VeDi: A Vehicular Crowd-Sourced Video Social Network for VANETs

Kazi Masudul Alam, Mukesh Saini, Dewan T Ahmed, and Abdulmotaleb El Saddik, *Fellow, IEEE*
Multimedia Communications Research Laboratory, University of Ottawa, Ottawa, ON, Canada
Email:{mkazi078, msain2, dahme080, elsaddik}@uottawa.ca

***Abstract*—As one of the important members of Internet of Things (IoT), vehicles have seen steep advancement in communication technology. With the advent of Vehicular Ad-Hoc Networks (VANETs), vehicles now can evolve into social interactions to share safety, efficiency, and comfort related messages with other vehicles. In this paper, we study vehicular social network from Social Internet of Things (SIoT) perspective and propose *VeDi*, a vehicular crowd-sourced video social network for VANETs. When a user shares a video in the *VeDi*, it can be accessed by other surrounding vehicles. Any social interaction (e.g. view, comment, like) with the video on the roadway are stored in the social network cloud along with the video itself. In *VeDi*, every vehicle maintains a list of video related metadata (e.g. blur and shakiness) of available videos which are used to selectively retrieve quality videos by surrounding vehicles. We also present a method to determine representative quality scores for an entire video clip using blur and shakiness values. The prototype implementations and experimental results denote that the proposed system can be a viable option to create video social networks such as youtube, vine, and vimeo by employing vehicular crowd.**

***Index Terms*—VANETs, Video Social Network, Vehicles, Social Internet of Things**

I. INTRODUCTION

State-of-the-art vehicles are equipped with advanced technologies that enable them to communicate with nearby vehicles by forming vehicular ad-hoc networks (VANETs) [17]. There has been growing interest in building a social network of vehicles that can ensure safety of the driver and passengers, and also improve travel efficiency through collaborative applications [1] [25] [9]. While main purpose of VANETs is safety and efficiency, there is plenty of room in the allocated bandwidth for comfort applications as well [17]. In this work we study vehicular social network from video sharing perspective. We propose *VeDi*, a crowd sourced video social network over VANETs. We envision it to be integrated part of future vehicular social network and eventually Internet of Things [18].

The distribution of multimedia content over vehicular networks is a challenging task for several reasons such as network partitioning due to nodes mobility [24], and medium contention due to broadcasting nature of the technology. Therefore users cannot browse through all the videos. In *VeDi*, OBUs automatically calculate metadata description of video through content processing. This metadata description is shared among other OBUs through a Dedicated Short Range Communication (DSRC[1]) type message called *tNote*. Furthermore, it is difficult for the users to comprehend quality of complete video from individual frame quality. We experimentally analyse mobile recorded short video clips and find representative blur and shakiness scores for the entire video. The main contributions of the paper are two-fold: an architecture of crowd sourced video social network and quality based metadata description of videos.

The rest of the paper is organized as follows. Section II presents the state-of-the-art on the topic. In Section III, we introduce proposed vehicular video social network. Additional details related to the video dissemination application and video metadata analysis are provided in Section IV. Section V presents the prototype implementation details, observations and initial results. The paper is concluded in Section VI with future work directions.

II. RELATED WORKS

There has been a number of attempts on media sharing over VANETs. Many adaptations of basic VANET protocols, ranging from application to physical layer solutions, have been proposed to efficiently support video dissemination.

One of the challenging tasks in video sharing over VANETs is data forwarding. In [12] the authors consider the issue of forwarding video packets over VANET nodes. Similarly, Soldo et al. [26] overlay a grid structure on physical topology to determine video packet forwarding nodes. A routing protocol that favours high quality frames is proposed by Asefi et al. [2]. In [22], the authors further improvize routing protocols for unicast video streaming. A receiver-based intermediate node selection protocol for video streaming is proposed in [21].

Researchers have also advocated various video encoding schemes for VANETs. Gadri et al. [19] adapt video encoding scheme and error concealment to meet VANETs constraints. In [20], the authors assign different paths to different layers of SVC encoded video according to their importance. Similarly, Xing et al. [29] choose video layers of SVC coded video based on current download speed and receiver buffer level. In [28] also the authors apply a modified scalable video coding for streaming video. Asefi et al. [3] propose modifications in MAC layer of IEEE 802.11p to suit video streaming over VANETs. In [7] the authors provide an application layer approach for video streaming using p2p approach.

Guinard et al. [11] discussed how Web-of-Things can share their functionality interfaces using available human social network infrastructure such as Facebook, Linkedin, Twitter

---

[1] http://www.sae.org/standardsdev/dsrc/

etc. Smart-Its Friends [13] looked into how qualitative wireless connections can be established between *smart-artifacts*. Their system introduces context proximity based match making and respective connections. Ning and Wang provided a model of future Internet of Things (IoT) architecture using human neural network [18]. They define Unit IoT and combine various Unit IoT to form Ubiquitous IoT. Atzori et al. have introduced Social Internet of Things (SIoT) terminology and focuses on establishing and exploiting social relationships among *things* rather than their owners [4][5]. Smaldone et al. first used vehicular social network (VSN) terminology in RoadSpeak [25]. They consider the vehicular network for human socialization from entertainment, utility, and emergency messaging perspective. Hu et al. also introduced *Social Drive* system which promotes driver awareness about fuel economy using cloud computing and traditional social networks [14].

