Social Vehicle Navigation: Integrating Shared Driving Experience into Vehicle Navigation

Wenjie Sha, Daehan Kwak, Badri Nath, Liviu Iftode
Department of Computer Science
Rutgers University
Piscataway, NJ 08854-8091
{wenjie, kwakno1, badri, iftode}@cs.rutgers.edu

ABSTRACT

In this paper, we propose a Social Vehicle Navigation system that integrates driver-provided information into a vehicle navigation system in order to calculate personalized routing. Our approach allows drivers registered into certain vehicle social network groups to share driving experiences with other drivers using *voice tweets*. These tweets are automatically aggregated into *tweet digests* for each social group based on location and destination. While listening to the tweet digests, a driver can instruct the social navigator to avoid or choose certain road segments in order to calculate a personalized route. We present our initial design along with a simple prototype implemented for the Android platform.

Categories and Subject Descriptors

H.4.m [Information Systems Applications]: Miscellaneous; K.4.m [Computers and Society]: Miscellaneous

General Terms

Design, Human Factors

Keywords

Social networks, vehicular networks, navigation systems, human-computer interaction.

1. INTRODUCTION

With the ever-expanding affordability of cars throughout the world, traffic congestion is a severe problem which can have a negative impact on the economy, human health and safety, environment, and human productivity. According to a recent study, TomTom published its 2012 1st-quarter congestion index, using six trillion data measurements of real-time data from vehicles. They calculated that Los Angeles drivers spent 33% more commute time when traffic was freely flowing and 77% more time during rush hours [1]. Thus, traffic congestion is a root cause of significant productivity loss, has an adverse effect on an individual's human well-being and energy, and causes vast economic loss.

ACM HotMobile'13, February 26-27, 2013, Jekyll Island, Georgia, USA.

There are a myriad of ongoing attempts to alleviate traffic jams, by using on-ramp flow meters, determining real-time traffic flow using cameras, better radio traffic reporting, improving the road infrastructure, using electronic informational displays along the roadway, etc. In addition, the navigation system manufacturers are integrating traffic-related specialized functions into on-board vehicle navigation systems. In-vehicle navigation systems are predicted to quadruple in North America by 2019, growing to about 13 million [2]. Apart from the traditional navigation manufacturers, such as TomTom or Garmin, drivers are moving to less expensive smartphone navigation apps, such as Google Maps and Apple Maps [3]. Vehicles equipped with these types of devices are expected to be an integral part of the Internet in the near future [4].

Today, most navigation systems and traffic apps can calculate the best route taking into account real-time traffic flow data, as well as historic data to predict traffic flow. For example, Google Maps calculates the current traffic condition using both real-time data from anonymous GPS-enabled device users and historic traffic data to provide optimal routes. Despite this, users may still not have the choice of the route should they have had access to more information, such as the road condition, or the actual reason and status causing the traffic jams. In many cases, such information cannot be determined automatically, while such information can be provided by other drivers driving ahead. For instance, with the information that an accident on a certain road is almost cleared, a driver can choose to stay on that road, even if current traffic is slow, as opposed to another instance where the cause of a traffic jam is a long term lane closure. Just knowing what is happening on the road ahead in a timely fashion can often alleviate stress and significantly improve the driving experience. In the future, driving autonomous or semi-autonomous cars [5,6] will likely make such information exchange among drivers even more desirable and safe at the same time.

In this paper, we propose *social navigation*, which integrates driver-provided information into a vehicle navigation system to calculate personalized routes. Our approach allows drivers to share driving experiences with other drivers using *voice tweets*. These tweets are automatically aggregated into *tweet digests* for each social group based on location and destination. Finally, while listening to the tweet digests, the driver can instruct the social navigator to avoid or choose certain road segments when calculating the route.

The outline of the paper is as follows: Section 2 provides an overview of Vehicular Social Networks. In Section 3, we motivate the reader with a sample scenario of integrating information provided by other drivers. In Section 4, we present our Social Vehicle Navigation design. In Section 5, we describe our implementation for the Android platform and in Section 6 we discuss related work. Finally, Section 7 and 8 concludes with a discussion and plan for future work.

