

VInsight: Enabling Open Innovation in Networked Vehicle Sensing and Control

Yuehua Wang¹, Yu Chen¹, Chuan Li¹, Hongwei Zhang¹, Jayanthi Rao²,
TJ Giuli², Patrick Gossman³, Xiangying Yang⁴, Jing Zhu⁴

¹ Department of Computer Science, Wayne State University, {yuehua.wang, yu_chen, chuan, hongwei}@wayne.edu

² Research and Innovation Center, Ford Motor Company, {jr1, tgiuli}@ford.com

³ Computing and Information Technologies, Wayne State University, patrick.gossman@wayne.edu

⁴ Mobile Wireless Group, Intel Corporation, {xiangying.yang, jing.z.zhu}@intel.com

Abstract—Transforming the traditional single-vehicle-based safety and efficiency control, networked vehicle-sensing-and-control (VSC) is expected to be a basic element of addressing the challenges of safety and sustainability of road transportation. Yet vehicles remain largely as black-boxes to researchers and application developers, and there still lack open-source platforms for evaluating networked VSC solutions at high-fidelity and scale. To exploit the potential of networked VSC, we develop the research infrastructure *VInsight* that provides basic building blocks for gaining insight into vehicle internals and for enabling high-fidelity, at-scale evaluation of networked VSC solutions. More specifically, *VInsight* integrates the OpenXC vehicular sensing platform into the Global Environment for Network Innovations (GENI), *VInsight* deploys a WiMAX research network for integrating high-fidelity, real-world vehicular sensing with at-scale ns-3 emulation in the GENI cloud computing facilities, and *VInsight* develops WiMAX measurement tools for understanding WiMAX link characteristics and for driving high-fidelity emulation of networked VSC systems.¹ The WiMAX network of *VInsight* also enables immediate applications such as campus safety patrol; this synergy with opt-in users enables the permanent deployment of *VInsight* for open innovation in networked VSC systems, thus helping develop solutions to address the grand challenges of road transportation today.

I. INTRODUCTION

While being a basic enabler of our modern society, road vehicle transportation has become a major source of societal concerns. Vehicle accidents cause over 1.4 million fatalities and 50 million injuries per year across the world, and motor vehicles account for over 20% of the world's energy use and over 60% of the world's ozone pollution [1, 2]. Transforming the traditional, single-vehicle-based safety and efficiency control, next-generation vehicles will cooperate with one another and with transportation infrastructures (e.g., traffic lights) to improve transportation safety and efficiency. For instance, vehicles can coordinate with one another to avoid collisions just as how human avoid colliding with one another; based on real-time road and traffic conditions as well as the control actions of surrounding vehicles, a vehicle can control its speed,

throttle, and gear to improve fuel economy by up to more than 50% [3, 4].

To realize the potential of networked Vehicle Sensing and Control (VSC), researchers have been actively investigating a broad spectrum of problems such as vehicle behavior (e.g., fuel economy) modeling, vehicular wireless networking, and networked vehicle control. Several projects by the Car2Car consortium in Europe [5] and the CAMP consortium in USA [6] have also conducted field trials of V2X wireless communications as well as their applications in safety and mobility; USDOT has been operating several field testbeds for networked VSC [7], and the USDOT Safety Pilot program has experimented with active safety applications with close to 3,000 vehicles in Ann Arbor, Michigan [8]. While these efforts have helped gain insight into the problem space of networked VSC systems, vehicles remain as black-boxes to researchers and application developers in general. For instance, the existing field trials by the Car2Car and CAMP consortia as well as the large scale USDOT safety pilot deployment are not open to the general public and the broad research community. In addition, the current field trials and testbeds also focus on specific aspects (e.g., active safety) of networked VSC, and they do not support broad exploration of different potential aspects of networked VSC systems (e.g., joint optimization of vehicular wireless networking and vehicle control). Lack of access to ground-truth, real-world information about vehicle behavior makes it difficult to establish high-fidelity vehicle models (e.g., those on dynamics and fuel economy) and to develop field-deployable networked vehicle control algorithms. Lack of flexibility to support broad, open-source exploration of networked VSC, existing field trials and testbeds of networked VSC systems do not engage the broad community in developing innovative VSC solutions.

Due to the high cost of deploying and operating a large number of vehicles in field, another challenge of networked VSC research is the lack of a high-fidelity infrastructure for evaluating networked VSC solutions at scale. Even though there exist simulation systems that model vehicle mobility and wireless communication in an integrated manner [9], they do not model detailed vehicle behavior (e.g., real-world

¹The design and implementation of *VInsight* are readily extensible to LTE and other cellular technologies such as the emerging 5G networks.

fuel consumption), and they do not support multi-disciplinary exploration of networked VSC in general. One approach to enabling high fidelity, at-scale evaluation of networked VSC solutions is to integrate small-scale field-deployment of networked VSC platforms with large-scale, real-time emulation of networked VSC systems. The Global Environment for Network Innovations (GENI) [10] is an infrastructure for exploring next-generation networking and computing technologies, and a major part of GENI is a set of high-performance compute servers deployed across tens of campuses across USA. These GENI compute servers (also called *GENI racks*) are interconnected by high-speed (e.g., up to 10 Gbps) GENI VLANs, and they can support real-time, large-scale emulation of networked systems. Considering the parallel emulation capability of the ns-3 simulator, in addition, there exists the opportunity of integrating parallel ns-3 emulations in GENI racks with high-fidelity, real-time operation of networked VSC platforms in field as long as we interconnect the GENI racks with in-field vehicles.