There are considerable works have been conducted on video dissemination protocols and video encoding on VANETs. In the new paradigm of SIoT, vehicles will play increasingly different role from multimedia application perspective. In our research, we present vehicular crowd-sourced social network application inline with SIoT philosophy. With the growing popularity of mobile video capturing devices and increasing personal/public vehicle culture, there comes an opportunity to utilize the VANETs infrastructure for video related social network creation. Our proposed system is placed into that intersection which leverages existing VANETs technology and focuses on future applications. In our proposed system *VeDi*, users can share their mobile videos with surrounding vehicle users. Later, users can view the videos on the roadway and engage with social interactions (e.g. comment, like, and dislike) which are aggregated and stored in the *VeDi* cloud. Such system can be employed to create shorter video social networks such as Vine[2], Instagram[3], etc. with the participation of vehicular crowd.

III. THE VEHICULAR VIDEO SOCIAL NETWORK

Vehicular Video Social Network (*VeDi*) is a virtual overlay application on top of the physical vehicular ad-hoc network of WAVE [15] communication (IEEE802.11p) model. In the *VeDi* social graph, every vehicle represents a node and any relationship between two vehicles is a link. The overall architecture of (*VeDi*) is shown in Figure 1. It consists of five components: DSRC type messages (we call it *tNote*), On-Board Unit (OBU), Road Side Unit (RSU), Home Based Unit (HBU), *VeDi* Cloud and the *VeDi* User Interface. We have adopted VANETs acronyms to describe our system and following is the detailed description of all the given components.

1) tNote Message: In a vehicular social network, vehicles share information with each other mainly through messages. To be consistent with our other ongoing works on vehicular social networks, we call these messages *tNote* messages. Every *tNote* message consists of multiple parts including user information, vehicle status, and messages related to safety, efficiency, and comfort. From video sharing perspective of *VeDi*, video metadata is stored in the *tNote* message.

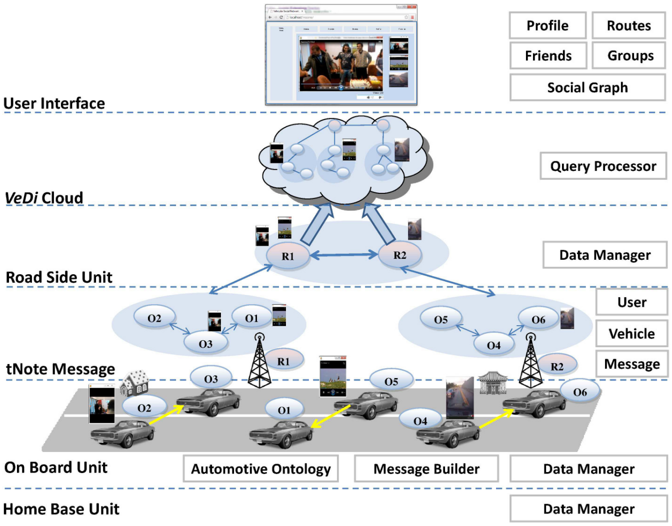


Fig. 1: Vedi: Vehicular crowd-sourced video social network architecture

TABLE I: The proposed metadata for sharing on VANET.

Attribute	Source	Range	Desired
Shakiness	Content	0 to 1	Low
Blur	Content	0 to 1	Low
Frame rate	Encoder	20 to 30	High
Resolution	Sensor	upto full HD	High
Length	Header	upto 4 minutes	User pref.
Size	Header	Content dependent	User pref.
Time	Clock	Continuous	User pref.
Location	GPS	Continuous	User pref.

Video Metadata: A user on VANETs may select video based on its spatiotemporal attributes such as time and location, or video specifications such as compression type. While these attributes are readily available at the recording devices such as smartphone, user would still want a video with good perceptual quality. Therefore, we propose use of two types of metadata: (1) video specifications (2) content analysis (i.e. metadata extraction through video processing). The metadata attributes, their sources, and desired values are described in the table I. Figure 2 shows an XML representation of video metadata snapshot of a *tNote* instance.

The main video specifications we propose to use are resolution, frame rate, length, and size. Users may prefer a particular resolution to suit their viewing device screen. While higher frame rate is generally preferred, it requires additional bandwidth. Although time and location are not exactly video specifications, we describe them here because they are fixed for a given video and do not depend on the content. While some users prefer to watch latest video, others may want to see something interesting that happened some time ago and choose an old video. By exploring a huge mobile video dataset [23], we found that the most common artifacts of mobile videos are blur and shakiness. We want to find videos that are least blurred based on [8]. Also, our goal is to find relatively stable videos from the given set by using a method inspired by prior work in [6].

