012.

.