

ISSN 1999-5903 www.mdpi.com/journal/futureinternet

Article

# Sending Safety Video over WiMAX in Vehicle Communications

Jun Steed Huang<sup>1</sup>, Oliver Yang<sup>2,\*</sup> and Funmilyo Lawal<sup>2</sup>

- <sup>1</sup> Suqian College, Jiangsu University, Jiangsu 223800, China; E-Mail: steedhuang@ujs.edu.cn
- <sup>2</sup> Computer Communication Network Research Lab, School of Electrical Engineering and Computer Science, University of Ottawa, 161 Louis Pasteur, Ottawa, ON K1N 6N5, Canada; E-Mail: funmilayo.lawal@gmail.com
- \* Author to whom correspondence should be addressed; E-Mail: yang@eecs.uottawa.ca; Tel.: +1-613-801-8402.

Received: 6 June 2013; in revised form: 8 August 2013 / Accepted: 28 August 2013 / Published: 31 October 2013

**Abstract:** This paper reports on the design of an OPNET simulation platform to test the performance of sending real-time safety video over VANET (Vehicular Adhoc NETwork) using the WiMAX technology. To provide a more realistic environment for streaming real-time video, a video model was created based on the study of video traffic traces captured from a realistic vehicular camera, and different design considerations were taken into account. A practical controller over real-time streaming protocol is implemented to control data traffic congestion for future road safety development. Our driving video model was then integrated with the WiMAX OPNET model along with a mobility model based on real road maps. Using this simulation platform, different mobility cases have been studied and the performance evaluated in terms of end-to-end delay, jitter and visual experience.

Keywords: vehicular camera; WiMAX; Ad-hoc; video; modeling

# 1. Introduction

By virtue of its versatility and utility convenience, mobile wireless communication has now become a necessity. Over the years, different technologies have been developed to increase the data rate in wireless networks. Technologies like WiMAX (Worldwide Interoperability for Microwave Access) and 3G/4G devices can support 100 Mbps per data channel [1], and LTE (Long Term Evolution) can go as high as 200 Mbps. This has created room for different types of applications in different

specialties. Users have come to desire not only the basic voice functionality of wireless devices, but also different forms of mobile multimedia communication. Video communication via a VANET (Vehicular Adhoc NETwork) is one such application.

Safety and environmental issues in managing existing road networks have become one of the greatest concerns of both the automotive industry and governmental bodies. A VANET consisting of clusters of vehicles communicating with each other and also with roadside infrastructure can provide a solution to these issues. Telematics for the telecommunications of safety information is one major tool used in this area. Progressively, sharing real-time driving video traffic among vehicles and roadside infrastructures would be the next natural step to enhance the safety feature. One motivation of this paper comes from the statistics of road traffic fatalities which shows that there were 44,192 accidental deaths in Canada from the years 2000 to 2004 [2]. California, with a population of 37 million people reports, one million vehicle accidents each year [3]. The increase in road traffic volume over the years has led to a concern among researchers to create an effective, efficient and safe automotive environment with the help of video communication. One example is the "smart car" in the state of Nevada which, as a first step towards this goal, uses video to assist hands-free driving.

Unlike other wireless environments that are mostly stationary or under slow motion, transmitting video signals in a VANET poses more challenges to resolve. Electromagnetic interferences such as those from engine electronics, from Additive White Gaussian Noise, or those due to weather conditions can affect the QoS (Quality of Service) as seen by a wireless user. Since the topology is constantly changing, a vehicle could move out of sight to cause an outage in data transmission. Obstacles along LOS (Line of Sight) communication are another major reason to cause an outage. Finally, packet loss due to data traffic congestion will definitely make things worse, and communication disruptions may become inevitable.

Understanding the characteristics of real-time video would allow the design of efficient real-time transmission/transportation protocols to address the above challenges and to allow us to carry out further study. There has been continuous and extensive work on VBR video from various specific video sources, (e.g., from video conferencing cameras, web cameras, broadcast TV or HDTV [4,5], and non real-time processing/storage of data taken from the cameras installed in a real public safety law enforcement car system [6]. However, images from a vehicular camera have higher activity around all edges than the middle portion. This is very different from a normal camera constructed for stationary scenarios, and can be addressed/utilized in the design of video communication. So far, we are not aware of any comprehensive study related to real-time driving video traffic originating in the actual vehicle camera, such as GV1500 from GenieView Inc. [7]. Furthermore, all these prior arts focus on the modeling of separate and independent video traffic before the wireless networks, and there is no work done on the video model of the traffic coming out of a wireless link, as will be needed in our later study of highway video through a VANET.

Higher layer protocols are usually required to support various video applications. TCP (Transport Control Protocol) [8] and UDP (User Datagram Protocol) [9] for real-time transportation of video have been shown to be ineffective. As is further reviewed in Section 2, TCP is best used for lossless data transfer as it ensures packet delivery by its support of packet retransmission. This poses a problem for real-time traffic, as there is no time for retransmission [10]. UDP is better for video traffic, but unfortunately still cannot guarantee real-time and loss performance. RTP (Real-Time Transport

Protocol) [11] offers an alternative but has other shortcomings that need to be remedied, as is discussed later. Most of the aforementioned transport algorithms and the traffic control protocols below are built on top of TCP and have only been tested in wired networks. Quite often, packets in a wireless network are stored in a buffer waiting for proper bandwidth allocation before they can be delivered. In real-time communication, a poorer quality can be tolerated, but delay should be reduced to a minimum if not eliminated.

Since a large volume of video traffic can easily consume the limited wireless bandwidth, there is a need to throttle the video traffic for better bandwidth utilization and avoid congestion. Traffic/Congestion control can provide smooth throughput and high link utilization for real-time data. Of the two main types of traffic congestion, control in computer networking (reviewed in Section 2), implicit congestion control does not provide smooth throughput for real-time and low link utilization. Also, the control is based on packet loss-events which are not a big concern for real-time traffic as mentioned earlier. Explicit congestion control of traffic from the router node becomes impractical in a high speed when the CPU cannot catch up with the updating of events arising from changing driving speeds. Consequently, a different protocol (desirably with a low complexity) should be considered.

Finally, it is safer and more cost efficient to simulate possible solutions than actual field experimenting with highway driving. An effective VANET simulation platform/model has pertinent importance in research and industry to emulate a real world situation. One major challenge is the development of an effective platform that can bring all issues described earlier under one simulation model. A major issue is the integration of an effective mobility model that considers vehicle-to-vehicle interaction and vehicle-to-infrastructure interaction along with the full functionalities of a communication device capable of effective receiving, processing and transmitting capabilities. Although there has been work on creating a similar platform, e.g., [12–14], little work has been done on certain issues related to video streaming from vehicle camera such as the characterization of the data traffic (e.g., its unique packet size distribution) and the behavior under different driving conditions (e.g., highway *vs.* local).

In view of the above and others reviewed in the related work in the next section, we seek to create a robust platform for effective VANET simulation, one that can provide an environment to test, develop and deploy effective communication protocols that would enhance real-time communication of video information from vehicle camera. As forerunners, in this paper, we would like to:

- (i) create a real-time camera video model under VANET conditions;
- (ii) integrate a realistic VANET mobility model to test the testbed at various speeds;
- (iii) integrate a practical algorithm to survive network congestion;
- (iv) demonstrate the capability of such a platform.

To achieve the above objectives, we first adopt the protocol architecture of our simulation platform as shown in Figure 1. We need to adopt appropriate protocols for the lower layers consisting of the Data Link Layer and the Physical Layer, and the upper layers for the remainder. After a comparison of several viable technologies for the lower layers such as the pros and cons as reviewed in Section 2, we have chosen WiMAX (Worldwide Interoperability for Microwave Access) for its cost-effective approach that can provide a high data rate that can satisfy the needs of our vehicle camera users (low latency and high coverage) at high speed at an affordable cost. It is scalable because its uses OFDMA (Orthogonal Frequency Division Multiple Access) to allow flexible frequency re-use. It implements full MIMO (Multiple-Input and Multiple-Output) which is good for mobile applications by enhancing timely information delivery to save lives and improve quality of life. In view of the complexity problem in implementing an explicit congestion control mechanism in the upper layers to control streaming traffic from driving video in real-time, our present approach is not to "cure" congestion but to "survive" congestion. We propose to reduce the compression rate of the video packets once congestion is detected in the network. We use the RTCP (Real Time Control Protocol) to feed such information back to the sender.