2. BACKGROUND

Ever since the invention of the first automobile, the power of *mobility* has improved the frequency at which people meet to maintain social relations over even greater distances. In the past, the automobile has been an effective tool for socialization. However, today, as *social networks* have become an essential part of our lives, people who have a common interest can easily form virtual social relations, without having to drive to meet one another. Social networks such as Facebook and Twitter have redesigned how people socially connect with their family and friends with no restrictions to frequency and location. Nevertheless, trends to integrate vehicle and social networking are in development and have garnered increasing attention in recent years [4,8].

Traditional social networking services allow people who share interests to form virtual social communities. Mobile Social Networks [8], on the other hand, take into account the physical location and temporal connectivity. In [7], we proposed to integrate vehicular networks with social networking, calling the result *Vehicular Social Networks* (VSN). In VSN, users can opportunistically form periodic virtual communities based on their interest and commuting patterns.

This paper discusses the integration of vehicular social networks into navigation systems taking into account the *shared driving experiences* and *driver preferences*. VSNs allow drivers to form social groups based on their daily commute patterns. Drivers use these ad hoc social groups to post voice tweets whenever they experience unusual road conditions. The collected voice tweets are then converted into digests and are provided to other users in the same VSN group. Based on this shared information, the driver can input a route preference into their navigation system.

3. MODEL

Consider the example of a road layout as shown in Figure 1:

John commutes to work every day. He has the choice of Route 22 or Route 66 towards his destination. John would like to consider safety first when choosing a route, but unfortunately existing navigators do not provide such information. So, John switches to a *social navigator*, which allows him to benefit from the safety recommendations provided by other drivers. John's social navigator joins two VSN groups, one defined for the route 66 and the other one for route 22.

Lucy, driving on route 66 is experiencing busy traffic due to an accident ahead of her and she shares this information by posting a voice tweet (T1). Around the same time, Sam posts a voice tweet (T2) mentioning that the bridge on route 22 is slippery but luckily, there is not much traffic. Other drivers in the VSN group have already posted tweets (T3-T6) about the traffic accident prior to Lucy's tweet. The server realizes that T2-T6 are tweets on the same traffic accident, so it discards the older tweets meanwhile retaining T2, the latest one. The server aggregates T1 and T2 into tweet digests and sends them to every driver in the group. When the tweet digests are played back, John knows about the conditions on both Route 66 and Route 22. John decides to take route 66 even though the traffic is slow. John makes the decision despite route 22 not having much traffic because he prefers a safe, albeit slow traffic. John tells his social navigator "avoid" after listening to the tweet T2, and "choose" after listening to the tweet T1. Based on these preferences, the social vehicle navigation system recalculates the route for John. Had John's navigation system computed the route simply based on real-time traffic data, he would likely have taken route 22.

Such real-time information sharing service can be made possible using vehicular social network (VSN). Users can join VSN groups that are of interest, and can either post or listen to other users' real-time voice tweets about the traffic. Then, based on the user's perception of the traffic situation, the social navigator can avoid or choose certain routes. To demonstrate its feasibility, we present the design and mechanism of *NaviTweet*, a social vehicle navigator that allows drivers to join a VSN group, post or listen to traffic related voice tweets and consequently include the driver's preference into the navigator's route calculation.

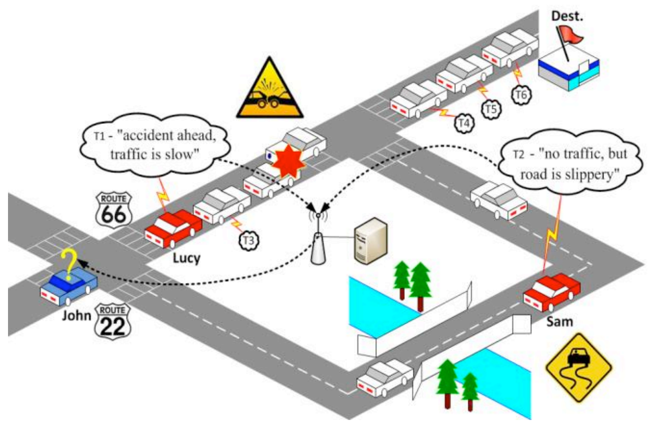


Figure 1 Example Scenario

4. *NaviTweet* SOCIAL NAVIGATOR

A typical workflow of the navigator is depicted in Figure 2.