To enable open innovations in networked VSC systems and to leverage the opportunities enabled by GENI, we develop the research infrastructure *VInsight* that provides basic building blocks for gaining insight into vehicle internals and for enabling high-fidelity, at-scale evaluation of networked VSC solutions. In particular, *VInsight* provides the following unique capabilities:

- *VInsight* GENI-fies OpenXC vehicular sensing platform by integrating it with the GENI ORBIT Management Framework (OMF) [11]. The GENI-fied OpenXC platform enables real-time sensing and archival of real-world vehicle behavior by the broad community. These real-time and archival sensing data serve as a basis for vehicle modeling (e.g., on fuel economy) in the development of innovative networked VSC solutions, and they also serve as realistic traffic input to evaluating VSC-oriented wireless networking solutions. The GENI-fied OpenXC platform is open-sourced and designed for easy integration with third party libraries to interact with hardware such as GPS and WiMAX dongles.
- *VInsight* deploys a WiMAX research network that covers the campus of Wayne State University (WSU), the highways around the university campus, as well as the majority of midtown and downtown Detroit where standard city roads exist. The WiMAX network connects field-deployed, GENI-fied OpenXC platforms with GENI racks, thus enabling the integration of in-field vehicle sensing with at-scale emulation in GENI racks. The WiMAX network also connects field-deployed police patrol vehicles and officers with hundreds of surveillance cameras on WSU campus, thus facilitating real-time, accurate emergency response. Given that WiMAX features advanced, widely adopted communication techniques such as OFDM and MIMO, the WiMAX network also enables studying wireless link characteristics which helps optimize vehicular Internet access as well as its

applications.

- *VInsight* develops GENI-fied WiMAX measurement tools that enable real-time measurement and archival of WiMAX link characteristics. The enabled understanding of WiMAX link characteristics helps develop networked VSC applications, and the real-time as well as archival link measurement data also serve as input to high-fidelity emulation of networked VSC systems as we have discussed above.

To the best of our knowledge, *VInsight* is the first infrastructure that enables open-access of vehicle internals as well as open-innovation in networked vehicle sensing and control. Unlike existing vehicular testbeds Car2Car, CAMP, DOT-testbeds, SafetyPilot, DRIVE, FleetNet, NoW, CarTalk, PATH, TELCO, and DOME, *VInsight* not only establishes and maintains publicly-available resources, it also enables integrating field-deployed vehicles with large-scale simulation, thus enabling high-fidelity experimentation in a wide range of scenarios. The ease of deployment and the high-fidelity of *VInsight* make *VInsight* readily replicable to other institutions, to extend the reach of VSC and to scale up federated VSC experimental infrastructures.

The rest of the paper is organized as follows. Section II introduces the system architecture of *VInsight*, Section III presents the GENI-fied OpenXC platform, and Section IV describes the WiMAX measurement tools. Finally, we make concluding remarks in Section VI.

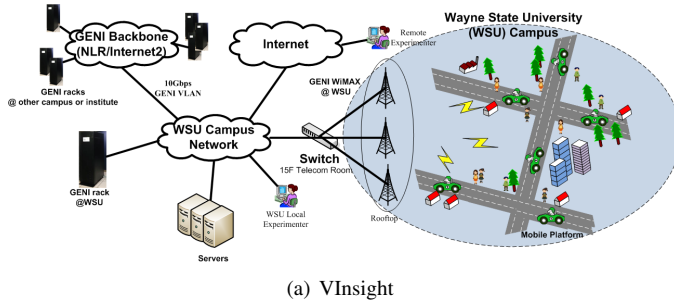
II. SYSTEM ARCHITECTURE OF VINSIGHT

We design and deploy *VInsight* as a research infrastructure that provides basic building blocks for sensing vehicle internals and for enabling high-fidelity, at-scale evaluation of VSC solutions. The overall design of *VInsight* and its building blocks are readily replicable to other institutions and testbeds to extend the reach of open innovation infrastructures for networked VSC systems. In what follows, we present the system architecture of *VInsight* and then describe its components in details.

As can be seen in Fig. 1(a), *VInsight* consists of three main components: the GENI WiMAX base stations, servers, and mobile platforms.

GENI WiMAX base stations. GENI WiMAX base stations (as shown in Fig. 1(b)) provide network researchers and users with wide-area coverage and the ability to support both mobile and fixed users. The base stations operate with an educational/experimental spectrum license; in addition to connecting the mobile platforms to GENI racks for high-fidelity, at-scale emulation of networked VSC systems, the WiMAX base stations provide wireless network services to opt-in users such as the campus police patrol vehicles of WSU. With low-cost WiMAX dongles, users within the coverage areas of base stations can easily access GENI services as well as the public Internet.

Fig. 2 shows the coverage of the base stations in terms of the RSSI (Received Signal Strength Indicator) and CINR (Carrier



(a) VInsight



(b) GENI WiMAX base stations

Fig. 1. System architecture and GENI WiMAX base stations

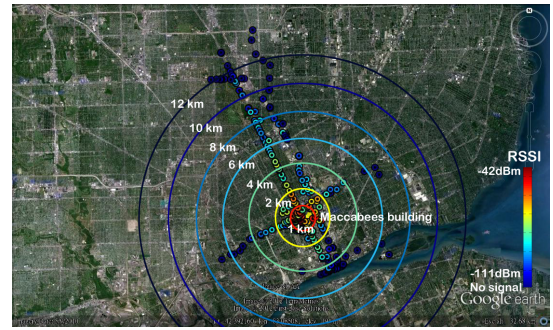
to Interference plus Noise Ratio) at different locations. We can see that the WiMAX base stations cover a wide area. It can reach up to 12 km north of the base stations on WSU campus.

Servers. To manage resource allocation and provide services for measurement data collection and management, a Netspan server and an OMF server are deployed in a lab of the Maccabees Building on WSU campus, and the servers run on a 24×7 basis.