	Sub-layer 3	RTCP
Layer 4	Sub-layer 2	RTP
	Sub-layer 1	Data Traffic Controller
T	Sub-layer 2	UDP
Layer 3	Sub-layer 1	IP
I	Sub-layer 2	Data Link Port
Layer 2	Sub-layer 1	Data Link Buffer
т 1	Sub-layer 2	Physical Port
Layer I	Sub-layer 1 Physical Transpo	Physical Transponder

Figure 1. Overall Architecture.

The next step is to find a simulation tool and a realistic vehicular mobility model for integration. Of the few available network simulation languages such as Qualnet and NS-2, we have chosen OPNET because it has different wireless models including WiMAX. It has a rich set of video streaming protocols, and it provides a good framework, which allows us to integrate our mathematical driving video model effectively so that we can explore different scenarios during network traffic congestion. After searching, we have also decided that VanetMobiSim [15,16] is the only VANET mobility model that would allow the output code generator to be modified in order to generate and output traces useable by other simulators. Consequently, we can integrate its traces into our existing OPNET model using the open-loop approach.

To support the experiments from our simulation platform, we need to use realistic driving video traffic streams. Since we do not have an additional ACE [17] license to support the feeding of life data into our OPNET simulator, we shall generate our traffic streams based on a model established from real video traffics transported over wireless networks. Not only do we want to use this model to generate traffic traces quickly, we would also use it to alter the traffic characteristic under different scenarios later on without having to recapture traffic traces from life experiments, which can be both costly and time consuming.

The contributions of this paper lie in the following:

(i) The design and implementation of a theoretical but more realistic driving video model than the traditional video model. To the best of our knowledge, there is no comprehensive study related to real-time driving video traffic originated from a vehicle camera, nor any work done on the modeling of the video traffic from the vehicular camera system through the wireless link;

- (ii) The integration of a VANET mobility model with such a driving video model to provide a practical VANET simulation platform for the evaluation of video traffic under different scenarios;
- (iii) The design and implementation of a practical traffic controller for driving video over RTP applications.

The rest of the paper is organized as follows. Section 2 discusses existing works related to our research, especially their deficiencies. Section 3 discusses the network operations and models used in our testbed. To illustrate the procedure of modeling the driving video traffic in VANET, Section 4 presents an example of the trace collection and stochastic analysis of the trace. We then discuss the OPNET simulation of different scenarios and evaluate their performance in Section 5.

## 2. Related Works

Below are works related to our research work on VANET testbeds, models for driving video, as well as the lower and upper layer protocols.

## 2.1. VANET Testbeds

There have been practical implementations of VANET testbeds. For example, a testbed was implemented to extract H.263 video streams through an ADSL (Asymmetric Digital Subscriber Line) gateway to an SQL server in the Internet [18]. A prototype testbed [19] using a handheld device with an Android operating system was presented for data-intensive application. While these approaches are effective, their hardware implementations are expensive and do not allow quick and easy changes to adapt to various industry scenarios and implementation of special safety applications. Video cameras make up a key component in VANET due to their ability to aid navigation safety and give road users and relevant authorities a precise and clear image of the traffic conditions when necessary. Intel Corporation has decided to lead the development of a video camera system on board of a car [20]. Others include the CogniVue system [21]. Since a special high-definition camera can pump into the network to degrade network performance and video quality, this has been a key concern in VANET design. Solving this issue will unlock the power of driving camera video applications.

In light of the outlined efforts, one of the major focuses of VANET research is the creation of an effective simulation platform that can integrate a network simulator with a realistic vehicular traffic simulation model. Unlike other network environments, VANET mobility has a peculiar and unique nature due to the randomness of human behavior. One classification groups vehicular traffic under four models [22]: Submicroscopic, Microscopic, Mesoscopic and Macroscopic. It is important to choose a realistic mobility model to correspond to each one. An open-loop integration approach and a closed-loop integration approach have been discussed for the integration of the network model with a VANET mobility model. Although there has been other work on creating a similar platform (e.g., [12–14]), to the best of our knowledge, there is hardly any work on the characterization (e.g., the packet size distribution) of video data streaming from the vehicle camera and the behavior under different data traffic conditions. The closest work is on UAV video [23], but this was done from an image process perspective. A broad view of the state-of-the-art mobility models adapted for VANETs was discussed [24], which also provided a detailed survey and comparison of mobility models.

However, it only provides a model for data transmission and not for video transmission. Such a model needs to consider other factors (like the physical layer communication mechanism and control algorithms in the application layer for real-time video streaming) for effective VANET simulation. Note that a model or a tool may be good for one aspect/feature, but their integration is necessary to model an effective safety solution.

# 2.2. Modeling Driving Video Traffic

Unlike other forms of multimedia traffic (like data, voice and video), driving video traffic has a very dynamic stochastic nature. Different statistical studies [25] have already revealed that the inherent characteristics (not source nor codec specific) of Variable Bite Rate (VBR) video traffic is the LRD (Long Range Dependence) which has a significant impact on the queuing performance and characterization of video traffic [25,26]. Video traffic modeling over wireless has been studied extensively. There is work on the video traffic modeling in a wireless environment with emphasis on the streaming video over WiMAX networks [27]. The I, P and B frames have been modeled separately according to their types, and a fixed pattern is used to combine these frames together to form the GOP (Group of Picture) [27-29]. A video traffic model was obtained using a trace generated with a fixed frame inter-arrival time [30]. All these prior works have focused on the separate and independent video modeling before the traffic is fed into the wireless networks, and none has considered the video model after coming out of the wireless link. Such modeling is required in our highway video forwarding through a number of consecutive cars. Furthermore, we are not aware of any comprehensive study related to real-time driving video traffic originated from the actual vehicle camera, such as GV1500 made by GenieView Inc. [7]. Due to the very driving nature, such a video has an even higher statistical fluctuation that is very hard to control. More discussions can be found in further sections such as 1 and 4.3.

## 2.3. Lower-Layer Protocols for VANET

VANET applications can be supported through V2R (Vehicle-TO-Roadside) and V2V (Vehicle-to-Vehicle) communications. IEEE 802.11 (WiFi) [31], IEEE 802.16 (WiMAX) [32] and DSRC (Dedicated Short Range Communications) [33] technologies can be used in this model of communication.

With the DSRC standard, OBU (OnBoard Units) placed at each vehicle can send or receive data to or from roadside units (RSUs), as well as communicating directly with each other. There are thus many potential applications in urban areas (or in populated countries in Europe or Southeast Asia) such as forward collision warning, electronic parking payments, and approaching emergency vehicle warning [34]. Unfortunately, it is limited in its distance coverage and may not cover rural areas in countries like Canada and USA. Furthermore, DSRC is based on 802.11a and cannot handle high-speed mobility while maintaining an acceptable data rate. It also does not have the ability to hand over connections at high speed, which is essential for high-speed motion and cannot support high data rates. DSRC would probably be integrated with other technologies for use in the VANET if it is to be universally accepted. For example, we can use dynamic spectrum sharing for both DSRC and WiMax to improve the communication efficiency as well as the spectrum utilization to cover both the rural and urban areas [35].

WiMAX (Worldwide Interoperability for Microwave Access) [14] is a 4G equivalent technology that enables the delivery of last mile wireless broadband access. WiMAX technology [36] can eliminate the use of cables and can save costs when used in remote and rural areas. The technology is scalable and has a flexible frequency re-use scheme because it uses OFDMA (Orthogonal Frequency Division Multiple Access) technology. Unlike DSRC, WiMAX was designed to handle high data rates. It implements full MIMO (Multiple-Input and Multiple-Output) which is good for mobile and car applications by enhancing timely information delivery to save lives and improve quality of life. Many laptops in USA already support such interface even though its present price is still high. With the above observations on and comparison work with DSRC, one sees that WiMAX is the most cost effective approach by providing a data rate that can satisfy the needs of our vehicle camera users (low latency and high coverage) at high speed and at an affordable cost.