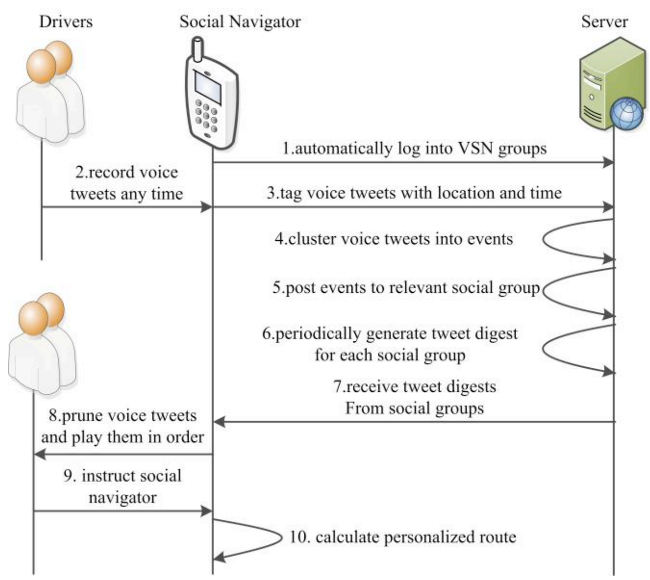


Figure 2. The interaction among the driver, the social navigator and the server

The following steps are performed as part of the workflow:

1. The social navigator automatically logs the driver into the previously registered VSN groups, based on location or destination.
2. The driver records a voice tweet anytime they experience a situation potentially relevant to other drivers.
3. The social navigator tags the voice tweets with the vehicle's location, speed, current time, and driver id, and sends it to the server.
4. The server clusters voice tweets with similar locations and times into events.
5. The server posts events to the relevant social groups.
6. The server periodically generates a tweet digest for each social group out of the most recent voice tweets of each cluster.
7. The navigator receives the tweet digests for the social groups the driver is logged into.
8. The navigator further prunes the voice tweets based on their relevance using criteria such as the vehicle's current location, trajectory and driver's preference for certain group members, then plays them to the driver in the increasing order of distance.
9. After each voice tweet is played, the driver can instruct the social navigator to avoid, choose or ignore it.
10. The social navigator uses this information to automatically calculate a personalized route for the driver.

Each step will be detailed in the following subsections.

4.1 Register and join VSN groups

A vehicular social network (VSN) is a social network of drivers who travel the same set of roadways or have the same destination. NaviTweet uses VSN groups for sharing similar driving experiences. In NaviTweet, we extend the semantics of VSN group definition as follows:

1. Destination group. For example, *group for Manhattan*, or *group for JFK airport*. is characterized by *(i) group name, (ii) center location, (iii) and radius.*
2. Road segment group. Drivers who travel on a major road segment should register the group profiled by the road segment. i.e. *group for NJ turnpike from exit 9 – exit 12 southbound*, or *group for Verrazano-Narrows Bridge eastbound.* Road segment group is characterized by *(i) group name, (ii) road name, (iii) start intersection, (iv) and end intersection*

NaviTweet allows drivers to create VSN groups and manage their memberships through a web portal. We assume that drivers are interested only in the tweets of the VSN groups they have joined. There can be many policies for joining groups. In this work, NaviTweet automatically joins a user to all the VSN groups for which they are registered. Therefore, only those voice tweets will be visible to the NaviTweet client. Further clustering, pruning and sorting on these voice tweets are explained in the following sections. Policies on how to join VSN groups is an interesting issue that we plan to study in the near future.