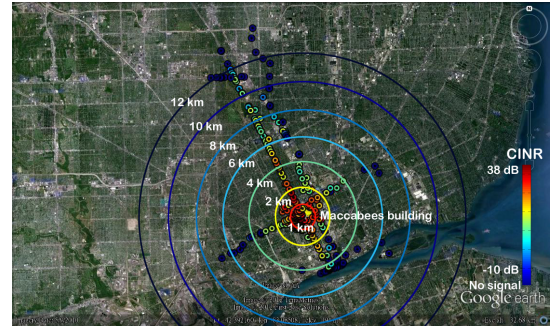
The Netspan server is set up for the configuration and management of the base stations and mobile platforms via SNMP (Simple Network Management Protocol). The software architecture of the Netspan server is a master/slave architecture as shown in Fig. 3(a). The master is composed of a SQL database server, an IIS web server, and a set of daemon services. The use of the SQL Database server is to store configuration files, statistics, and alarm history from the WiMAX network.

To facilitate resource allocation and experiment deployment, the OMF [11] server in Fig. 3(b) runs as a controller to take care of the implementation details of experiment and measurement data collection according to an experimenter's requirements. The Aggregate Manager (AM) manages all the resources that are provided by VInsight and consists of a set of services such as measurement collection, inventory, Frisbee, and slice manager. More details will be further described in Section III.

Mobile platforms. A fundamental unit of VInsight is the mobile platform. In VInsight, each mobile platform is a laptop equipped with a Teltonika WiMAX dongle with two build-in antennas, a GPS receiver, and a GENI-fied OpenXC vehicular sensing module (a.k.a. CAN reader [12]). The laptop has a 2.3 GHz Intel Core, 8 GB RAM, and a 500 GB hard disk for enabling access to vehicle internals and for understand-



(a) RSSI



(b) CINR

Fig. 2. Coverage of the GENI WiMAX base stations

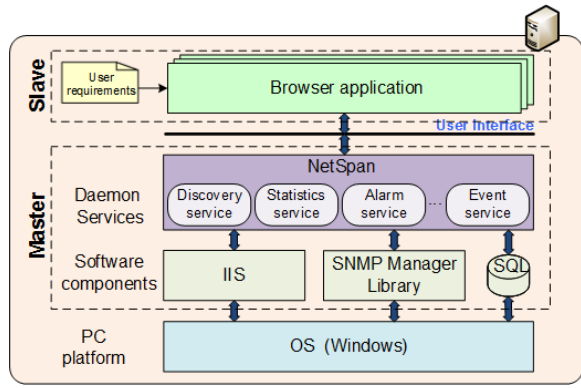
ing WiMAX link characteristics that help develop networked VSC applications. To ensure high-fidelity in VSC networking experiments, the laptop runs Linux Ubuntu 13.04 which is characterized by its software and hardware compatibility. In experiments, the laptops invoke the services provided by the GENI-fied OpenXC platform and WiMAX measurement tools, and act as agents to collect the measurement data from the connected devices (i.e., WiMAX dongle, GPS, and CAN reader) and deliver them to the OMF server, the experimenters, and/or the users. We will discuss the GENI-fied OpenXC platform and WiMAX measurement tools in Sections III and IV respectively.

III. GENI-FIED OPENXC PLATFORM

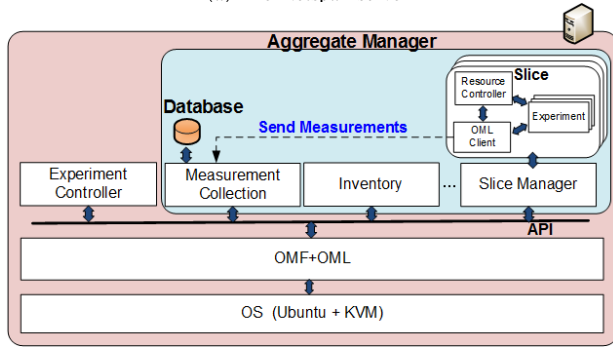
The GENI-fied OpenXC platform is a key building block of VInsight, which enables experimenters to sense and archive vehicle behavior in real-time. In what follows, we present the design objectives and software architecture of the GENI-fied OpenXC platform. To demonstrate the use of the GENI-fied OpenXC platform and VInsight in general, we will discuss its application in vehicle fuel economy sensing in Section V-A.

A. Software architecture

One of the challenges for open innovation in networked VSC is how to enable the broad community of researchers to interact with vehicle internals directly, for instance, sensing vehicle behavior in real-world settings for deriving models for vehicle fuel economy which can in turn assist the design of fuel economy optimization algorithms. In the existing deployments of networked VSC system, however, vehicles

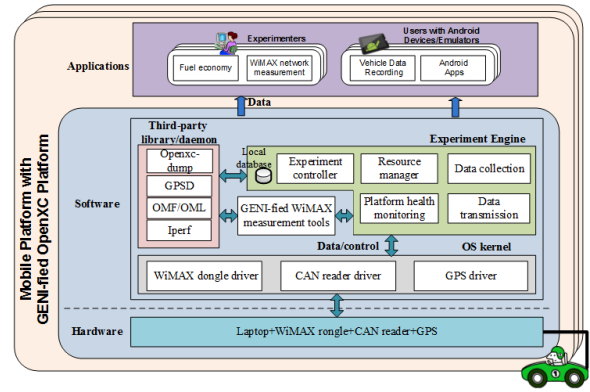


(a) The Netspan server

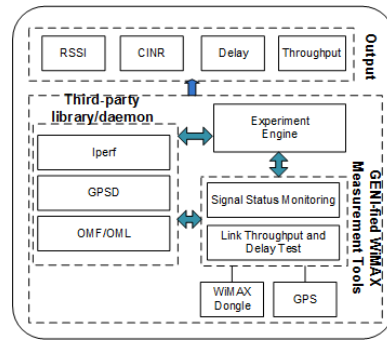


(b) The OMF server

Fig. 3. Architectures of the NetSpan and OMF servers



(a) GENI-fied OpenXC platform



(b) GENI-fied WiMAX measurement tools

Fig. 4. Architectures of two building blocks of VInsight

have remained as black-boxes to the broad community of researchers and application developers.