## 2.4. Higher-Layer Protocols for VANET

Real-time data has an arrival deadline to reach its destination, or else its information is no longer valid. TCP (Transport Control Protocol) [8] and UDP (User Datagram Protocol) [9] are protocols that have been used but were found to not be effective. TCP is best used for lossless data transfer as it ensures packet delivery via packet retransmission. This poses a problem for real-time traffic, as there is no time for retransmission [10]. UDP can be better for real-time video traffic, but unfortunately still cannot guarantee real-time nor loss performance because their flows cannot adapt their rates to congestion control mechanisms at the router [37]. They can provide the best-effort service only since they have no priority over TCP flows. Congestion will also lead to random jitter and eventually loss, as video traffic becomes prevalent in f VANET. Although in theory TCP or UDP should not be affected by the choice of physical transmission medium such as fiber optic, they often perform poorly in wireless networks [38] because their design and implementations were carefully optimized for wired networks. RTP (Real-time Transport Protocol) [11] is a protocol designed for end-to-end real-time transport of video traffic. However, it does not take many performance issues into account including channel outage, traffic congestion, the effect of overheads in uncast/multicast applications, and the end-to-end delay in a mobile environment.

In summary, the behavior of these protocols in the VANET environment are quite different as the wireless medium is prone to interference from forces of nature and manmade effects, and the situation is exacerbated when transmitter and/or receiver are moving.

Various works have been conducted on congestion control in wired environments [39], wireless environments [40,41] and VANET [42]. Congestion control protocols can also be classified into two main types [39]: window-based and rate-based. The window-based congestion control algorithms adjust the window-size according to the level of congestion detected in the network. Algorithms like RED (Random Early Detection) [43], its TCP variants (like Tahoe [44], NewReno [45]), and AIMD (Additive Increase and Multiplicative Decrease) [44] fall under this category. This type of control usually corresponds to the implicit congestion control as congestion is detected at the receiver through the observation of events such as packet loss. So many algorithms in this type of approach can cause fluctuations in the source sending rates, thus producing non-smooth throughput and low link utilization. These are evidently not effective for real-time video streaming [46] in practice.

Rate-based congestion control algorithms on the other hand, allow each source in the network to directly adjust its sending rate based on the congestion notification received, and usually from the router that knows the state of the network. TFRC (TCP Friendly Rate Control) [47] and API-RCP (Adaptive PI Rate Control Protocol) [39] falls under this category. API-RCP presents an effective way to calculate the source-sending rate based on the queue deviation, and is quite capable of producing steady queue length to meet the bandwidth requirement of a driving camera. The rate-based control usually makes use of explicit congestion control due to the nature of its operation (as a contrast to the implicit control discussed earlier). However, this type of control may not be effective when a vehicle is moving because constant updating of the router due to motion consumes more CPU resources. Eventually, the CPU cannot catch up with the events arising from high speeds.

Most of the algorithms mentioned previously are built on top of TCP and have only been tested in wired networks. In wireless networks, delay is caused mostly by wireless interference rather than network congestion. Quite often packets are stored in a buffer waiting for proper bandwidth allocation before they can be delivered. As mentioned earlier, in real-time traffic, buffering (which incurs delay) is not wanted. We can tolerate a poorer quality, but delay is unacceptable. Explicit congestion controls handle traffic more effectively by calculating the flow throughput at the router node, and then smoothing throughput and achieving high link utilization in real-time. However, they are not effective for practical purposes. This is because they are more tedious to implement; one has to take time to load the controller on every router, and constant updates are required for the router of the moving vehicles. This consumes more CPU resources, and may not catch up with the driving speed. We thus consider an automatic predictive source control to assist the explicit control. When P frames start piling up, we drop the even frames first and then the odd frames without waiting for control commands to come from the base station.

#### 3. Operation and Models of the VANET Testbed

Figure 2 is a simple example of a VANET consisting of cars communicating their video information with each other and with the Internet (IP cloud) via the RSUs, which can be implemented as a WiMAX-enabled Base Station. The server is to forward or to archive the streaming video from the cars. Each car is acting as a WiMAX Mobile Station (MS) capable of receiving, processing and forwarding data to the destination and consequently to the backhaul network via the WiMAX Base Station.

In our VANET testbed, each vehicle has the ability to communicate with any neighboring vehicles. Depending on the nature of the messages, they will either stay within the VANET or exit the backhaul network via the RSU. For instance, tailgating, collision and braking (from preceding cars) are warning messages that can remain within the VANET network, while detailed regional weather forecast, notification of traffic jams and collision are usually exchanged between the backhaul network and the VANET. In many video applications we envision that video messages from the vehicular camera are usually required to be forwarded from the point of interest (such as traffic congestion, road block, and accidents) to a control center in the backhaul network to aid various parties (such as traffic personnel, emergency agents, insurance companies) to respond to the situations more effectively. We shall refer to these types of traffic as "safety driving video" or simply "safety video" later on.

To implement a testbed for the study safety driving video traffic among vehicles, we shall provide the operations at different layers as well as different models used in the testbed in the following two sections.





## 3.1. Protocol Layers and Operations

With reference to Figure 2, an RSU (BS 1 or BS 2) has the capability of handling up to N cars simultaneously. Each car/MS (Mobile Station) is associated with the closest RSU in the coverage area. Safety driving video packets are routed and given priority by recognizing their service class and scheduling type. For example, the silver service class and the rtPS (real-time Polling Service) scheduling [17] are assigned to the safety video. For each service class, there is a maximum sustainable traffic and reserved traffic rates (e.g., 384 kbps for the silver service class). Each video packet arriving from the higher layer at a MS is expected to be within a size limit. Otherwise, it will be segmented before being encapsulated into a PDU (Protocol Data Unit) with the appropriate header information, and then transmitted. For example, video packets longer than 1500 B must be segmented to fit the SDU (Service Data Unit) for the silver (safety video) service.

The driving video traffic is generated from the application layer when an MS wants to transmit. We use UDP (User Datagram Protocol) as the transport layer protocol to cater for the end-to-end delivery of packets. RTP (Real Time Protocol) is used at the application layer to combat the possible unreliability in UDP. RTP is a real-time streaming protocol designed for streaming audio and video. In conjunction, we use RTCP (Real-Time Transport Control Protocol) for controlling, signaling and quality service monitoring.

The unique nature of driving video traffic is due to its burstiness and persistence. The size of each frame generated is dependent on its previous frames and the driving speed, and the exact time a frame is generated or received also depends on previous frames and road conditions. These two correlated uncertainties contribute to the high variable bit rate characteristics of driving video communication. Coupled with the real-time and accuracy requirement of our safety applications, these factors pose a

modeling challenge. Of the various video codec schemes that have been developed e.g., MPEG2/4, H.263/4, we have chosen the MPEG4 video compression for our vehicular cameras and studied its traffic model for our testbed. Our array-based MPEG encodes the input video into three types of compressed frames called the I, the Peven, and the Podd frames. I-frames are compressed using intra frame information only. Peven or Podd-frames are coded similarly but with motion compensation for even or odd lines respectively according to the previous I frame. Depending on the vehicle driving speed or congestion condition, Peven and Podd frames may be dropped alternatively, or altogether. In general, I–frames are 10 times bigger than P frames. After coding, the frames are arranged in a deterministic order, which is called GOP (Group Of Pictures). This pattern is not specified hence different coders may use different patterns for subsequent GOPs. A common feature, however, is that all video frames exhibit strong time dependent correlations among themselves. This gives rise to different queuing statistics and time-dependent rush hour congestion phenomenon. This is the subject of streaming traffic analysis and modeling later on, as well as the need for a delicate adaptive congestion controller design.

Each car has a WiMAX capability in transmitting (and receiving) at the physical and the MAC layers after receiving the safety video packets from (and delivering to) the network layer. At the MAC layer, the destination address is examined and the nearest RSU is chosen to handle the request. The packet is sent to that RSU, and further forwarded accordingly. In our investigation, the IP cloud in Figure 2 can be set to its default values and acts a router. The server is configured to accept packets generated by our simulator.

As suggested above, we shall use mobile WiMAX technology at the physical and MAC layers for our VANET. This is a revision of the fixed WiMAX IEEE 802.16e-2005 standard [48] and it provides functionalities such as Base Station handoffs, MIMO (Multiple Input Multiple output) transmit/receive diversity, and scalable FFT (Fast Fourier Transform) sizes. Table 1 lists some of the more relevant features. Although WiMAX can support high data rates providing up to 128 Mbps downlink and 56 Mbps uplink using its MIMO antenna techniques, we shall use SISO antenna technique, which supports up to 1 Mbps uplink and downlink. We use the WiMAX MAC architecture to define service flows that can be mapped into gradual IP sessions to enable end-to-end IP based QoS.