4.2 Post voice tweet

In our implementation, a voice tweet is limited to 15 seconds. When recorded, a voice tweet is automatically tagged with location, time, speed and driver id. In the current implementation, the driver must touch the screen to begin and end a voice tweet, although a voice-driven interface can also be used.

A voice tweet is posted to the server immediately through a wireless network as soon as the driver finishes recording. Figure 3 shows a typical scenario when many users post voice tweets and download tweet digests every few minutes.

By default, the system expects the tweet to be recorded as close as possible to the event it is referring to in order to allow automatic location tagging. When this is not possible (the driver is busy listening to music, on a call, talking to another passenger, etc.), the location must be explicitly provided. This can be achieved using voice recognition technology or other dedicated user interfaces conducive to drivers.

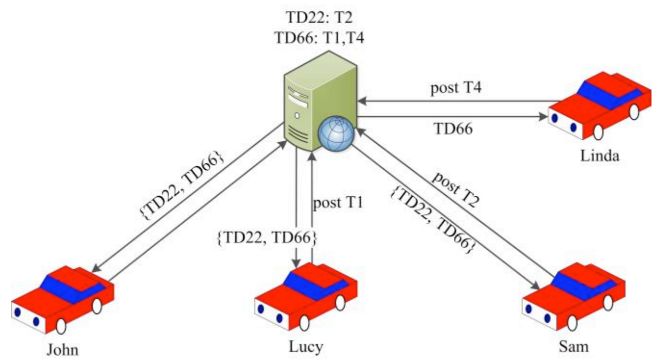


Figure 3. Post voice tweets and receive tweet digests. TD abbreviates for tweet digest. TD22 is the tweet digest on Route 22, and contains voice tweet T2. TD66 contains voice tweets T1 and T4. John has automatically joined groups for Route 22 and Route 66, so he will receive TD22 and TD66.

4.3 Prepare tweet digest

The NaviTweet server periodically prepares a tweet digest for each VSN group. A tweet digest contains selected voice tweets that are recorded near *a destination* or on *a road segment*. For example, in Figure 3, the tweet digest for group 66 only contains selected voice tweets that are published on route 66.

When there is a notable event on the road, multiple users will tend to record voice tweets on that event. This provides a potential opportunity for the server to identify those events by clustering the voice tweets within a short period of time by location. The idea is illustrated in Figure 4. The NaviTweet server clusters all the voice tweets on each road using *DBSCAN* in a two-dimensional space, i.e., distance and time, to identify possible traffic events that triggers the tweeting. In our prototype, we set the minimum tweets per cluster to two to filter out noise and unnoteworthy events. Another important property of traffic events is that they are transient. As the situation evolves, the older voice tweets reflecting past stages of the event will get invalidated. The latest state of the event is likely to be the most relevant one for the drivers. Therefore, NaviTweet packages the voice tweets with the most updated timestamp in each cluster into the corresponding tweet digest for the group. For example, in Figure 4, T1, T4, and T8 are the latest voice tweets for respective events and form tweet digest 66, because they are all on route 66. A NaviTweet client receives the tweet digest but only plays the voice tweets that are on the user's possible routes to the destination. Therefore in

Figure 4, John plays T1, T4 and T8, whereas the other driver only plays T4 and T8. We will further explain the generation of tweet digests on NaviTweet clients in section 4.4. On the other hand, *a noise tweet*, which is a singular tweet that has no neighbor, will never be selected into a digest.

After clustering, selected voice tweets are assigned to the corresponding groups based on their location. NaviTweet server maintains all tweet digests in a hashtable using *groupid* as the key and the set of voice tweets as the value. NaviTweet server updates the tweet digests every five minutes.