To address this issue, we develop the GENI-fied OpenXC platform as a major component of the VInsight infrastructure, aiming at enabling researchers/experimenters to access vehicle internal state in real-time. The implementation of the GENI-fied OpenXC platform follows the publisher/subscriber communication paradigm where the vehicles serve as data sources while either an experimenter, the OMF server, or both act as data subscribers.

B. Design and implementation

The GENI-fied OpenXC platform consists of the resource manager, experiment controller, data collection, data transmission, platform health monitoring, and local database. On behalf of the GENI-fied OpenXC platform, an experiment engine is employed as a manager responsible for managing the platform components and communicating with the hardware devices. The platform uses third-party libraries/daemons to interact with hardware devices such as the CAN reader, WiMAX dongle, and GPS, as shown in Fig. 4(a). Although each component of the platform is well-defined, using common and open-source libraries/daemons helps resolve many incompatibility issues. We provide a detailed description of how the different components of the GENI-fied OpenXC platform cooperate well with each other as follows.

Experiment engine. The design of experiment-engine (EE) is the most challenging part of the GENI-fied OpenXC platform.

The mobile platform must not only run experiments for researchers and opt-in users, it also needs to connect to the servers for experiment management and data transmission and archival. In OMF/OML, the OMF Resource Controller (OMF RC) is used to communicate with the Experiment Controller (EC) on the OMF server using a separate Control Network (CN) to minimize the impact of control command/reply transmission on the flow of the experiment traffic. However, this design comes at the cost of implementation complexity as any update or extension may imply a redesign of the system architecture or node configuration. In some cases, the node hardware might not have enough number of interfaces, or the system infrastructure required for a CN may not be available for the nodes (e.g., mobile platforms). It happens very often that a mobile platform moves to a location with poor WiMAX signal strength or moves out of the coverage range of the WiMAX network for a period of time. Under such conditions, the OMF server may be unreachable from the CN. This could pose two challenges. The first challenge is how to prevent an ongoing experiment from being interrupted by the interruption of communication with servers. The second challenge is how to eliminate the impact of the network measurement traffic on the experiment traffic. In [11], an OML-based proxy server is used to buffer measurements on the local mobile platform. But it remains unclear how the experiments are going to run without interactive communication to the server.

To address the challenges, we decouple the operations related to experiments from those of the OMF RC. In VInsight,

the EE runs as an experiment manager on each mobile platform, being responsible for resource management, experiment control and operation, data collection and transmission, and mobile platform health monitoring and management. The OMF RC in VInsight only coordinates with the EE to listen to commands from the EC.

Based on the ED, the EE configures the associated local resources and executes the experiment. When there are replies/outputs from the executions, the EE caches them first and then sends back to the EC when the WiMAX network is idle. For some urgent messages like software/hardware errors, EE stops the experiment and notifies the OMF RC to report to the EC.

Experiment controller. The EC is responsible for setting up, launching, and cleaning up experiments. In VInsight, when an experimenter submits an experiment to the OMF server, the AM will notify the EE to take over the control of the experiments. For each experiment, the EC creates a job based on the experiment description. All the jobs are queued by the EC and scheduled based on a priority-based policy, Fixed Priority Preemptive Scheduling (FPPS) with the consideration of resource utilization and job waiting time. At any given time, the EC selects and executes the highest priority job among all the jobs that are ready to execute in the queue. There are two main types of jobs: maintenance job and experiment job. Typically, the EC gives the maintenance jobs (e.g., resetting device or software building blocks, emergency stop) higher priorities than experiment jobs. This provides flexibility in optimizing resource utilization and in reducing experiment waiting time.

Resource manager. The resource manager administrates the usage of mobile platform resources, and it interacts with the EE for resource publishing when an experimenter interacts with the AM to reserve resources. With the experiment description, the resource manager reserves and configures the resources for experiments. All the resources, be physical or logical, are abstracted as components such as openxc-dump and WiMAX dongle. For short-term experiments, resource management such as discovery, reservation, and configuration may be performed at the beginning of the experiment in a one-shot fashion, assuming the underlying environment is relatively static in a short time window. However, for long-term experiments, resource management has to be an ongoing process to adapt to network changes. For example, the WiMAX dongle or CAN reader could crash or fail due to improper use.

Data collection. For the GENI-fied OpenXC platform, the EE invokes data collection if it is requested in the experiment description. Data collection uses the openxc-dump daemon to read OpenXC messages from the attached CAN reader and direct them to the EE for message processing and transmission. By default, each message is formatted using JSON (JavaScript Object Notation) which is a lightweight text-data interchange format and uses JavaScript syntax for describing data objects.

In VInsight, a message from the openxc-dump is considered

as a valid message if and only if it is in a pre-defined format. More precisely, a valid message must include at least the following fields: *timestamp* that has 10 bits and is the Unix time when the message is read; *name* that indicates the type of the vehicle sensing data, e.g., odometer, fuel level, engine speed, and steering wheel angle; *value* that denotes the value of the data specified by the filed *name*. *event* is an optional field and mostly used with the vehicle sensing data *door status*. Invalid messages are dropped by the EE.

Data transmission. It is responsible for delivering data to their destinations. It runs as a daemon and is immediately notified if any data is ready to be sent out after the EE has processed them. With regard to the diversity of the subscribers, VInsight provides two transmission methods to support the subscribers in collecting data and feeding the applications with those data. One is designed for the subscribers using OMF/OML. OMF/OML defines and implements the data transmission on behalf of the subscribers and the CAN reader. After data processing, the data streams are forwarded by the OML client to the OMF server. The OMF server stores them in a SQL database created for the experiment. The subscribers can run SQL queries to fetch the data of interest. This OMF/OML-based transmission method simplifies the design and implementation of the GENI-fied OpenXC platform, and it also provides the subscribers with the ease of experiment initialization and deployment as well as archiving vehicle sensing data.