Standard	IEEE 802.16e-2005	
Multiplexing	OFDMA	
FFT size Scala	Scalable (512, 1024, 2048, etc.)	
Duplexing mode	TDD	
Modulation scheme	QPSK, 16-QAM, and 64-QAM	
Subcarrier spacing	10.94 kHz	
Signal bandwidths	5, 7, 8.75, and 10 MHz	
Spectrum	2.3, 2.5, and 3.5 GHz	
	Standard Multiplexing FFT size Duplexing mode Modulation scheme Subcarrier spacing Signal bandwidths Spectrum	

 Table 1. IEEE 802.16e characteristics.

Of the two-duplexing modes allowed by mobile WiMAX, we shall use the TDD (Time Division Duplexing) mode (the other is FDD (Frequency Division Duplexing) mode) as done in most deployments in order to accommodate the asymmetric VANET traffic. That is, our uplink traffic

requires more bandwidth than the downlink, due to the uploading of safety (not downloading entertainment video that the other researchers are focused on) driving video traffic to the RSU. The WiMAX technology uses OFDMA frames for multiple users to receive data from the BS at the same time thereby increasing bandwidth utilization. The frames are structured into UL (Uplink) and DL (Downlink) subframe as shown in Figure 3 [49].





All the MSes receive the frame control section of a DL subframe. The DL mapping field in the frame control section specifies the location and duration of DL data burst in the frame from a particular MS. Conversely; the UL mapping field specifies the location and number of time slots allotted in the frame for a particular MS to transmit its data on UL. All MSes listen to the control section in the DL frame, and receive or transmit data during the slots assigned to them by the BS [50]. As seen from Figure 3, this structure enables dynamic allocation of DL and UL resources to efficiently support asymmetric DL/UL traffic. A subchannel is allocated to each user there by reducing the channel interference in the frequency domain. OFDMA (Orthogonal Frequency Division Multiple Access) is the scheme used allowing multiple accesses to every user on our network.

## 3.2. Network Models

Figure 4 summarizes the various components of our Driving Video VANET OPNET model. It consists mainly of the video model and the VANET model. OPNET modeler then provides the platform for the communication model and allowed for the integration of the various components of the Driving Video VANET OPNET model and other considerations like the congestion controller. They are discussed in the following.



Figure 4. Driving video VANET OPNET Model structure.

# 3.2.1. VANET Model

The VANET model consists of the VANET mobility model for the movement of a vehicle, and a communication model for information exchange with neighboring MS and BS.

# 3.2.1.1. VANET Mobility Model

Of the four categories of VANET mobility models described previously, we adopted both the macroscopic and microscopic models to allow the modeling of vehicle-to-vehicle and vehicle-to-infrastructure interaction. VanetMobiSim is chosen to integrate these models in OPNET due to its flexibility to manipulate output files (by coding its output generator file to produce a format desired by our simulator). It also generates traces in a special Universal Trace Format [15] in the form of (*time node\_id pos\_X pos\_Y velocity acceleration*) where *time* records the simulated time; *node\_id* identifies the specific node; pos\_X and *pos\_Y* provide the X and Y coordinates of the specific node at the given time respectively; and the *velocity* and *acceleration* show respectively the velocity at which the node moves and its acceleration at that time. With these capabilities, VanetMobiSim can model traffic lights, stop signs, human mobility dynamics and safe inter-distance management. It allows the integration of road topologies from databases such as GDF [51] or TIGER [52] to user-defined topologies as well as random topologies. VanetMobiSim provides a flexible platform in which the user can configure the shortest path during a trip based on the criteria of Dijkstra, road speed and road density. The trip can either be generated by random source-destination or according to activities [15].

# 3.2.1.2. RSU and Vehicle Communication Model

The RSU and the vehicles are the major communication nodes in VANET. Our RSU is a WiMAX BS. Each RSU is non-application-sensitive so that a wide range of information can be sent and received. Each vehicle has WiMAX receiving and transmitting capabilities to enable vehicle to vehicle and vehicle to infrastructure (RSU in our case) interaction. Table 2 shows the basic essential characteristics of our model along with some typical settings.

Parameter	Value	Unit
Physical Layer	IEEE 802.16e	NA
BS TX Power	5	W
Number of TX	SISO	NA
BS Antenna Gain	15	dBi
Minimum Power Density	-80	dBc
Maximum Power Density	-30	dBc
Link Bandwidth	20	MHz
Base Frequency	5.8	GHz
Physical Layer Profile	OFDM	NA

Table 2. RSU parameters.

#### 3.2.2. Driving Video Model

The driving video model is used to analyze and characterize the video trace of a live driving scene captured by a vehicle camera. The results are then used to generate models for use in our simulator for real-time driving video communication in a VANET environment. The implementation of this wireless driving video traffic model allows us to handle a wide range of burstiness in terms of packet inter-arrival and packet size arising from the unique nature of wireless video traffic during driving as mentioned above. A guideline to check the efficiency of a traffic model is, however, needed as to set a standard to be followed. Parsimony, analytic, relative accuracy, flexibility, implement ability and absolute accuracy are factors listed in [53] showing the engineering tradeoffs among different traffic models.

Using these factors on a scale of 1 to 3, 3 being the highest, Table 3 shows the settings of a few popular models including the FBM (Fractional Brownian Model) and the OPNET-emulated TCP traffic. By comparison, we adopt the Mini-Pareto model to be used later due to its overall score. In particular, we have taken a systematic approach in developing our mini-Pareto model. Driving video traffic traces were collected using the same camera used for a road test. In our testbed, we traced the frames from the receiver instead of the transmitter in order to correspond to video traveling from vehicle to vehicle, and through the RSU to the city's road monitoring center in a safety network. The traces were analyzed and stochastically represented and plugged into our simulation platform. A detailed discussion about these steps is presented in Section 4.

Items	FBM	<b>TCP Model</b>	Mini-Pareto
Parsimony	3	1	2
Analytical	1	1	2
Flexibility	1	1	1
Implemental	2	1	3
Accuracy	2	3	2
Mobility	2	1	3

Table 3. Traffic model comparison.

#### 3.2.3. Traffic Controller Model

As discussed in Sections 1 and 2.4, we only need to survive congestion to meet the QoS requirement of our real-time safety application scenario. This has given us the idea of reducing the complexity of the explicit control algorithm in order to cut down the CPU consumption when the vehicular camera is moving at higher speeds. Our careful study shows that reducing the sending bit rate is the key factor of success, and this can be achieved in two ways: dropping some frames (packets) or reducing quantization level per frame. There are different consequences and tradeoffs.

For driving video traffic, the drop pattern is very important to the quality of service as seen by user. Dropping just the even frames or just the odd frames is usually better than dropping both consecutively. The frequency of a drop can be a more critical factor than the percentage of drop. For instance, dropping 10 consecutive packets out of 100 packets is a 10% drop, which is the same as randomly dropping 1 out of 10 packets from a stream of 100 packets. However, the latter is more tolerable than the former in our system. Since there are many more P frames than I frames, we only drop P frames, first even then odd in our testbed. Thus we can use an automatic predictive source control to assist the explicit congestion control. That is, when P frames start piling up, we drop the P-frames by following a particular drop patter. For example, dropping the even frames first and then the odd frames without waiting for control commands to come from the base station. We can also set an upper limit of approximately 25% beyond which the video becomes less comfortable to watch (and may not satisfy customer expectations).

According to our industrial partners, changing the quantization level will be the next practical and cost-effective approach to reduce bit rate in terms of complexity and ease of deployment. We lower the quantization level when we use fewer coefficients from the coarse DCT (Discrete Cosine Transformation) algorithm, and thus reducing the packet size as well. Depending on the end-to-end delay, the receiver notifies the sender of the quantization level to use in encoding the subsequent video frames. This method will bring the total bit rate down to 50% of the original coding/sending rate. Beyond 50%, we need to drop the color components, thus leaving the video in its black-and-white mode.

In summary, the sending bit rate = frameRate\*FrameSize which can be lowered by lowering one or both of the frame rate or frame size. There are tradeoffs between the two. Dropping packets (frames) will lower the frame rate while reducing the quantization level that will lower the resolution. The challenge in lowering the resolution is to keep the user experience/satisfaction high as long as possible, and various experiments have been conducted to find out the optimum point. Although our real GenieView camera has implemented in both methods in firmware to improve user experience, our OPNET model has chosen to drop the frames/packet only. This is reflected in Figures 5 and 6 below where the called party collaborates with the calling party to determine the end-to-end delay based on how our controller uses one byte to notify the source to reduce its sending rate by dropping selected frames as discussed above. We also make use of RTP/RTCP at the application layer to provide an effective feedback mechanism.