---

[2] https://vine.co/
[3] http://instagram.com/

```
<tNote>
     <Id/>
     <Time/>
     <Version>1.0</Version>
     <Privacy>...</Privacy>
     <User>...</User>
     <Vehicle>...</Vehicle>
     <Message>
          <Safety>...</Safety>
          <Efficiency>...</Efficiency>
          <Comfort>
               <Video>
               <ID>Vehicle-ID-Combined-Metadata</ID>
               <Shakiness>0.2</Shakiness>
               <Blur>0.2</Blur>
               <Frame-Rate>30</Frame-Rate>
               <Resolution>720*480</Resolution>
               <Length>100</Length>
               <Size>16.66</Size>
               <Time>2014-01-12,5:33,pm</Time>
               <Location>45.422427,-75.680512</Location>
               </Video>
          </Comfort>
     </Message>
</tNote>
```

Fig. 2: XML based representation of *tNote* message with video metadata

2) On Board Unit (OBU): The OBU plays the key role in sensing and building the vehicular interaction *tNote* message. Every data unit is built following the adopted automotive ontology of vehicles [10]. *Message Builder* manages the *tNote* message structure building part. *OBU Data Manager* helps managing the data generated in OBU-OBU communication.

3) Road Side Unit (RSU): Whenever a travelling OBU comes in contact with an RSU, the RSU either pulls the *tNote* messages from the OBU or the OBU pushes the messages to the RSU. RSU collects the *tNote* bulks before storing on a cloud. The *Data Manager* ensures that the message and multimedia data is stored on the cloud with appropriate tags.

4) Home Base Unit (HBU): Every HBU has a *Data Manager* which takes *tNote* messages from OBUs and stores on the cloud. Statically neighbour OBUs are connected to the cloud by the super node HBU. Also, HBU-HBU network is established based on geographical location.

5) VeDi Cloud: This is the final infrastructure that retains all the vehicular interactions (i.e. OBU-OBU, OBU-RSU, OBU-HBU) and their related *tNote* data along with corresponding media files. It is hosted in the cloud. Timestamp allows to do various date time related roll-up, drill-down operation on the data from the UI.

6) User Interface: VeDi User Interface is a component which is further divided into subcomponents such as *Profile, Routes, Friends, Groups,* and *Social Graph*. Note that this user interface is mainly for users not driving such as passengers, and users at home. The drivers receive safety and efficiency messages through custom designed interface that may include haptic, audio, and visual elements.

IV. VIDEO DISSEMINATION APPLICATION

The proposed vehicular video social network architecture can be applied to safety, efficiency, and comfort related applications on VANETs. In this paper, we only focus on video distribution related comfort application. Passengers of a vehicle can share the metadata of their already recorded and/or downloaded videos with others vehicles in the VANETs. This allows other vehicles to choose the right video based on its passenger's interest and quality requirements. It is very common in VANETs that a vehicle leaves one network and joins other network because of varying vehicular velocity and direction. This pattern of network dynamics gives many incomplete downloads. On the other hand, if OBU settings is aware of the average duration of peer-to-peer connections in VANETs as well as the size of target media, the number of such incomplete downloads can be reduced. Thus video size is also an important factor which needs to be shared in the metadata.

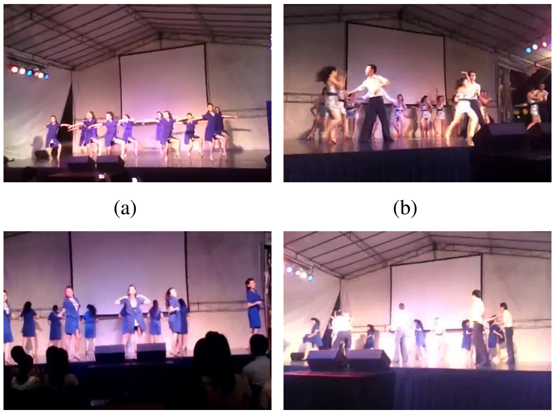
A. Video Specific Architectural Settings

In the *VeDi* system, one vehicle generates (i.e. passengers use cellphone cameras, video recorders while on the road) a video or shares already available mobile videos and the *Message Builder* augments video related metadata with the *tNote* message. When the owner of a video shares the media in the *VeDi* then only related metadata is added to *tNote* and it becomes visible to other neighboring vehicles. Every shared video in the *VeDi* gets a unique key by combining vehicular details, user details, time and the video metadata. In the OBU-OBU interaction, the other OBU passengers see a ranked list of videos shared around them and can consume the better one. This whole procedure can be automated in the OBU video consumption settings as well. Video consumption is initiated by the OBU passenger and the *tNote* video metadata transmission follows *pull* command rather than *push* for the safety messages.

As soon as, an OBU consumes the video data from another OBU and the transmission is completed, it is stored in the on board video storage of the consumer vehicle. The overlay OBU-OBU logical relation also saves a link with the stored video. Passengers of the consumer vehicle can enjoy the video from their vehicular storage. In the OBU-RSU interaction phase, only the bulk *tNote* messages are transferred from the OBU-RSU excluding the stored videos. Only the video owner OBU can transfer the unique video to the RSU. When a video is consumed multiple times in different RSUs even then OBU-RSU video transmission of a particular video occurs once. This method virtually eliminates data redundancy related to video sharing in the cloud. The video viewing history is stored in the *tNote* message of the consumer vehicle. Then RSU combines the video viewing history (i.e. count, like, dislike, and comment) before transferring any video to the *VeDi Cloud*. We can also reduce the video transmission load in the VANETs by diverting the original video upload in cloud to the OBU-HBU interaction. In this case, video is synced to the *VeDi Coud* when the video owner's vehicle arrives to its home HBU. Any types of social activities are synced from RSU to the *VeDi* cloud.