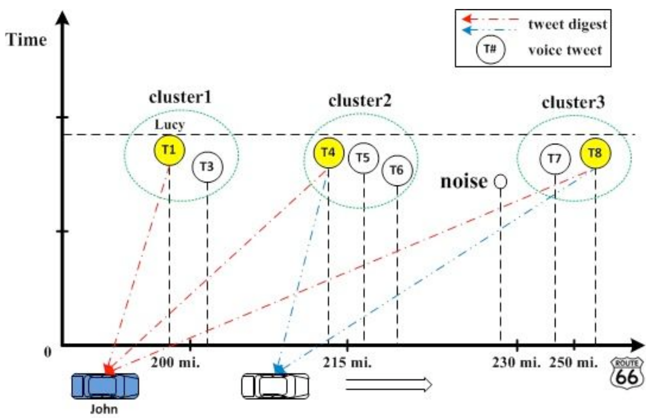


Figure 4. Cluster voice tweets referring to the same traffic event by location and time. T1, T4 and T8 form the tweet digest for the group 66, because they are all located on Route 66. NaviTweet client further prunes the voice tweets in the digests and plays only the relevant voice tweets. The pruning process is explained in section 4.4.

4.4 Play tweet digest

NaviTweet client periodically downloads all tweet digests for the groups the user has automatically joined. The NaviTweet client combines the voice tweets in all digests into a temporary set by excluding duplicate tweets, and matching the set with all possible routes to discard the tweets that do not reside on any of the alternative paths to the destination. After this pruning pass, there will be fewer tweets left in the final *candidate set*.

The calculation of alternative routes is performed using A* search algorithm. Whenever we expand the possible route segment set into an open list, we check if the new segment contains any voice tweets. If so, we attach that segment to the voice tweet meta-data. In the end, NaviTweet client sorts the final *candidate set* in increasing order of a tweet's distance from the current location and selects the top N voice tweets (four in our implementation), which form the local digest that is played.

When playing the local digest to the user, it is important to minimize the cognitive load on the driver. Because drivers are more interested in real-time data, we sort the voice tweets in the local digest in the increasing order of distance to the user's current location. We believe it is easier for drivers to process and understand the information in this manner.

Having generated the local digest to be played, NaviTweet client will download the voice tweet audio data from the server and will play each tweet of the digest in that order. The *local digest* is played in the following format.

```
t1: <road name, audio content>;
t2: <road name, audio content>;
                ……
tn: <road name, audio content>;
```

We affix the road name to the voice tweet. In this way, users can figure out which road each voice tweet relates to. We assume users are familiar with the roads they registered for in the VSN.

A new digest is generated and played to the user automatically every few minutes unless she has stopped the navigator. The driver can set the time interval based on how close they want to watch the road events.

4.5 Select and calculate route

As mentioned in Section 3, the criteria for a good route can vary for different drivers on different trips. Therefore, it is crucial to integrate some extent of personalization to the routing engine. NaviTweet client allows a driver to instruct the navigator which road segments to avoid or choose based on what they have heard in the local digest.

After a digest is played, the user will be prompted to compose a decision in the form of "avoid route 1 and 3; choose route 4". Here, route i refers to the road segment in the i^{th} voice tweet. When a user says "avoid route i", road segment i will be penalized by the routing engine, and thus, becomes highly unlikely to be included in the final route. Contrary to this, when a user says "choose route i", then the NaviTweet client will select the destination via road segment i.

Safety is one of the most important design aspects, so we use speech-to-text recognition for inputting driver's decisions. When a user's command is recognized as irregular input or a user does not provide feedback, NaviTweet client simply ignores the input.

A digest is played and the user decision is considered at the beginning of each interval. There is a slight difference between the first run and later runs. For the first run, after the user's decision is provided, a new route is calculated and rendered; whereas for the later runs, rerouting and re-rendering only applies when the user tries to avoid some segment on the previous route result or chooses some segment that is not contained in the previous result.