Figure 5 depicts the video called party (receiver) procedure. Once the variables are initialized successfully, and a connection is established, data transmission can begin upon request. As soon as a frame is received, the frame count is incremented until it reaches the threshold value p. Once p is reached (e.g., p = 5 in our experiments), the frame count is reset to zero and the end-to-end delay

associated with the packet is checked. If the end-to-end delay is less than the threshold value t set by the end user on the receiving end (e.g., 100 ms), a default/designed rate of  $x^2$  is fed back to sender via the ACK packet. Else a lower rate  $x^1$  is returned instead. The  $x^1$  value can be a default value such as half of  $x^2$ , or computed according to some formula)

Figure 5. Video called party flow diagram with RTCP implementation.



Figure 6 shows the flow chart of the controller operation controller at the calling party manager (sender). Once the initialization process is successful, the video characteristics/properties are collected and a connection is established with the receiver. For every packet received, the algorithm checks if it is a control ACK packet. If not, the required statistics on packet type are collected and the packet is destroyed. If the packet is an ACK packet, the rate is retrieved and used to determine whether the sender should use the normal sending rate  $x^2$  or the reduced rate of  $x^1$ . The parameters for the call establishment are also updated according to the new value obtained.



Figure 6. Video calling party flow diagram with traffic controller implementation.

# 4. Experimental Setup

By collecting and studying video traces we want to come up with a model that can generate video traffic with characteristics as close to real life traffic as possible. This section discusses on the setup of our simulator testbed illustrates the trace collection process, its analysis and modeling. This will set the stage for traffic generation and performance evaluation in the next section.

# 4.1. Trace Collection

As shown in the setup of Figure 7, we use the monitor to play a series of driving video clips at 10 min each from driving action movies. Each video was captured by a PTS (Pan Tilt Zoom) camera taken from a vehicle. The video is encrypted and highly compressed at a ratio of 250:1 at the transmitter block. Using one processor for each section of the screen, the 96-array processors in the transmitter block can achieve fast independent image processing. The transmitter sends the highly compressed encrypted video images to the control center via a wireless link using the non-licensed 900 MHz FHSS radio-band. Please note that 900 MHz is more affordable to us presently than WiMax and DSRC. Since they are all wireless links, they will all be subjected to the same noise sources, such as EM interference. We only need to obtain similar type of interference and their statistics first to verify our platform.



Figure 7. Setup for taking video traces.

The receiver block decodes and decompresses the received video frames and plays the image at the control center at about 20 fps. The control-center laptop and the receiver are connected using a RS232 serial port. The control center is then connected through a router to a computer hosting the packet trace-capturing tool—Wireshark<sup>TM</sup> [54]. Once the system is turned on, the computer with the Wireshark<sup>TM</sup> software is set to access the "capture" folder in the control center which would then stream the video via a router for trace capture.

A video trace (a record sequence of time and size of each arriving packet) captured by Wireshark is shown in Figure 8. Due to the loss and duplication that will change its original characteristics, this type of traffic trace is closer to those from our application in the real environments. We want to characterize such a life driving video trace using proven stochastic processes and integrate it into a model that has as few parameters as possible but which can produce traces to match the real measurements.





In order to model video traces of different correlation/fractal characteristics arising from different activities, we collected traces playing three different video clips. The video clips were chosen based on their activity rates. Actions like highway chasing movies (with a lot of movement on the edge

of the screen/camera and hence unique variation in frame sizes), stories in a parking lot (with less and slower movement) and dramas on the street (on the average driving speed) were chosen. We combine these clips to form a two-hour video trace (consisting of the time and size of each arriving frame), which is sufficient to characterize the fractal nature of the driving video traffic as well as the wireless channel effect.

### 4.2. Trace Analysis and Modeling

The next challenge is to analyze the trace and to create a driving video traffic model for traffic generation later. To verify the correlation of video frame and to obtain the modeling parameter, we obtain the frame-to-frame correlation of the video traffic. Figure 9 is an example using our video trace, which shows correlation is present up to 20 seconds apart.



Figure 9. Correlation length of driving video.

We have used the Pareto distribution to characterize the general correlation of "action" video frame duration in wireless video streaming as well as the strong correlation between a cluster of byte loss and/or duplication caused by the air interface. The Pareto distribution is a long-tailed distribution with the following pdf (probability density function) g(x) for the consecutive I-frame duration, where the edge activity level is high. Note that *x* here is not the frame size but the group length rather.

$$g(x) = c\alpha^{c} / x^{c+1} \quad \alpha \leq x < \infty$$
<sup>(1)</sup>

where  $\alpha > 0$  is the location parameter and c > 0 is the shape parameter. The location parameter value can be obtained from the mean correlation length E(x) of the trace where  $E(x) = c\alpha/(\alpha - 1)$ . On the other hand, the shape parameter has to be obtained via the Hurst parameter. That is,

$$H = \frac{1}{2}(3-c) \implies c = 3-2H \tag{2}$$

Note that the Hurst parameter is a useful parameter to reflect the invariance according to the entropy conservation property [26]. Since our video is highly compressed with little loss of information, the entropy/uncertainty remains the same. Therefore, we can determine c from Equation (2).

An indirect but easy way to obtain the Hurst parameter value is via the *slope* of the IDC (Index of Dispersion for Count) curve as a function of the level of aggregation. Here IDC is defined to be

normalized variance of aggregated variables [26], and the level of aggregation refers the logarithm of the number of variables base-10 (therefore level-0 has 1 variable, level-1 has 10 variables, level-2 has 100 variables and so on). Mathematically, the relationship is given by [26]

$$H = \frac{1}{2} (1 + slope) \tag{3}$$

The slope of the curve is usually measured from the first point to the last point on the graph. For example, Figure 10 gives the IDC of the video trace we use. From this figure, the slope is measured to be 0.33. Hence the Hurst parameter H is 0.67, and the shape parameter is c = 1.66.



Figure 10. Entire trace IDC curve.

Once we obtain the pdf g(x) to characterize the activity duration (via the correlation of I-frames), we shall next determine the packet (frame) size distribution f(x), which is not Pareto in general. Furthermore, unlike the traditional video (e.g., video conference) with the moving part in the middle, our driving video has the moving part on the edge. Whereas the traditional video has a normal distribution in the middle for its packet size after 2D-DCT (Discrete Cosine Transform), the distribution for driving video is at the edge, which becomes very hard to model analytically to match the measured curve. After many trials, we have determined that an empirical model using nine mini-sources laid out in a grid can capture the distribution properly. The surrounding sources will capture the distribution of packets further away from the middle (edges and corners) which are usually moving and which produce large packets. The mini-sources closer to the center will capture the scenes in the middle which are relatively still and that produce smaller packets.

Figure 11 demonstrates the empirical analysis we performed on frame size (horizontal axis) of our traffic trace. For simplicity, here we are using the basic camera configuration, which produces one packet per frame. We use a tool called EasyFit<sup>TM</sup> [55] to obtain the pdf of the n = 9 video traces. The orange curve in the figure shows that the lognormal distribution is the best distribution to fit the given histogram data.



**Figure 11.** The pdf f(x) of video trace frames.

Finally, we want to determine the mean off-time of each source in order to model the inter-arrival time between each driving scene (such as trees or buildings on the road side). We have determined that the distribution is relatively memoryless, and therefore its pdf is exponential given by

$$a(x) = \begin{cases} \lambda e^{-\lambda x} & x > 0\\ 0 & \text{otherwise} \end{cases}$$
(4)

where  $1/\lambda$  is the mean inter-arrival time.

#### 4.3. Driving Video Traffic Generation

Having obtained all the characterization parameters for traffic generation in previous sections, we are ready to emulate our wireless video traffic. Figure 12 shows the schematics of our wireless driving video traffic generation model from the superposition of n mini-sources [56]. Each mini-source represents a video traffic stream regulated by a switch to give a long-range dependency. We shall model the on-time switch by a Pareto distribution given in Equation (1) and the off-time by a Poisson process [an exponential distribution as given in Equation (4)].