B. Video Score Modeling

Since video metadata plays a key role in the video social network building block hence, we developed blur and



(a) (b)

(c) (d)

Fig. 3: Representative frames of the dataset from four videos.

TABLE II: Dataset Description.

Item	Value/Description
Number of videos	12
Frame rate	30 frames per second
Resolution	720*480
Duration	4 minutes each
Compression	mp4
Type of event	Dance performance
Recording device	Android based smartphones

shakiness models which are presented below. Other metadata attributes are readily available through sensory devices and video file headers.

1) Objectives: We conduct a number of experiments to find appropriate blur and shakiness scores for the videos. The main objective of the experiments is to analyze shakiness and blur in mobile recorded videos over time and obtain representative scores. We study the distribution of shakiness and blur and propose metrics that can effectively represent the quality of the entire clip.

2) Dataset: We used twelve simultaneous recordings of a dance performance from Jiku dataset [23]. We have chosen these videos because these are recorded by mobile users in unconstrained scenario. Since the connection life is generally short on VANET, we have only considered 4 minute clips.

3) Shakiness Analysis: We first calculate camera motion and then find corresponding shakiness component of that motion. To calculate camera motion, we project the pixel values on x-axis and y-axis. For an image $I(i,j)$ of size $(m \times n)$, the projection of x-axis, $P^x(i)$, and y-axix, $P^y(j)$, are calculated as follows:

$$P^x(i) = \sum_{j=0}^{n-1} I(i,j) \tag{1}$$

$$P^y(j) = \sum_{i=0}^{m-1} I(i,j) \tag{2}$$

There can be two types of motions, smooth intentional camera motion (pan) and shakiness due to hand movement. While camera panning makes a video interesting, shakiness is an undesired feature. To differentiate between these two types of motions, we collect τ frames and apply a median filter on the recorded motion vectors. The residual is the absolute shakiness value ξ. We normalize the shakiness value by using the following equation:

$$\psi^i = \begin{cases} \frac{\xi}{\beta} & \text{if} \quad \frac{\xi}{\beta} < 1 \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

where ψ^i is the shakiness value of an individual image and β is normalizing coefficient. It is found to be 300 for the given dataset. The value of β should not vary significantly for other videos as well since it mainly depends of human arm movement. The shakiness value is shown in the Figure 4. We notice periodic spikes in the shakiness value. These spikes are because of tiredness, excitement, or any other physical phenomenon of the recorder. We want as less spikes as possible.

Figure 5 shows the histogram of the shakiness values. Most of the shakiness values are around zero due to natural hand movements. For Video 1 and Video 2, we notice significant number of frames having large shakiness value (Figure 5(a) and 5(b)), while for Video 4 and Video 6, most of the frames have shakiness close to zero (Figure 5(d) and 5(f)). This result is also verified by watching the videos. To abstract a single value for given video clip, we define a tolerance coefficient $\lambda(0,1)$. For a given tolerance coefficient λ, the shakiness score for the video is the fraction of video that has less shakiness than λ, i.e.,

$$\psi^v = \frac{1}{T} \sum_{i=1}^{T} 1.\delta(\psi^i) \tag{4}$$

where ψ^v is shakiness score for full video, T total number of frames in the video and $\delta(x) = 1$ for $x > \lambda$, otherwise 0. For experimental purposes, we have take $\lambda = 0.1$, $\tau = 30$ and $T = 7200$ because the videos are recorded at 30 frames per second and each video is 4 minutes long. Note that even we have already filtered the smooth camera motion. Therefore, even small values of ψ^i indicates camera shake, which could be annoying to the viewer.

4) Blur Analysis: Since we do not have any reference video, we opt for a no-reference blur calculation method described in [8]. The method is based on the following observation: absence of high frequency components causes blurred image. If we blur a sharp image, the pixel neighbours will change largely because it has high frequency components. On the other hand, if we change already blurred image, the pixel neighbours will only change by small extent.

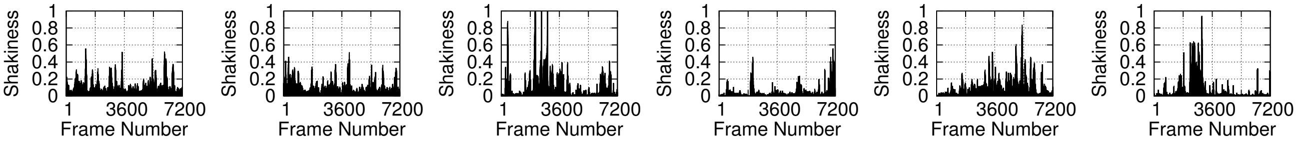
Let I^b the blurred image obtained by applying a strong low pass filter h to image I, i.e.,

$$I^b = h * I \tag{5}$$

Now if function V returns the pixel neighbour variations in a given image, the blur value for a given image, ν^i, is calculated as follows:

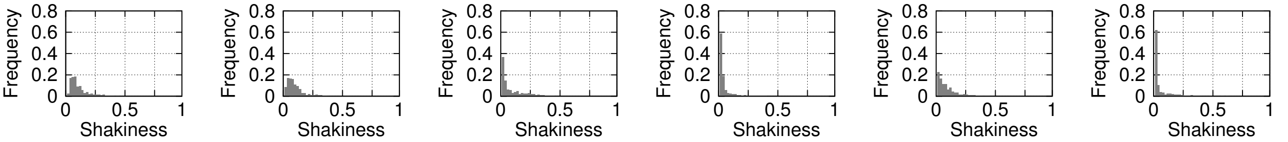
$$\nu^i = \frac{V(I) - V(I^b)}{V(I)} \tag{6}$$

Figure 6 shows the blur values for the twelve videos of our dataset. We notice that the blur values are mostly consistent



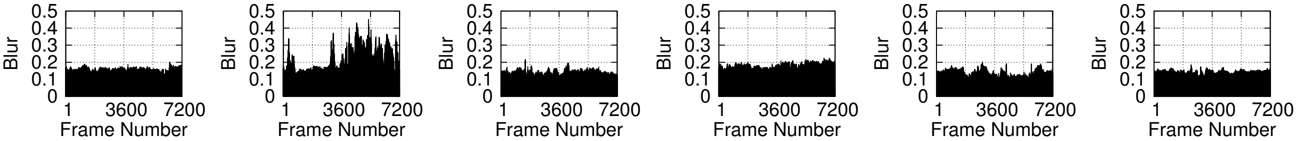
(a) Video 1 (b) Video 2 (c) Video 3 (d) Video 4 (e) Video 5 (f) Video 6

Fig. 4: The shakiness values for all frames.



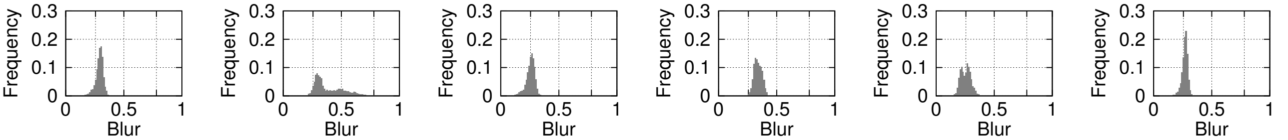
(a) Video 1 (b) Video 2 (c) Video 3 (d) Video 4 (e) Video 5 (f) Video 6

Fig. 5: The histograms of corresponding shakiness values from Figure 4.



(a) Video 1 (b) Video 2 (c) Video 3 (d) Video 4 (e) Video 5 (f) Video 6

Fig. 6: The blur values of all video frames.



(a) Video 1 (b) Video 2 (c) Video 3 (d) Video 4 (e) Video 5 (f) Video 6

Fig. 7: The histogram of the corresponding blur values.

for a given short video clip. Because short connections over VANETs only allow sharing small clips, we can take a tuple consisting of mean (μ_v) and variance (σ_v) of blur values as representative score for the video. Figure 7 shows the normalized histogram of blur values which further confirms that the blur values follow Gaussian distribution. Except Video 5 (Figure 7(e)), all other videos have a single peak. We manually analyzed Video 5 and found that two main zoom settings are used to capture the video, resulting in two different blur values. Yet, the the two peaks are close enough to each other for the give short video clip of 4 minutes.

5) Final Scores: Figure 8 shows the blur and shakiness scores for all videos. For blur, we have only plotted mean values while for shakiness we have plotted ψ^v (Equation 4). We see that shakiness varies largely across videos while blur has less variation except video 2. Video 3 has small blur but large shakiness whereas for video 11 it is opposite, i.e., more blur and very small shakiness. Hence, videos may have conflicting scores.

The final quality scores for each video is calculated as weighted sum of individual quality scores, i.e.,

$$f_v = \omega_1 * \psi_v + \omega_2 * \mu_v \tag{7}$$

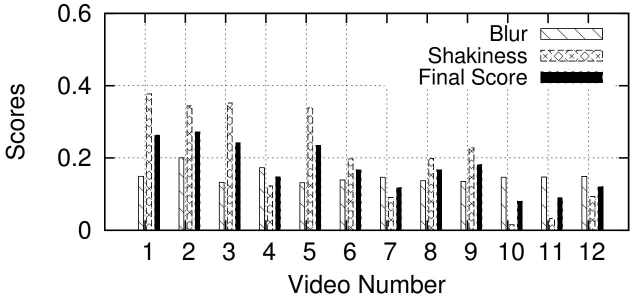


Fig. 8: Extracted blur and shakiness metadata for the videos in dataset.

where $\omega_1 + \omega_2 = 1$. We can see in the Figure 8 that Video 10 is best quality with minimum score. The final video ranking would also depend on user preferences in terms of ω_1 and ω_2, which are taken equal (0.5) in the given experiments. For a dance video, user may tolerate shakiness to some extent, but it could be annoying in a singing video. Similarly, we could be more tolerant to blur in a singing video.