4.6 Scalability

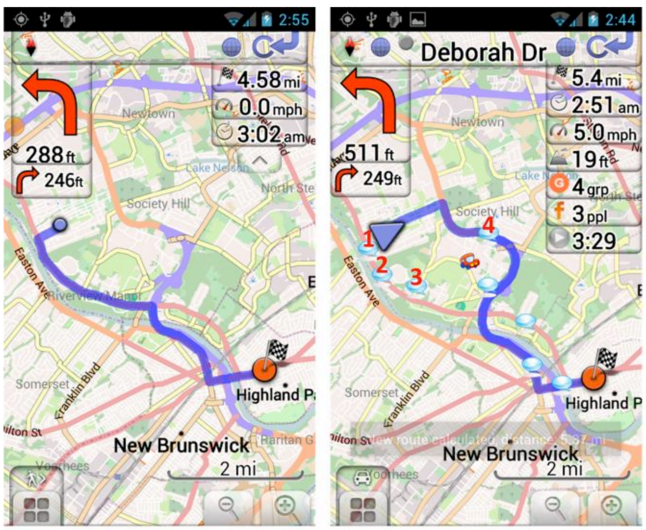
We mention two design issues where scalability comes into play.

1. Tweet Digest. NaviTweet server can easily become bottlenecked if a local digest is generated on the server for each driver. Therefore, we chose to create tweet digest for each VSN group and leave the final selection of the voice tweets to be played to the client. This greatly reduces the number of tweets a client needs to prune and lowers the load of the server as well.
2. Tweet meta-data. The metadata contains a messageid and a timestamp of the voice tweet and is stored in an RDBMS, whereas the voice tweet audio data is stored as a file on a file system. Having generated the final local digest, NaviTweet client retrieves the audio file of each tweet by opening an input stream on the URL locally calculated from the messageid and timestamp of the tweet. Storing the audio data as files organized by timestamp makes archival and clean up much easier, knowing the fact that voice tweets are transient resources.

5. IMPLEMENTATION

We elaborate our work on a freely available open-source GPS navigator, OsmAnd. As a result, our code base only accounts for around 3000 lines of Java code and 2000 lines of PHP, much less than that expected if everything is built from scratch.

NaviTweet server runs on one Amazon EC2 instance using Ubuntu 12.04 LTS. It stores all the metadata in MySQL 5.5, and keeps the voice tweet files on local disk. NaviTweet client runs on Android 2.2+. We use Google Map for VSN group management, and Android speech-to-text recognition and text-to-speech synthesizing library to quickly implement the features that we want. We use OpenStreetMap as our map data to do the pruning and routing. Beyond this, the availability of the source code of OsmAnd and maps was instrumental for implementing the service.



(a) Naïve shortest-path routing from A to B using OsmAnd. The blue dot is the user's current location, and the flag is the destination.

(b) Personalized routing using NaviTweet. The blue bubbles with number are the voice tweets. The red car icon stands for friends' location. User said "avoid route 1, 2, 3; choose 4".
T1: "a dead dear on the road side"
T2: "the traffic is dead"
T3: "a broken tree blocks the road"
T4: "they have Chinese food festival here"

Figure 5. Comparison of traditional routing with NaviTweet

When posting voice tweets and downloading tweet digests to and from the server, there is a network usage cost. Because we limit the length of each tweet message to 15 seconds, an audio message usually requires about 5 to 8 Kbytes using 3GP compression. By default, the digest interval is set at 15 minutes and 4 tweets are played for each digest, so if a user makes 20 voice tweets every day, they will use approximately 6.7 Mbyte every month, which is a reasonable amount of traffic these days.

As a prototype and a first step toward integrating shared driving experiences into the routing engine, NaviTweet provides a smarter way to navigate. Figure 5 demonstrates a real life example of how NaviTweet can help drivers find a better route. We plan to do more sophisticated user studies in the near future.

In the current implementation, to record a voice tweet when a user observes a road event, the user would wave their hand over the screen to trigger the proximity sensor. A microphone image will immediately cover the screen to prompt recording. The user simply records a voice tweet by speaking. The user finishes their recording by touching any part of the navigation interface screen. If the recording exceeds 15 seconds, it ends automatically. NaviTweet takes care of sending the voice tweet to the server automatically in the background. At fifteen minute intervals, a NaviTweet client automatically generates a new tweet digest containing four tweets and plays them in distance-ascending order to the user. Rerouting starts based on the feedback the user gives after hearing the tweet digest.