## Figure 12. Model description for driving video traffic generation.



Table 4 shows an implementation of using nine mini-sources, each representing one of the  $3 \times 3$  sectors of a camera screen. Here, frame duration is the transmission time of a frame (*i.e.*, frame size divided by bit rate). Assume the camera is facing the front when driving on a road. The middle is usually relatively still when compared to the sides that see more activities. The left turns, right turns, and uphills or downhills are all different scenes that can lead to special packet size distributions. Big packet size is produced by a fast side view of the camera while the small size is produced by the almost still middle view. In the characterization of each mini-Pareto source in the table, we use the same shape parameter obtained from the Hurst parameter, but the location parameter is different for different packet size bin.

Bins	Mean Off-Time (ms)	Mean On-Time (ms)	Frame Duration (ms)	Mean Packet Size (Bytes)
mini1	7531	195	8	125
mini2	20603	453	18	290
mini3	6911	702	28	450
mini4	5515	960	38	615
mini5	7800	1216	32	779
mini6	10400	1472	39	943
mini7	18200	1723	46	1104
mini8	27300	1984	53	1272
mini9	10400	2232	59	1430

 Table 4. Mini-sources with characterization values.

We have also tried other numbers such as  $n = 3 \times 4 = 12$  mini-sources to match the horizontal and vertical ratio of the camera. As verified by our experiments, the more mini-sources created, the higher the accuracy. However, the computational complexity also increases due to the more parameters needed to match the trace. This is translated into longer simulation time. So we have chosen n = 9 for our model for the best trade-off we found between parsimony and accuracy.

Figure 13 shows the video traffic generated mini-pareto video traffic, which can be compared to the live video traffic in Figure 8. Although not exactly the same, one can verify the simulated traffic captures the level of fluctuation of the driving video traffic.

## Figure 13. Mini-Pareto video traffic.



#### 4.4. OPNET Simulation Setup

Table 5 gives the values of the general parameters we used in our simulations. The terrain has an area of  $1300 \times 1250$  m. The relative position (x, y) on the terrain is used to integrate the VANET mobility model trajectories and obtain the initial positions of the vehicles. Vehicular environment for the path loss parameter is modeled according to the description in the "Radio Tx Technologies for IMT2000" white paper of the ITU [57]. The shadow fading standard deviation was set to 10 dB. The mobility pattern (trajectory) during the simulation is predefined using the *VanetMobiSim* discussed in Section 3.2.1.1.

Parameter	Value	Note
Physical Layer	IEEE802.16e	WiMax
Data Rate	10	Mbps
BS $T_X$ Power	5	W
MS Tx Power	1	W
Antenna Type	Omni-directional	Horizontal
BS Antenna Gain	15	dBi
MS Antenna Gain	9	dBi
Link Bandwidth	20	MHz
Modulation Scheme	16-QAM	Model
Path Loss Parameter	Vehicular Environment	ITU
Number of Vehicles	10	Cars
Mobility Model	VanetMobiSim	Open
Number of RSUs	2	Base
Simulated Time	3600	secs
Seeds	127	random
Terrain Dimensions	$1300 \times 1250$	$m^2$

Table 5. Simulation parameters.

We have investigated different scenarios such as highway and residential, and have studied the effect of varying traffic congestion (density), wireless interference and traffic speed limits. Due to space limitation, we shall present the highway scenario for this paper.

In the highway scenario shown in Figure 14, we consider cars moving along the highway with a minimum speed of 60 km/h and a maximum speed of 100 km/h. There is one traffic light, just before the cars enter the highway. The car trajectories are represented by the white lines. We use the free space loss model for our pathloss. Unless otherwise stated, the following are assumptions taken throughout the paper for our simulation study:

- (1) WiMAX BS is a "stationary" node. This is required due to the limitation of our OPNET model and we need it to act as an intermediate node for packet forwarding to the destination;
- (2) OFDMA is used by the RSU and there is always a slot available for each SS sending video traffic once the SS is within the communication range;
- (3) No disruption in a communication channel once the node is allocated a dedicated channel;
- (4) No obstacle along the LOS between the RSU and the mobile cars to allow the simulation to monitor the vehicles within the range of the RSU;

(5) Finite buffer size for each transmitter in order to be more realistic so that the trade-off between buffer size and end-to-end delay can be determined.



Figure 14. Highway scenario.

# 5. Performance Evaluation Using OPNET Testbed

With the simulation platform established, we can now evaluate the performance of a practical controller through measurements of different performance measures using different network parameters such as network size and driving speeds, and different protocol parameters.

There are two network sizes we use to evaluate the scalability of our explicit congestion control algorithm: a small network consisting of 10 cars and 2 RSUs, and a larger network of 30 cars and 2 RSUs. Each car in these networks has a maximum sustainable traffic rate of 384 Kbps. With reference to Figure 2, the link between the RSU and the backhaul network was a DS3 link with a capacity of 44.736 Mbps. Section 4.4 has also provided the values of other parameters.

Each simulation took 3600 s, which is determined to be a good duration for the performance measures to converge. The following are the definitions of the performance measures we used in our investigation:

- (1) End-to-End Delay (in seconds): This is defined as the time duration from the instant when a packet is generated at the source node (e.g., the safety camera on board a car) until the time it is received at the destination node. This performance measure presents a wholistic view of the time components in video capture, encoding and network delay before an end user visualizes the images. In real-time traffic, this parameter is very critical as it shows how "current" the information at the end user (e.g., a law enforcement officer) is. Since we track this at the application layer, the end-to-end delay also takes into account compression delay and all other delays incurred along the communication path and system;
- (2) Jitter: Jitter is defined to be the variation of end-to-end delay between consecutive packets. If two consecutive packets leave the source node with time stamps t1 and t2 and arrive at the

destination at time t3 and t4 after reassembly and playback, then the jitter will be represented by Equation (5).

Jitter = 
$$(t4 - t3) - (t2 - t1)$$
 (5)

Negative jitter means that the packets where received in different time range *i.e.*, (t4 - t3) < (t2 - t1);

(3) Visual MOS Rating: This rating is a subjective parameter that measures the quality of video received by the server, judging the end-user experience. This measure was borrowed from the same parameter used to measure the MOS (Mean Opinion Score) in voice. Using the definition given by the ITU-T recommendation G.107 [58], the score R is given by

$$R = Ro - Is - Id - Ie + A , (6)$$

where *Ro* is the SNR (Signal to Noise Ratio); *Is* is the simultaneous impairment factor; *Id* represents impairment caused by delay; *Ie* is the equipment impairment factor as a result of low-bit codecs and packet loss and *A* is the advantage factor that compensates for the loss cause by other factors.

To customize this parameter for video evaluation, we have created a testbed case with all the "ideal" conditions. We allocate one base-station to one car so that there is more than sufficient bandwidth, hence keeping R as high as possible. We obtained a Visual MOS value of 3.6 that we should use as a reference to represent the best visual experience and 1 the worst visual experience, which is plot as the Y-axis in Figure 15.

(4) Buffer Overflow Percentage: This measure is the fraction of time during which arriving packets find the buffer at the Subscriber Station full and are not served. This measure implicitly reflects the percentage of time video packets lost due to buffer overflow.



#### Figure 15. MOS value for a highway scenario.

# 5.1. Small Network

We only use 10 cars in this small network. After 10 s into the simulation, the WiMAX bandwidth is observed to vary from 125 kHz to 20 MHz (due to the adaptive modulation used as specified in Table 5). At the same time, we observe the video bandwidth also varies randomly with a peak of about 200 kbps and with an average of about 50 kbps at the Subscriber Station. Under this peak to average ratio of

6 dB, we set the controller threshold to 0.1 s in order to keep the controller active most of the time and to verify the operation of our congestion control algorithm. Below are more discussions of various performance measures.

#### 5.1.1. Time Evolution Performance

Figure 16 shows the highly varying end-to-end delay, which can be attributed to the wireless interference and the high speed of the vehicular nodes in the highway. One can see that the controller is always active in reducing the end-to-end delay (blue solid curve) but not always to the minimal point. This is due to the coupling effect of the car speed and periodic feedback so that some packets go "unnoticed", thereby losing it effectiveness.

MOS value for the highway scenario is shown in Figure 15. The score reflects the effect of fluctuations in end-to-end delay on the quality of video seen by the user. The controller tends to keep the visual experience at a constant rate, and improves when the end-to-end delay is reduced substantially.

Figure 17 shows the effect of the controller on the queue size. By controlling congestion in the network, our controller is able to reduce the occasions of buffer saturations and therefore the loss of video quality due to packet loss.





Figure 17. Instantaneous queue size for highway scenario.