The other transmission method is developed for the subscribers who do not use OMF/OML. In this case, the mobile platforms with GENI-fied OpenXC platforms act as “servers” and send the live data stream to the subscribers via TCP connections. To build the TCP connections, the IP addresses and ports of the subscribers should be specified in the experiment description. For archiving the data, the mobile platforms also send the data to the OMF server using OMF/OML.

Platform health monitoring. It is quite common for unexpected errors or failures to occur in the middle of an experiment, which may affect the quality or even the correctness of output data. Platform health monitoring is a component used to keep track of the hardware and software of a mobile platform and monitor their status. The real-time health monitoring information about the mobile platform is periodically collected and inserted to the job queue sorted by the priority mentioned earlier. If there is no job running on the mobile platform, platform health monitoring will be invoked to check if the hardware and software are in a good state.

Local database. In the platform, the database is used to store all the information necessary for running experiments, monitoring the platform, and analyzing experiments. More specifically, there are two categories of information stored in the database. One is related to the experiments, which includes the configuration files, starting/end time, input files, outputs, and replies/measurement data in response to requests of the EC. The other is about the mobile platform status. The mobile platform configuration, and the software and hardware status

are data of this type.

During the experiment, the replies/measurement data in response to requests of the EC are buffered on the mobile platforms. In this manner, the experiment traffic will not be influenced by that of other traffic so that the network utilization can be maximized. But this comes at the cost of additional delay due to buffering. We are currently working on a method of deploying a single WiMAX sub-channel for control and measurement transmission in the presence of environmental and network dynamics. Note that channels possess a variety of properties in different signal conditions. Techniques of channel estimation and channel allocation will be used in VInsight.

Toward effective storage utilization, the EE also provides some filtering functions, which allow the subscribers to compress the real-time data stream according to application requirements and resource utilization situation. Based on the experiment description, the data filters are created automatically and used without rewriting the openxc-dump code. It, hence, provides a flexible way of changing the data collection behavior for the subscribers with various requirements. Typically, the data filters are registered with the EE and define the data streams going to the subscribers. It supports 1) time-triggered filtering, where filters become disabled after a time period specified in the experiment description, 2) type filtering, where the data with certain types are filtered, and 3) event-triggered filtering, in which the filters are fired upon the occurrence of certain events.

IV. GENI-FIED WiMAX MEASUREMENT TOOLS

To enable real-time measurement and archival of WiMAX link characteristics, a suit of measurement tools is needed for purposes such as feeding link characteristics data to the emulation of networked VSC systems in GENI racks, monitoring link status, and collecting experimental results. Many factors affect WiMAX link properties: environment, distance to the base station, the number of Vehicles with Mobile Platforms (VMPs), locations of VMPs, diversity of VMPs, etc. It is difficult to isolate these factors in order to study the impact of different factors on the performance of the network. Rather, we strive to provide a tool to help measure WiMAX link properties in a systematic fashion. In what follows, we elaborate on the architecture and components of our GENI-fied measurement tools.

A. Architecture and components

Fig. 4(b) shows the software architecture of the GENI-fied WiMAX measurement tools. It consists of two main components: signal status monitoring, and link throughput and delay test. The two components interact with the hardware WiMAX dongle and GPS using the third-parties libraries and daemons.

Signal status monitoring. It communicates with the WiMAX dongle, monitors the connection between the mobile platform and the base station, and measures the RSSI and CINR at locations selected for WiMAX network measurement. For example, if an experimenter would like to study the spatial

behavior of links between the mobile platforms and the base stations over the WiMAX network, line-of-sight areas, local urban areas, and freeways may be candidate measurement locations.

Link throughput and delay test. Iperf is triggered by the EE when a link capacity measurement request is received from the EC. Using UDP, Iperf on the mobile platforms of VInsight runs for uplink and downlink capacity estimation. There are several types of services (i.e., UGS, rtPS, nrtPS, and BE) supported by the WiMAX base stations. By default, the BE service type is configured as the default in VInsight since BE incurs lower overhead than other service types. Experimenters can customize service types based on the requirements of the experiments or the applications. We refer the reader to a use case of the GENI-fied WiMAX measurement tools in Section V.B.

B. Discussion

When VInsight was designed and deployed as a part of the NSF GENI program, LTE was not available and WiMAX was the only 4G cellular technology available for purchase and installation. The design and implementation of VInsight is such that, when equipment for LTE and other cellular technologies (e.g., the emerging 5G) become available, they can be integrated into VInsight. Thus VInsight is readily extensible to LTE and other emerging cellular technologies². Even though the traditional consumer-electronics market has favored LTE over WiMAX, using WiMAX in VInsight is also of value. Technically, WiMAX and LTE use similar air interface technologies such as OFDMA (Orthogonal Frequency Division Multiple Access) and MIMO (Multiple-Input and Multiple-Output), and WiMAX and LTE have comparable spectral efficiency up to the first order of approximation [13]. The WiMAX network enables us to understand properties of MIMO-OFDMA wireless links, which can shed light on LTE networks. On the campus of Wayne State University, the WiMAX network connects field-deployed, GENI-fied OpenXC platforms with GENI racks and connects police patrol vehicles with infrastructure-mounted surveillance cameras, thus enabling the integration of in-field vehicle sensing with cloud-based emulation and enabling real-time campus safety surveillance.