C. Algorithm

When a user chooses to watch a video over *VeDi*, the application first retrieves the metadata of the videos that are available with other OBUs. The requesting OBU filters the metadata according to the user's preferences and then ranks the videos according to the quality. Finally, the best video is selected followed by lower ranked videos if time permits.

Code 1: A pseudo instance of *tNote* with video social data

```
{
  id{
    messageID 100010,
    time{
      year 2014, month 4, day 24, hour 4, minute 53
    }
  },
  version 1.0,
  generator{
    vehicleID{
       name ”Toyota”,
       vin ’vehicle-identification-number’H,
       ownerCode ”owner-code  ,
       vehicleType car
    },
    driver{
    },
  },
  message{
   comfortMessage{
       videoMessage{
         videoID ”unique-video-id”,
         shakiness 0.2, blur 0.2,
         frameRate 30, resolutionX 720, resolutionY 480,
         length 100, size 16.66,
         like 10, dislike 2, view 50,
         comment{
           ”Awesome video! thnx for sharing”
         }
       }
    }
  },
  privacy{
    {
      publicPrivacy{ comfort }
    }
  },
}
```

V. PROTOTYPE IMPLEMENTATION AND INITIAL RESULTS

The objective of the prototype implementation is to verify the proof of concept. For this purpose, we have developed the *tNote* message using Abstract syntax notation one (ASN.1)[27] following the DSRC[4] type structure and have mocked the *tNote* VANETs system architecture (Figure 11). In our mock VANETs setup, vehicles are represented using mobile tablets since they are equipped with Wi-Fi and GPS, and RSU is represented using laptop. The final *VeDi* social network server is connected to the laptop using a Ethernet network interface.

A. tNote Message

We have adopted the ASN.1 syntax to describe the full *tNote* message structure. OSS Nokalva ASN.1 Studio has been used to build the tNote message structure. The details about the message structure is out of scope for this paper. Above Code 1 represents a value instance of a *tNote* message with video related meta and social data.

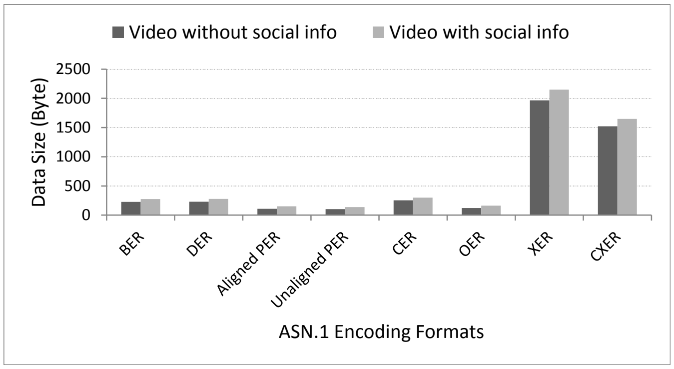


Fig. 9: Video meta data message size for different ASN.1 data encoding type

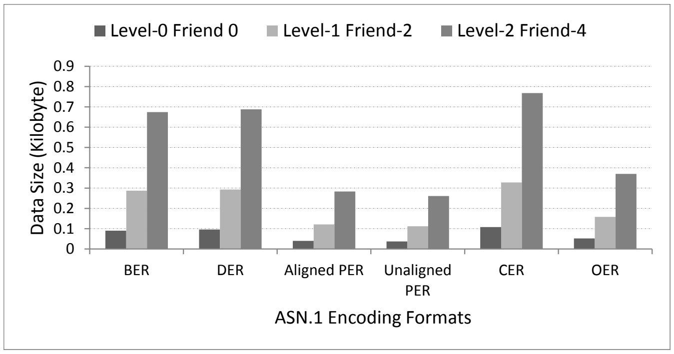


Fig. 10: Different ASN.1 data encoding size for tNote friends information

In Figure 9 , we find a comparative presentation of data size using various X.690 encoding formats of *tNote* messages. In this analysis, we compare the data size of *tNote* message both with presence and absence of social information such as count of view, like, dislike and related comments. Here we see that PER type encoding gives the smallest data. And, XML representation XER, CXER encoding gives the highest size. But, BER-DER-CER are preferable for their Type-Length-Value (T-L-V) structure. Again from Figure 10, we find the comparison of byte type encoding formats at different levels of friend population size. It is evident from the figure that with the increase of depth level of friend tree, the size of *tNote* grows. Here we have considered that each vehicle has seen only two other vehicles which will increase with the speed and communication range of a deployed OBU. By considering the OBU-OBU communication range, the availability of RSU, bandwidth availability of VANETs service channel [16], and infrequency of comfort message exchanges, the system appears viable for vehicular crowd-sourced video social network generation.