The mean end-to-end delay performance for the highway scenario is shown in Figure 18. As expected, as the service rate increases, the end-to-end delay reduces. Increasing the service rate reduces the end-to-end delay and increasing the buffer size increases the end-to-end delay.



Figure 18. Mean end-to-end delay performance for highway scenario.

#### 5.1.2. Buffer Overflow Percentage

The percentage of buffer saturation of the highway scenario is shown in Figure 19. As expected, the percentage of time for which the buffer is full decreases with respect to increasing buffer size. Note also the larger bandwidth also reduces the percentage as packets can be served faster.





#### 5.2. Large Network

We next study a larger network with 30 cars. All other parameters and scenario remain the same as the small network unless otherwise stated.

#### 5.2.1. Time Evolution Performance

Figure 20 shows the end-to-end delay for the highway scenario in the large network. The solid line shows the end-to-end delay with controller implemented and the dotted line shows the end-to-end delay without the controller. The behavior seen here is similar to what was seen in Figure 16 except that the end-to-end values are much higher in this case due to an increase in the network size (longer path length and queuing delay, *etc.*). The controller again helps to reduce the end-to-end delay in this case, although its effectiveness is influenced by other factors, especially the speed of the car.

Figure 21 shows the MOS value for the highway scenario for the large network. Similar to the small network, the effect of the controller on the quality of service as seen by the user is not obvious as the end-to-end delay and their fluctuations are still high in this scenario.

Figure 22 shows the maximum queue size is reached more in this case due to the increase in network size. This also contributes to the visual experience discussed earlier.





Figure 21. MOS value for highway scenario (large network).



Figure 22. Instantaneous queue size for highway scenario (large network).



# 5.2.2. Mean End-to-End Delay

Figure 23 shows the mean end-to-end delay performance for the highway scenario in the large network for varying buffer sizes and service rates. Observations are similar to those of Figure 18 and hence the same discussion applies, except the mean end-to-end delay is higher in this case due to the increase in network size.

Figure 23. Mean end-to-end delay performance for highway scenario (large network).



## 5.2.3. Buffer Overflow Percentage

The percentage of buffer saturation performance curve for the highway scenario is shown in Figure 24 for the large network. It is seen that as the percentage of buffer overflow reduces as the buffer size increases and the service rate increases. This behavior is consistent with what is presented in Figure 19 except that the percentage of buffer saturation is higher in this case since the network size is increased.





## 6. Conclusions

We have created and evaluated an effective driving video VANET simulation platform with which other researchers can develop and test various VANET applications. The platform was built using the OPNET simulation tool to integrate our theoretical driving video model with a VANET model. We have built a simple but practical traffic controller for driving video traffic over RTP applications, and provided a complete tier of communication layers for proper performance analysis.

With the initial work done, there is still a lot of room for improvement and for exploring new model futures. Concurrent simulation using the simulation model and the mobility model would allow the wireless communication to affect the mobility of vehicles more directly so that closer reality results can be obtained. The traffic controller can be improved to keep the end-to-end delay closer to constant in order to reduce the quality impact on the viewers. Others include the comparison of our traffic controller under DSRC, and the validity of the Rician/Raleigh propagation model for moving vehicles.

## Acknowledgments

The research was supported by Auto21 Center of Excellence in Canada. Thanks go to Botao Zhu and Xiaoxiao Liu for reviewing and comments.

## **Conflicts of Interest**

The authors declare no conflict of interest.

# References

- IEEE P802.16-REVd/D4, Air Interface for Fixed Broadband Wireless Access System. Available online: http://hunter.hosted.pl/tts/Strony\_www\_TTS\_2004.2005/LMDS\_\_Marcin.Gilewski\_ Lukasz.Waksmanski/airInterface.pdf (accessed on 17 July 2012).
- Ramage-Morin, P.L. *Motor Vehicle Accident Deaths*, 1979 to 2004; Component of Statistics Canada Catalogue No. 82-003-X Health Reports; Statistics Canada: Ottawa, Ontario, Canada, 2008. Available online: http://www.statcan.gc.ca/pub/82-003-x/2008003/article/10648-eng.pdf (accessed on 17 July 2012).
- 3. Transport Canada. Canadian Motor Vehicle Traffic Collision Statistics: 2007. Available online: http://www.tc.gc.ca/eng/roadsafety/tp-tp3322-2007-1039.htm (accessed on 17 July 2012).
- 4. Heyman, D.; Lakshman, T. What are the implications of long-range dependence for VBR-video traffic engineering? *IEEE/ACM Trans. Netw.* **1996**, *4*, 301–317.
- 5. Ryu, B.; Elwalid, A. The importance of long-range dependence of VBR video traffic in ATM traffic engineering: myths and realities. *ACM SIGCOMM Comput. Commun. Rev.* **1996**, *26*, 3–14.
- Li, X.; Huang, H.; Shu, W. VStore: Towards Cooperative Storage in Vehicular Sensor Networks for Mobile Surveillance. In Proceedings of the IEEE Wireless Communication and Networking Conference (WCNC), Budapest, Hungary, 5–8 April 2009; pp. 1–6.
- Huang, J.; Qian, H. Video Encryption for Security Surveillance. In Proceedings of the 41st Annual IEEE International Carnahan Conference on Security Technology, Ottawa, Ontario, Canada, 8–11 October 2007.

- 8. Jacobson, V.; Braden, R.; Borman, D. TCP Extensions for High Performance, May 1992. Available online: http://www.ietf.org/rfc/rfc1323.txt (accessed on 17 July 2012).
- 9. Postel, J. User Datagram Protocol 28 August 1980. Available online: http://www.ietf.org/rfc/ rfc768.txt (accessed on 17 July 2012).
- Balk, A.; Maggiorini, D.; Gerla, M.; Sanadidi, M.Y. Adaptive MPEG-4 Video Streaming with Bandwidth Estimation. In Proceedings of the 2nd International Workshop on Quality of Service in Multiservice IP Networks, Milano, Italy, 24 February 2003; pp. 525–538.
- 11. Schulzrinne, H.; Casner, S.; Frederick, R.; Jacobson, V. RTP: A Transport Protocol for Real-Time Applications. Available online: http://www.ietf.org/rfc/rfc3550.txt (accessed on 17 July 2012).
- Liu, B.; Khorashadi, B.; Du, H.; Ghosal, D.; Chuah, C.-N.; Zhang, M. VGSim: An integrated networking and microscopic vehicular mobility simulation platform. *IEEE Commun. Mag.* 2009, 47, 134–141.
- 13. Sommer, C.; Dressler, F. Progressing toward realistic mobility models in VANET simulations. *IEEE Commun. Mag.* **2008**, *46*, 132–137.
- Wegener, A.; Hellbruck, H.; Wewetzer, C. VANET Simulation Environment with Feedback Loop and its Application to Traffic Light Assistance. In Proceedings of the 2008 IEEE GLOBECOM Workshop, New Orleans, LA, USA, 30 November–4 December 2008; Volume 3, pp. 1–7.
- Fiore, M.; Harri, J.; Filali, F.; Bonnet, C. Vehicular Mobility Simulation for VANETs. In Proceedings of the 40th Annual Simulation Symposium, ANSS 07, Norfolk, VA, USA, 26–28 March 2007; pp. 301–309.
- D1.3.1—VanetMobiSim: The Vehicular Mobility Model Generator Tool for CARLINK. Available online: http://carlink.lcc.uma.es/doc/provis/D1.3.1-VanetMobiSim%20The%20vehicular% 20mobility%20model%20generator%20tool%20for%20CARLINK.pdf (accessed on 17 July 2012).
- 17. OPNET. Analyze and Troubleshoot Application Performance from a Network Perspective Available online: http://www.opnet.com/solutions/brochures/ACE\_Analyst.pdf (accessed on 17 April 2010).
- Tian, L.T. Research on Intelligent Transportation Systems in Taiwan. In Proceedings of the 27th Chinese Control Conference, 16–18 July 2008; pp. 18–23.
- Laskowski, M.; Friesen, M.R.; McLeod, R.D.; Ferens, K. Rapid Prototyping Vehicle-to-Infrastructure Applications Using the Android<sup>TM</sup> Development Platform. In Proceedings of the 2009 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST), Lille, France, 20–22 October 2009; pp. 273–278.
- Intel. Intel CTO Rattner Outlines Research Challenges in Home, Car and Network Energy Consumption. Available online: http://www.intel.com/pressroom/archive/releases/2010/ 20100413comp.htm (accessed on 17 April 2010).
- Lalonde, J.; Laganière, R.; Martel, L. Single-View Obstacle Detection for Smart Back-Up Camera Systems. In Proceedings of the 2012 Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, RI, USA, 16–21 June 2012.
- 22. Hoogendoorn, S.P.; Bovy, P.H. State-of-the-art of vehicular traffic flow modelling. *Proc. Inst. Mech. Eng.* **2001**, *215*, 283–303.