V. USE CASES

VInsight offers a flexible and extensible research infrastructure for experimenters and users to evaluate their ideas or solutions in VSC networks. Its capabilities of supporting vehicle internal sensing, long-distance communication, and the use of GENI compute servers make it suitable for a wide variety of experiments and applications in VSC networks. To illustrate this, three example use cases are presented as follows.

A. Fuel economy sensing

Fig. 5(a) shows a use case of VInsight for fuel economy sensing. It illustrates how the filtered real-time vehicle internal

²In fact, we are currently planning the deployment of additional LTE base stations into the VInsight infrastructure.

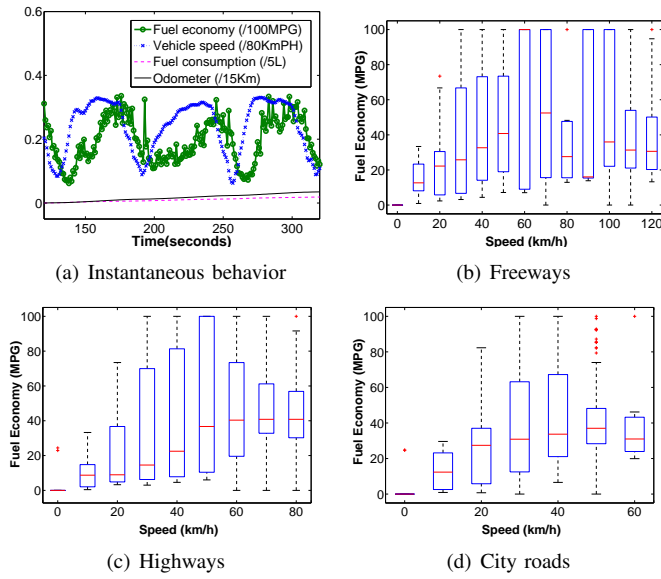


Fig. 5. Fuel economy sensing via VInsight

data (i.e., odometer, fuel level, and speed) collected on the campus of WSU can be utilized by remote experimenters. In this figure, the values of the vehicle internal state are updated during the experiment, and the instantaneous fuel economy is updated after periodic calculation based on the data collected through VInsight. The instantaneous fuel economy is the measurement of fuel usage of the vehicle over the distance traveled during the sampling period.

Fig. 5(b), 5(c), and 5(d) show the boxplot of the fuel economy of the vehicle with different speeds on the roads. We drive the vehicle across different types of roads and the vehicle status data was collected. For each experiment only one vehicle is used, and the roads of three main types around the campus of WSU are traveled. They are freeways (i.e., I-75 and I-696), highways (i.e., M-102, M-1), and city roads (i.e., Woodward Ave, Warren Ave, Cass Ave, and 2nd Ave). On the freeways, there are no stop signs nor traffic lights. Cross traffic is prohibited and speed limits are usually 88 or 112 km/h. Highways are usually rural roads where cross-traffic, stoplights, and stop signs are allowed and the speed limits are variable. Urban roads are the normal roads in the midtown of Detroit where the speed limit is usually 40 km/h and is, very occasionally, 48 or 56 km/h.

From the results, we observe the following: 1) The vehicle fuel economy is not a static indicator provided by the motor companies or authoritative organizations [14], but a dynamic parameter highly related to the instantaneous distance traveled by a vehicle, the fuel consumed by the vehicle, and the current speed of the vehicle; 2) Under certain conditions, the increase of speed may reduce the fuel economy, as shown in the dash line circles; 3) The same speed may lead to very different fuel economy due to the road and traffic dynamics; 4) The changes of vehicle fuel economy lags behind that of vehicle speed. The main reason behind this phenomenon is due to the fact that the measurement accuracy of vehicle speed is higher

than the fuel consumption. In reality, it is common that there is no change in fuel consumption even though the vehicle has been traveling at a dynamic speed for a while; 5) We find that, for a given vehicle, the fuel economy increases with the speed until the speed reaches a certain threshold value after which the fuel economy decreases. In our test, the vehicle achieves better fuel economy at a high speed ranging from 60 to 70 km/h. We also see that, constrained by the speed limit and the traffic lights, the fuel economy on city roads and highways is lower than that on freeways, as shown in Fig. 5.

B. Link property measurement

The example use case in this section concerns network performance. We implement several measurement mechanisms, which span from RSSI and CINR to the characteristics of UDP connections to determine the available throughput and practical link delay. Over 150 different measurement locations are selected as measurement points for network performance measurement via VInsight.³

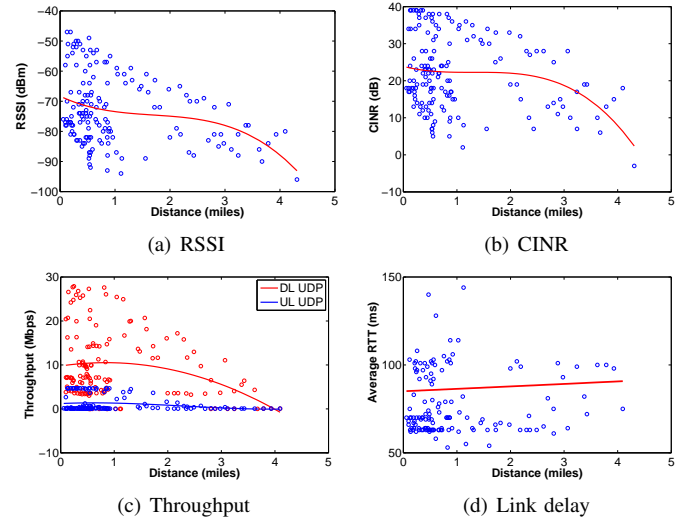


Fig. 6. WiMAX link measurement results

Fig. 6(a) and 6(b) show the measured RSSI and CINR at different distances in VInsight respectively. From the fitted curve, we can see that both RSSI and CINR decrease as distance increases. The relatively large variation in RSSI and CINR within 1 mile is due to the fact that the density of buildings in certain area of the campus is high. While we observe that the data collected at neighboring measurement points is relatively close, we also find that the position of the WiMAX dongle affects the performance in experiments due to the presence of obstacles (e.g., seats and passengers) either on the vehicle or in the vicinity (e.g., pedestrians and other vehicles).