B. tNote System Infrastructure

1) Setup: We represent OBU as Samsung Galaxy Tab 10.1 Android tablet (Wi-Fi 802.11a/b/g/n), Google Nexus 7 Android Tablet, and RSU as windows Acer Aspire V5 laptop (Wi-Fi 802.11b/g/n), and *VeDi* cloud as Dell desktop server (i.e. RAM:12GB, Processor:Intel(R)Core(TM)i7 CPU

---

[4] http://www.sae.org/standardsdev/dsrc/DSRC_R36_Source.ASN

930@2.80Gz) (Figure 11). Acer laptop and Dell desktop are connected using Ethernet Local Area Network. Tablets use Wi-Fi Direct technology to behave as network access points. Tablets can communicate with each other and with the laptop wirelessly.

2) Implementation Details: Three tablets play their role in vehicular OBU-OBU communication. The *tNote* messaging platform for vehicles are implemented in Android platform. Two Galaxy tabs have special settings for Wi-Fi Direct technology, and the Google Nexus 7 Wi-Fi has builtin support for Wi-Fi Direct. We have used the *WifiP2pManager*[5] Android class for device discovery and peer-to-peer connection. When each OBU tablet comes into the wireless range of another OBU tablet they establish a TCP socket connection to exchange *tNote* messages. Dynamic vehicle join in and leaving from the network is mocked by taking the tablets away from peers range and bringing them close again.

When a user wants to share a video in the *VeDi* social network, he uses the tablet user interface and gives his consent. The meta data related to the video is then available through the socket connections to the neighboring tablets. If the receiver tablet application is configured to accept floating videos then it saves the video in the local storage and creates a viewer *intent*[6].

For simplicity, *tNote* is an XML message and the video interaction (i.e. comment, like, dislike) data are also stored in the XML structure. When the video creator tablet comes in close to the laptop, then video data is transferred to the laptop (i.e. RSU). RSU runs JAVA application developed in JDK1.7.0. In the RSU laptop, Wi-Fi network interface manages the tablet communication, and the Ethernet network interface manages the *VeDi* server and laptop communication. RSU laptop runs Windows 7 operating system. We have used the *Software Network Bridging* method built-in to Windows 7 to connect the wireless and the wired network interfaces.

RSU laptop continuously maintains two socket connections: one with the Dell server *VeDi* cloud and another from wireless to wired network interface ports. Video data are forwarded through these two sockets to the *VeDi* cloud. *VeDi* social network presentation only displays the videos in the HTML format along with their likes, dislikes, and comments. Videos are stored in the file system along with their links in the MySQL database server. We use JDBC for database communication. The *VeDi* interface is developed using JSP technology.

3) Observations: From the observation, we see that relatively longer video transfer can experience link drops as tablets in motion can get out of range to the connected peers. The limitation of the mock setup is the quick battery drainage problem. Since, tablets behave as network access points, hence they consume a lot of battery power which makes it difficult to test the system. Another difficult part for the implementation is the lack of Wi-Fi emulators which forces all the tests to the physical devices which requires many human resources to be physically available to carry tablets and log observations.

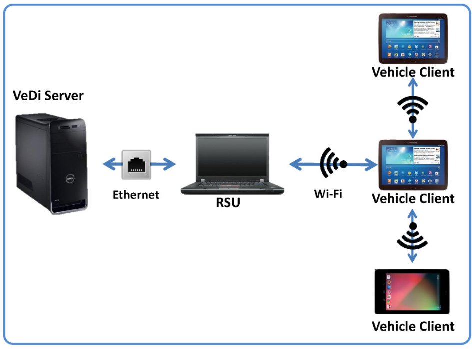


Fig. 11: Prototype implementation setup of *VeDi*

VI. CONCLUSION AND FUTURE WORKS

Sharing video over VANETs is challenging due to dynamic and unpredictable topology, low bandwidth, and fleeting connections. In this paper, we have proposed a framework, *VeDi*, for vehicular crowd sourced video social network over VANETs. In the proposed work, vehicles share metadata based description of videos that are captured by the occupants of the vehicle and are accessible to surrounding vehicles. The metadata consists of video specifications and derived blur and shakiness measures. These metadata scores help video consumers to select the right video while on the roadway. *VeDi* reduces the overall bandwidth consumption as users can select most appropriate video without downloading them all. We have provided implementation technique of DSRC type *tNote* message and encoding size analysis at various system instances. Details about system architecture implementation approach is also provided with various observations. In our future work, we envision to present the modelling and simulation results of the proposed system along with scalability measurements and required optimizations.

REFERENCES

[1] N. Abbani, M. Jomaa, T. Tarhini, H. Artail, and W. El-Hajj. Managing social networks in vehicular networks using trust rules. In *Wireless Technology and Applications (ISWTA), 2011 IEEE Symposium on*, pages 168–173, Sept 2011.

[2] M. Asefi, J. W. Mark, and X. Shen. An application-centric inter-vehicle routing protocol for video streaming over multi-hop urban vanets. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[3] M. Asefi, J. W. Mark, and X. Shen. A mobility-aware and quality-driven retransmission limit adaptation scheme for video streaming over vanets. *Wireless Communications, IEEE Transactions on*, 11(5):1817–1827, 2012.

[4] L. Atzori, A. Iera, and G. Morabito. Siot: Giving a social structure to the internet of things. *Communications Letters, IEEE*, 15(11):1193–1195, 2011.