- 23. Zhang, S. Object Tracking in Unmanned Aerial Vehicle (UAV) Videos Using a Combined Approach. In Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, USA, 18–23 March 2005.
- 24. Harri, J.; Filali, F.; Bonnet, C. Mobility models for vehicular ad hoc networks: A survey and taxonomy. *IEEE Commun. Surv. Tutor.* **2009**, *11*, 19–41.
- 25. Beran, J.; Sherman, R.; Taqqu, M.S.; Willinger, W. Long-range dependence in variable-bit-rate video traffic. *IEEE Trans. Commun.* **1995**, *43*, 1566–1579.
- 26. Hong, S.H.; Park, R.; Lee, C.B. Hurst parameter estimation of long-range dependent VBR MPEG Video traffic in ATM Networks. *J. Vis. Commun. Image Represent.* **2001**, *12*, 44–65.
- 27. Haghani, E.; Parekh, S.; Calin, D.; Kim, E.; Ansari, N. A quality-driven cross-layer solution for MPEG video streaming over WiMAX networks. *IEEE Trans. Multimed.* **2009**, *11*, 1140–1147.
- 28. Krunz, M.; Tripathi, S.K. On the characterization of VBR MPEG streams. ACM Sigmetrics Perform. Eval. Rev. 1997, 25, 192–202.
- Matrawy, A.; Huang, C.C. MPEG4 Traffic Modeling Using the Transform Expand Sample Methodology. In Proceedings of the 2002 Gaithersburg Proceedings 2002 IEEE 4th International Workshop on Networked Appliances, Gaithersburg, MD, USA, 15–16 January 2002; pp. 249–256.
- Yang, S.; Daigle, J. A Source Model of Video Traffic Based on Full-Length VBR MPEG4 Video Traces. In Proceedings of the Global Telecommunications Conference, St. Louis, MO, USA, 28 November–2 December 2005; Volume 2, p. 5.
- Luo, J.; Hubaux, J.-P. A Survey of Inter-Vehicle Communication; EPFL Technical 1618 Report IC/2004/24; École Polytechnique Fédérale de Lausanne: Lausanne, Switzerland, 2004.
- Gray, D. Mobile WiMAX Part I: A Technical Overview and Performance Evaluation v2.8. April 2006. Available online: http://www.wimaxforum.org/techno-Logy/downloads/Mobile\_WiMAX\_ Part1\_Overview\_and\_Performance.pdf (access on 17 July 2007).
- Jiang, D.; Delgrossi, L. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. In Proceedings of IEEE Vehicular Technology Conference (VTC) Spring, Singapore, 11–14 May 2008; pp. 2036–2040.
- Tufano, D.R.; Knee, H.E.; Spelt, P.F. In-Vehicle Signing Functions of an In-Vehicle Information System. Available online: http://www.osti.gov/bridge/product.biblio.jsp?query\_id=3&page= 0&osti\_id=219502 (accessed on 17 July 2012).
- 35. Hossain, E.; Chow, G.; Leung, V.; MacLeod, R.D.; Misic, J.; Wong, V.W.S.; Yang, O. Vehicular Telematics over heterogeneous wireless networks: A survey. *Comput. Commun.* **2010**, *33*, 775–793.
- 36. Ghosh, A.; Ghosh, M. Fundamentals of WiMAX: Understanding Broadband Wireless Networking; Prentice Hall: Upper Saddle River, NJ, USA, 2007.
- 37. Zukerman, M. Back to the future. IEEE Commun. Mag. 2009, 47, 36-38.
- 38. Tanenbaum, A. Computer Networks; Prentice Hall: Upper Saddle River, NJ, USA, 1996.
- 39. Hong, Y.; Yang, O. Design of adaptive PI rate controller for best-effort in the internet based on phase margin. *IEEE Trans. Parallel Distrib. Syst.* **2007**, *18*, 550–561.
- 40. Hac, A. Congestion control in a wireless network. Int. J. Netw. Manag. 1999, 9, 185–192.
- 41. Rath, H.K.; Sahoo, A. Cross Layer Based Congestion Control in Wireless Networks. Available online: http://www.it.iitb.ac.in/research/techreport/reports/8.pdf (accessed on 17 January 2010).

- Wischhof, L.; Rohling, H. Congestion Control in Vehicular Ad Hoc Networks. In Proceedings of the 2005 IEEE International Conference on Vehicular Electronics and Safety, Xi'an, China, 14–16 October 2005; pp. 58–63.
- 43. Floyd, F.; Jacobson, V. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.* **1993**, *1*, 397–413.
- Jacobson, V.; Karels, M.J. Congestion Avoidance and Control. In Proceedings of the SIGCOMM 88 Symposium on Communications Architectures and Protocols, Stanford, CA, USA, 16–18 August 1988; pp. 314–329.
- 45. Hassani, M.M.; Tavakolaie, B.H. An Analytical Model for Evaluating Utilization of TCP TAHOE Using Markovian Model. In Proceedings of the International Conference on Networking, Architecture, and Storage, NAS 2007, Guilin, China, 29–31 July 2007; pp. 257–258.
- Huang, C.; Xu, L. SRC: Stable rate Control for Streaming Media. In Proceedings of the IEEE Global Telecommunications Conference GLOBECOM 03, San Francisco, CA, USA, 1–5 December 2003; pp. 4016–4021.
- 47. Hanley, M.; Floyd, S.; Padhye, J.; Widmer, J. TCP Friendly Rate Control (TFRC): Protocol Specification. Available online: http://www.ietf.org/rfc/rfc5348.txt (accessed on 17 July 2012).
- 48. Li, K. *IEEE 802.16e-2005 Air Interface Overview*; WiMAX Solutions Division Intel Mobility Group: Santa Clara, CA, USA, 2006.
- 49. IEEE WirelessMan 802.16, The 802.16 WirelessMAN MAC: It's Done, but What is it? Available online: http://ieee802.org/16/docs/01/80216-01\_58rl/pdf (accessed on 17 July 2009).
- Ahmadi, S. Introduction to Mobile Radio Access Technology: PHY and MAC Architecture. Available online: http://vivonets.ece.ucsb.edu/ahmadiUCSB\_slides\_Dec7.pdf (accessed on 12 December 2009).
- 51. Ertico. GDF—Geographic Data Files. Available online: http://www.eritco.com/en/links/links/ gdf\_-geographic\_data\_files.htm (accessed on 17 July 2010).
- 52. U.S. Census Bureau. Topologically Integrated Geographic Encoding and Referencing (TIGER) System. Available online: http://www.census.gov/geo/www/tiger (accessed on 17 February 2010).
- 53. Huang, J. Presentation for Generalizing 4IPP Traffic Model for IEEE 802.16.3. Available online: http://www.ieee802.org/16/tg3/contrib/802163p-00\_58.pdf (accessed on 17 July 2012).
- 54. WireShark. Available online: http://www.wireshark.org (accessed on 17 July 2013).
- 55. Easyfit. Available online: http://www.mathwave.com/products/easyfit.html (accessed on 17 July 2009).
- Gu, H.; Ji, Q. An Automated Face Reader for Fatigue Detection, Automatic Face and Gesture Recognition, 2004. In Proceedings of the Sixth IEEE International Conference, Seoul, Korea, 17–19 May 2004; pp. 111–116.
- ITU-R, Guideline for Evaluation of Radio Transmission Technologies for IMT-2000, 1997. Available online: http://www.itu.int/dms\_pubrec/itu-r/rec/m/R-REC-M.1225-0-199702-I!!PDF-E.pdf (accessed on 17 July 2012).

 Mehmood, M.A.; Jadoon, T.M.; Sheikh, N.M. Assessment of VoIP Quality over Access Networks. In Proceedings of the First IEEE and IFIP International Conference in Central Asia on Internet, Bishkek, Kyrgyzstan, 26–29 September 2005; pp. 26–29.

 $\bigcirc$  2013 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).