Fig. 6(c) shows the throughput of a mobile platform as a function of distance. We see that the downlink has a higher mean throughput than the uplink, which is consistent with

³The measurement locations are publicly accessible at <https://github.com/VInsightDNC/VInsight/blob/master/Testpoints.txt>.

the design of the WiMAX network. The fitted curve shows that as distance increases, both the downlink and the uplink throughput decrease. However, the uplink throughput remains low and drops to zero quicker than the downlink throughput due to transmission power limit of the mobile platform.

Link delay test is used to measure the round trip time among mobile platforms and base stations. To measure RTT and its variation, the “ping” command is repeatedly issued to the base stations or other mobile platforms in VInsight. The measurement results of a mobile platform connected to a base station are shown in Fig. 6(d). In the link delay test, we conducted the Ping tests 100 times at each location using the default packet size (32 bytes), then we calculated the mean value. When the mobile platform is with lower signal strength, the RTT is relatively high due to signal attenuation and dispersion. In addition, we note that the variation of RTT is high even when the measurement locations have the same/similar distances to the base station. This is caused by environmental and network dynamics. Even though this example use case is not a comprehensive study on the performance of WiMAX networks, we expect VInsight and its GENI-fied WiMAX measurement tools to help experimenters understand link properties with important real-time, real-world information.

C. Real-time emulation

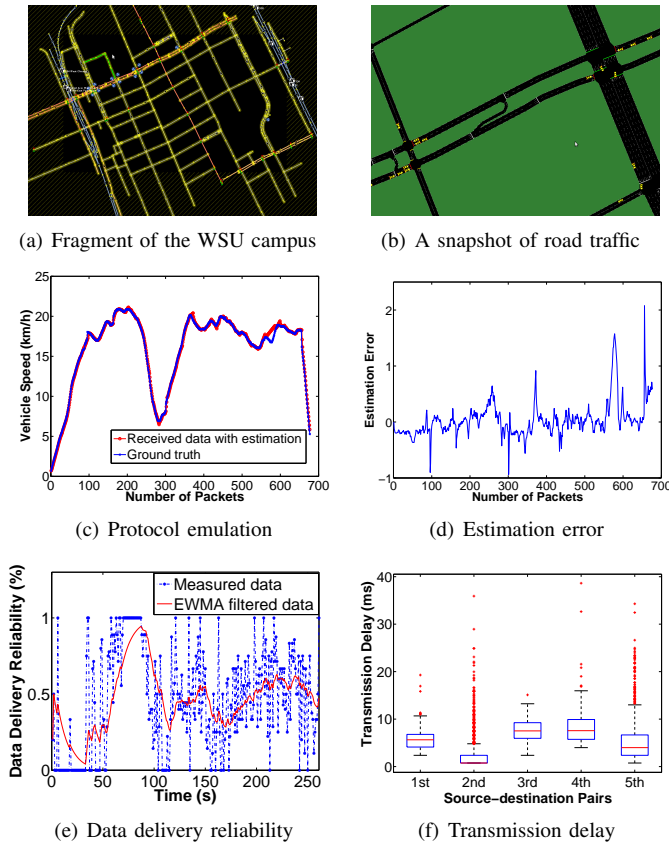


Fig. 7. Real-time emulation via VInsight

Fig. 7 illustrates an example of real-time emulation experiments of VSC networking. By combining VInsight with the

GENI compute servers, the emulation experiment is conducted on the servers with realistic application traffic generated by the GENI-fied OpenXC platforms. In practice, the actual sensing content can be very different for different applications. In this work, we focus on using the real-world vehicle speed to drive the emulation of a data dissemination/flooding protocol in a network of vehicles with mobility. The mobility is based on the Krauss mobility model using SUMO and performed with the road map of WSU campus. The selected road fragment covers an area of 2km by 2km.

Fig. 7(a) depicts the street layout used in the emulation. To ensure realistic results, we consider traffic lights, speed limits, one-way multi-lanes, and right-of-way rules, etc. This allows us to simulate common vehicular situations such as overtakes and stops at intersections. Fig. 7(b) shows a snapshot of the road traffic generated using SUMO.

In the emulation, 1,000 vehicles are simulated and the emulation time is 600 seconds which is long enough to transmit all the data. Each emulation run consists of one broadcasting task that is started by 5 random sources chosen from the vehicle set. Every source sends 5,000 packets at a data rate of 6 Mbps to its destination selected randomly from the same set of vehicles. The packet size is set to 500 bytes and the TTL (Time to Live) is 5. If the packet’s TTL expires (i.e., TTL=0), it is discarded no matter whether the destination vehicle is reached or not.

We run the emulation for 10 times and evaluate the broadcast protocol in terms of two metrics: data delivery reliability and transmission delay. The data delivery reliability is defined as the ratio between the number of packets received by the destination vehicle and the total number of packets sent from the source vehicle; for this metric, we focus on a specific source-destination pair. When multiple copies of a packet reaches its destination vehicle, the delivery latency of the copy first arriving at the destination is used as the packet’s transmission delay.

For a typical three-hop path in the emulation, Fig. 7(c) shows the ground truth and the speed data received from the source vehicle. In the presence of packet loss, we use estimated speed (through vehicles’ locations and speed using the Kalman filter) to represent the lost data. The difference between the ground-truth and estimated speed indicates the estimation error, as shown in Fig. 7(d). This can help verify the effectiveness of the mobility model used by researchers for protocol design and provide valuable feedback on how to further refine it.