6. RELATED WORK

The traditional online social networking services such as Facebook, LinkedIn and Twitter focus essentially on providing a foundation of social relations among users who have a common interest without restrictions to where the user is located. Twitter, a combination of an online social networking and microblogging service, allows users to post up to a 140 character text-based message called "tweet" so that the user can join a group to follow a conversation, opinion, story, idea, news or whatever interests the user. Recently, automobiles are integrating social networking services. Developed by Ford, the Twittermobile [9] is a car that can send and receive Twitter messages. Twitter messages sent out by the car can be any type of information ranging from just the driver's mood to informative real-time traffic notices. Also, features such as automatic check-ins via Foursquare or Facebook apps are included. Toyota [10] has also worked on integrating short message social media into the vehicle's dashboard. Both manufacturers are trying to integrate short message social services with cars, yet both are text-based messages. On top of that, the Toyota's version is a predefined template type, for example, "I am going to [destination] and arriving at [time]," where message types have limitations.

In [7], we presented a framework for VSN where people who are physically adjacent to each other construct a periodic virtual social relation. This is an integration of social and vehicular networks whose goal is to virtually build a community for commuters. We built RoadSpeak, a voice chatting system over vehicular social networks, which can be used by daily driving commuters or a group of people who are on a commuter bus or train. NaviTweet, in a similar way, is used to post or listen to traffic related voice tweets, so that the driver's preferences can be incorporated into the navigator's route calculation.

The award winning app for Ford apps competition was presented to students at the University of Michigan. The app is called Caravan Track [11] and has been designed to allow drivers to share vehicle and route information among a group of cars. The idea was derived from the Citizens Band Radio (CB Radio) [12], a short-range radio communications system, where traditionally truck drivers used to locally communicate amongst themselves on topics such as traffic problems, route directions or any other relevant matter. Caravan Track is known also as a tweeting car, allowing members of the group to track one another's specific entities such as location, speed and direction. Route alert functions based on incidents on the road were also applied. The limitation for this work was also the predefined message types where the app had a multiple-choice interface to eliminate typing for safety purposes.

Waze [13] is another popular navigation app that uses crowd-sourcing to provide real-time routing and traffic information along with functions to improve and edit the map itself. Here, social networks are used to send predefined push button messages stating incidents like the degree of traffic, police speed traps or accidents. Chatting functions similar to RoadSpeak [7] are also available (called ChitChat). Waze incorporates social feed; however, the feed is used in the calculation of the best route and does not accommodate human preference factors to the route selection.

7. DISCUSSION AND FUTURE WORK

There are many issues that require additional research. Building the right user-interface to enable drivers to interact with the social navigator, while driving, is a non-trivial task. Issues such as driver behavior, safety and cognitive load have to be further explored through a systematic user study.

Like any other social feeds, for the system to work, a suitable amount of users are necessary. Incentives for tweeters such as "likes" or points are used in existing apps such as Waze [13] so that many users contribute tweets for the system to work. Such mechanisms can also be integrated in our implementation to properly incentivize tweets. For instance, acquired points (or reputation) of a driver can be used to give priority to their voice tweets in the selection for tweet digests.

Selecting the most relevant tweets to be included in the tweet digest can be particularly difficult when the number of tweets is large. We plan to explore various approaches to achieve tweet selection either automatically, using additional criteria such as user reputation, or semi-automatically, by crowdsourcing this task to people willing to help in real time. Drivers' feedback can also help to eliminate improper or malicious tweets. For instance, when tweets are in large number, multiple tweet digests can be initially created for the same group and distributed to different drivers in order to collect feedback to select the quality tweets.

Finally, security, privacy, malicious users and last but not least passenger safety [14] must also be considered. As future work, we plan to evaluate our NaviTweet on roadways and extend the social voice tweets to link sensor networks (detect environmental pollution) with social networks.

8. CONCLUSION

This paper introduced the social vehicular navigation system that uses driver-provided traffic related voice tweets, an improvement over the current navigation systems that do not have such a feature. Many of the newer navigators do apply real-time traffic data for dynamic route calculation. However, only-computer-based route calculation may not be satisfactory in all situations. Thus, NaviTweet collects voice tweets from those who are in the same vehicular social network groups to allow drivers to share driving experience and decide routes based on personal preference, and suggest routes to the navigation system. We presented the design of NaviTweet, where voice feeds are collected and tweet digests are sent to users in the social group. The driver then instructs the social navigation system to avoid or choose certain routes when calculating a personalized route.