[5] L. Atzori, A. Iera, G. Morabito, and M. Nitti. The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer Networks*, 2012.

[6] M. Campanella, H. Weda, and M. Barbieri. Edit while watching: home video editing made easy. In *Proceedings of SPIE*, volume 6506, 2007.

[7] Y.-C. Chu and N.-F. Huang. Delivering of live video streaming for vehicular communication using peer-to-peer approach. In *2007 Mobile Networking for Vehicular Environments*, pages 1–6. IEEE, 2007.

[8] F. Crete, T. Dolmiere, P. Ladret, and M. Nicolas. The blur effect: perception and estimation with a new no-reference perceptual blur metric. *Human vision and electronic imaging XII*, 6492, 2007.

---

[5] http://developer.android.com/reference/android/net/wifi/p2p/WifiP2pManager.html

[6] http://developer.android.com/reference/android/content/Intent.html

[9] R. Fei, K. Yang, and X. Cheng. A cooperative social and vehicular network and its dynamic bandwidth allocation algorithms. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 63–67. IEEE, 2011.

[10] M. Feld and C. Müller. The automotive ontology: managing knowledge inside the vehicle and sharing it between cars. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 79–86. ACM, 2011.

[11] D. Guinard, M. Fischer, and V. Trifa. Sharing using social networks in a composable web of things. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 702–707, March 2010.

[12] M. Guo, M. H. Ammar, and E. W. Zegura. V3: A vehicle-to-vehicle live video streaming architecture. *Pervasive and Mobile Computing*, 1(4):404–424, 2005.

[13] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Ubicomp 2001: Ubiquitous Computing*, pages 116–122. Springer, 2001.

[14] X. Hu, V. Leung, K. G. Li, E. Kong, H. Zhang, N. S. Surendrakumar, and P. TalebiFard. Social drive: a crowdsourcing-based vehicular social networking system for green transportation. In *Proceedings of the third ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, pages 85–92. ACM, 2013.

[15] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *Communications Surveys & Tutorials, IEEE*, 13(4):584–616, 2011.

[16] Y. L. Morgan. Managing dsrc and wave standards operations in a v2v scenario. *International Journal of Vehicular Technology*, 2010, 2010.

[17] Y. L. Morgan. Notes on dsrc & wave standards suite: Its architecture, design, and characteristics. *Communications Surveys & Tutorials, IEEE*, 12(4):504–518, 2010.

[18] H. Ning and Z. Wang. Future internet of things architecture: Like mankind neural system or social organization framework? *Communications Letters, IEEE*, 15(4):461–463, April 2011.

[19] N. Qadri, M. Altaf, M. Fleury, M. Ghanbari, and H. Sammak. Robust video streaming over an urban vanet. In *Wireless and Mobile Computing, Networking and Communications, 2009. WIMOB 2009. IEEE International Conference on*, pages 429–434. IEEE, 2009.

[20] A. Razzaq and A. Mehaoua. Video transport over vanets: Multi-stream coding with multi-path and network coding. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pages 32–39. IEEE, 2010.

[21] C. Rezende, A. Mammeri, A. Boukerche, and A. A. Loureiro. A receiver-based video dissemination solution for vehicular networks with content transmissions decoupled from relay node selection. *Ad Hoc Networks*, 2014.

[22] C. Rezende, H. S. Ramos, R. W. Pazzi, A. Boukerche, A. C. Frery, and A. A. F. Loureiro. Virtus: A resilient location-aware video unicast scheme for vehicular networks. In *Communications (ICC), 2012 IEEE International Conference on*, pages 698–702. IEEE, 2012.

[23] M. Saini, S. P. Venkatagiri, W. T. Ooi, and M. C. Chan. The jiku mobile video dataset. In *Proceedings of ACM Multimedia Systems Conference*, pages 108–113. ACM, 2013.

[24] Z. Shi, C. Beard, and K. Mitchell. Competition, cooperation, and optimization in multi-hop csma networks. In *Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 117–120. ACM, 2011.

[25] S. Smaldone, L. Han, P. Shankar, and L. Iftode. Roadspeak: Enabling voice chat on roadways using vehicular social networks. In *Proceedings of the 1st Workshop on Social Network Systems*, SocialNets '08, pages 43–48, New York, NY, USA, 2008. ACM.

[26] F. Soldo, C. Casetti, C. Chiasserini, and P. A. Chaparro. Video streaming distribution in vanets. *Parallel and Distributed Systems, IEEE Transactions on*, 22(7):1085–1091, 2011.

[27] D. Steedman. *Abstract syntax notation one (ASN. 1): the tutorial and reference*. Technology appraisals, 1993.

[28] A. Vinel, E. Belyaev, O. Lamotte, M. Gabbouj, Y. Koucheryavy, and K. Egiazarian. Video transmission over ieee 802.11 p: real-world measurements. In *Communications Workshops (ICC), 2013 IEEE International Conference on*, pages 505–509. IEEE, 2013.

[29] M. Xing and L. Cai. Adaptive video streaming with inter-vehicle relay for highway vanet scenario. In *Communications (ICC), 2012 IEEE International Conference on*, pages 5168–5172. IEEE, 2012.