Fig. 7(e) shows the changes of data delivery reliability as a function of time. The dash-dot line with asterisk marks depicts the measured data, whereas the solid line shows the EWMA (exponentially weighted moving average) filtered data. The EWMA filter with a weight factor c ($c=0.9$) is used to deal with measurement fluctuation. From the results in Fig. 7(e), we can see that sudden data delivery reliability drops happen a lot with the arrival of packets. This always leads to a low throughput for data dissemination. There are two main reasons. 1) First, due to co-existing source vehicles and signal

attenuation, the packet loss is inevitable. 2) Second, the relative distance between the source vehicle and the destination vehicle tends to increase when they are moving towards different directions or after they meet. In this situation, the data delivery reliability may be degraded. Transmission under bad data delivery reliability should be omitted to avoid inefficient occupancy of the network resources. The aforementioned findings provide insight into data delivery properties of VSC networks, which are essential for designing appropriate protocols for VSC networks.

Fig. 7(f) shows the average transmission delay per packet for successfully delivered packets of each source-destination pair. The average path length of different pairs is 2.1, 4.3, 1.25, 1, and 2.9 hops respectively. It is important to note that the source-destination pairs are with different hop-distance between the source vehicle and the destination vehicle. Intuitively, the average transmission delay of longer distance source-destination pair may be larger than that of source-destination pair with shorter distance. However, it is not always the case. We find that a packet's transmission delay also depends on other factors such as real-time traffic data generation rate, physical distance between the source and the destination, the number of co-existing source vehicles in the network and in the neighborhood, and the moving directions of vehicles. Detailed study of these factors is beyond the scope of this paper, and interested readers are encouraged to conduct this future study using VInsight.

VI. CONCLUDING REMARKS

VInsight provides basic building blocks for enabling open innovation in networked VSC systems. The GENI-fied OpenXC platform opens up vehicle internals and enables innovative VSC application exploration as well as high-fidelity emulation of networked VSC systems. The WiMAX network of VInsight is operational and connects field-deployed GENI-fied OpenXC platforms to the GENI infrastructure for high-fidelity, at-scale evaluation of networked VSC solutions. The GENI-fied WiMAX measurement tools also enable the understanding of real-world WiMAX link characteristics as well as the high-fidelity emulation of networked VSC systems with WiMAX network access. The individual components of VInsight are open to the public, and the relevant software packages are also publicly accessible.⁴

Using the building blocks of VInsight, we will deploy ~20 GENI-fied OpenXC platforms together with the WiMAX measurement tools in the police patrol vehicles at Wayne State University; we will also develop a parallel, distributed VSC emulation system that, through the WiMAX network of VInsight, integrates the GENI-fied OpenXC platforms in field with the parallel ns-3 emulation in GENI racks. Together, these infrastructures will enable unconstrained, open exploration of networked VSC systems at the high-fidelity and scale that would be infeasible otherwise. The synergy between the WiMAX network of VInsight and the campus safety patrol at

Wayne State University will enable the long-term availability of these infrastructures to the general public for solving the grand challenges of road transportation.

REFERENCES

- [1] E. Strom, H. Hartenstein, P. Santi, and W. Wiesbeck, "Vehicular communications," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1158–1161, 2011.
- [2] S. Kobayashi, S. Plotkin, and S. K. Ribeiro, "Energy efficiency technologies for road vehicles," *Energy Efficiency*, vol. 2, pp. 125–137, 2009.
- [3] J. Lee, D. Nelson, and H. Lohse-Busch, "Vehicle inertia impact on fuel consumption of conventional and hybrid electric vehicles using acceleration and coast driving strategy," in *SAE*, 2009.
- [4] S. Li, H. Peng, K. Li, and J. Wang, "Minimum fuel control strategy in automated car-following scenarios," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, 2012.
- [5] CAR 2 CAR Communication Consortium, "C2C-CC manifesto: Overview of the C2C-CC system," Tech. Rep., 2007.
- [6] R. K. Deering, "Annual report of the crash avoidance metrics partnership," NHTSA Office of Advanced Safety Research Report, Tech. Rep. DOT HS 809 531, 2002.
- [7] U.S. Department of Transportation Intelligent Transportation Systems Joint Program Office, "Connected vehicle testbeds," http://www.its.dot.gov/connected_vehicle/pdf/DOT_CVBrochure.pdf, 2013.
- [8] "USDOT safety pilot model deployment," <http://safetypilot.umtri.umich.edu/index.php?content=about>.
- [9] V. Kumar, L. Lin, D. Krajzewicz, F. Hrizi, O. Martinez, J. Gozalvez, and R. Bauza, "ITETRIS: Adaptation of its technologies for large scale integrated simulation," in *the 71st of Vehicular Technology Conference (VTC 2010)*, 2010, pp. 1–5.
- [10] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "GENI: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, no. 2014, pp. 5–23, 2014.
- [11] T. Rakotoarivelo, G. Jourjon, and M. Ott, "Designing and orchestrating reproducible experiments on federated networking testbeds," *Computer Networks*, vol. 63, pp. 173–187, 2014.
- [12] Ford Motor Company, "OpenXC VI firmware documentation," <https://media.readthedocs.org/pdf/openxc-vehicle-interface-firmware/latest/openxc-vehicle-interface-firmware.pdf>, 2013.
- [13] I. Aldmour, "Lte and wimax: Comparison and future perspective," *Communications and Network*, 2013.
- [14] "2014 fuel economy datafile," <http://catalog.data.gov/dataset/fuel-economy-data/resource/adc91f11-c288-42e3-be4b-c36c03c69e1b>, 2014.

⁴The software packages can be downloaded from <http://www.cs.wayne.edu/~hzhang/group/software/VInsight.zip>.