9. ACKNOWLEDGEMENT

We thank our shepherd, Marc Langheinrich and the anonymous reviewers for their insightful comments. This work has been supported in part by the NSF grants CNS-1111811 and CNS-1117711.

10. REFERENCES

[1] Study finds Los Angeles most congested city in North America. *TomTom Press Release*, July 10, 2012, Retrieved Sept. 15, 2012, http://www.tomtom.com/en_gb/licensing/press-room/press-releases/2012/Study-finds-Los-Angeles-most-congested-city-in-North-America.jsp.

[2] Eisenstein, P.A. In-vehicle navigation sales will quadruple by 2019. *The Detroit Bureau*, Sept. 30, 2012, Retrieved Sept. 30, 2012, http://bottomline.nbcnews.com/_news/2012/07/11/12683063-in-vehicle-navigation-sales-will-quadruple-by-2019?lite.

[3] Eaton, K. Map Apps to Bypass the Rivalry of Google and Apple. *The New York Times*, Oct. 3, 2012, Retrieved Sept. 15, 2012, http://www.nytimes.com/2012/10/04/technology/personaltech/navigon-and-waze-map-apps-to-bypass-apple-and-google-review.html?_r=0.

[4] Gerla, M. and Kleinrock, L. Vehicular Networks and the Future of the Mobile Internet. *Computer Networks*, 55 (2), (Feb. 2011), 457–469. DOI=http://dx.doi.org/10.1016/j.comnet.2010.10.015.

[5] Self-Driving Car Test. http://www.google.com/about/jobs/lifeatgoogle/self-driving-car-test-steve-mahan.html.

[6] Coelingh, E., Solyom, S., All Aboard the Robotic Road Train. *IEEE Spectrum*, (November 2012). Retrieved Jan. 3, 2013. http://spectrum.ieee.org/green-tech/advanced-cars/all-aboard-the-robotic-road-train.

[7] Stephen, S., Han, L., Shankar, P. and Iftode, L. RoadSpeak: Enabling Voice Chat on Roadways using Vehicular Social Networks. In *1st Workshop on Social Network Systems*, (Glasgow, Scotland, April 01-04, 2008). SocialNets'08. 43–48. DOI=http://doi.acm.org/10.1145/1435497.1435505.

[8] Vastardis, N. and Yang, K. Mobile Social Networks: Architectures, Social Properties and Key Research Challenges. *IEEE Communications Surveys and Tutorials, PP* (99), (June 2012), 1–17. DOI=http://dx.doi.org/10.1109/SURV.2012.060912.00108.

[9] Quain, J.R. Social Networking for Cars. *The New York Times*, July 20, 2010, Retrieved Sept. 15, 2012, http://wheels.blogs.nytimes.com/2010/07/20/social-networking-for-cars/.

[10] Dowling, J. Twitter comes to cars. *Drive*, Sept. 28, 2012, Retrieved Sept. 30, 2012, http://news.drive.com.au/drive/motor-news/twitter-comes-to-cars-20120928-26p5g.html.

[11] Squatriglia, C. Ford's Tweeting Car Embarks on 'American Journey 2.0'. *Wired*, May 05, 2010, Retrieved Sept. 15, 2012 http://www.wired.com/autopia/2010/05/ford-american-journey/.

[12] CB Radio. http://en.wikipedia.org/wiki/Citizens_Band_radio.

[13] Waze. http://www.waze.com.

[14] Lindqvist, J., Hong, J., Undistracted Driving: A Mobile Phone that Doesn't Distract. In *12th Workshop on Mobile Computing Systems and Applications*, (Phoenix, Arizona, USA March 1-2, 2011). HotMobile'11. 70–75. DOI=http://dx.doi.org/10.1145/2184489.2184504.