Automotive V2X on Phones: Enabling next-generation mobile ITS apps

Jason H. Gao
Department of EECS
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
Email: jasongao@csail.mit.edu

Li-Shiuan Peh
Department of EECS
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
Email: peh@csail.mit.edu

Abstract—**Automotive connectivity standards promise to usher in new apps and services that utilize vehicle-to-vehicle communications, and mobile app trends point to the potential of direct device-to-device connections. The convergence of the two, made possible by new hardware, system software, and programming abstractions, promises to realize next-generation mobile ITS apps. The city-scale impact of such convergence can now be projected using new evaluation infrastructure that captures interactions between humans, vehicles, devices and networks.**

I. INTRODUCTION

Transportation networks and services are becoming increasingly intelligent and connected, resulting in intelligent transportation systems (ITS) and connected vehicles (CVs). Emerging standards such as 802.11p and DSRC promise to provide the interconnectivity required for ITS and CVs, but technology adoption in a typically safety-critical and government-regulated industry such as automotive can be slow relative to the mobile phone industry, which sees rapid introduction of new technologies. We foresee a faster path to adoption of automotive and mobile interconnectivity by leveraging the quick turnover of mobile devices and smartphones, paving the way for smart cities of the future.

A. Transportation connectivity trends

Transportation connectivity spans broadly from interconnections between components within a vehicle to wide area connectivity between vehicles in cities. Trends indicate that wide area connectivity for vehicles of the future is moving towards Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, collectively termed V2X. There is increasingly wide industry and government support, including programs in Europe (Car2Car [1]), Germany (simTD [2]), Japan [3], the United States [4], and Singapore [5]. Numerous V2X applications have been proposed or deployed, including apps in multimedia, safety, road pricing, and more [6]. The use of V2X is due to recognition of the need for fast, constant connectivity on-the-go, and that existing connectivity, such as WiFi or LTE cellular networks, does not suffice:

- WiFi and LTE's centralized architecture incurs higher latency, as communication between vehicles must traverse a base station; thus, they are unable to support the real-time response required for safety apps [7].
- Cellular data networks are increasingly overloaded, metered, and capped [8]. Even additional capacity will be overtaken by increased demand from: 1) Use of LTE in new vehicles and home broadband; 2) Higher screen resolutions that increase demand for high-definition content.
- WiFi service is often available at subway stations, bus stops, etc., but its short range makes it difficult to support uninterrupted connectivity with high mobility clients [9].

Heterogenous networks (HetNets) promote a vision of achieving pervasive internet connectivity by leveraging varying network technologies and adhoc networks. Related works on vehicular adhoc networks (VANETs) in particular address topics such as seamless 3G connectivity across fleets [10], vehicular routing [11], context-aware content delivery [12], and mobile IP optimizations [13], but work on HetNets has not focused on hosting services and applications directly on the mobile devices, within the VANET. Emerging V2X promises significant benefits for both HetNets, and for next-generation ITS apps that leapfrog the need to access Internet services:

Ultra-fast response times: Low latency communications is critical in transportation apps, especially safety applications.

Location-based services: V2X complements the location-aware nature of ITS apps, and naturally connects nearby nodes with relevant information.

The adoption of V2X technology, however, depends on the deployment of V2X-enabled devices in vehicles and roadside access points, which hinges on customer renewal rate of vehicles and transport authorities' policies. This is slow compared to the renewal rate of consumer technology products, such as smartphones. Smartphones are pervasive in most cities, and the very fast turnover of mobile devices means new smartphone connectivity technologies can be introduced very quickly.

B. Smartphone connectivity trends

Various forms of wireless device-to-device (D2D) communication for connecting mobile phones, analogous to V2X for connecting vehicles, are already available. These include interfaces such as ad-hoc WiFi, WiFi Direct, and Bluetooth. Since mobile phones are already equipped with increasingly powerful processors and sophisticated sensors such as GPS, camera, accelerometer, magnetometer, barometer, proximity, ambient light, etc., and many ITS apps now sense and generate large amounts of data, phones can leverage D2D to collaboratively compute on sensed data in situ and return results to the user with high responsiveness.

These existing phone-based D2D interfaces, however, are limited to short-range, low mobility scenarios. Furthermore, there are constraints on the number of devices, mobility of devices, and on the ease of setup and operating system support.

Mobile ITS apps will also require support for location-aware P2P discovery of services and devices, and programming interfaces to access this support. Current mobile operating systems provide some limited support for P2P communications and service discovery through standards such as Bluetooth, WiFi Direct [14], and iOS Multipeer Connectivity [15], but significant limitations make them unsuitable for highly responsive, mobile, and pervasive ITS apps:

Centralized topology: WiFi Direct facilitates easier setup of D2D networks, but one device must serve as an access point (the *group owner*) and all other devices must communicate through it, negatively affecting latency and resulting in a centralized architecture that does not support highly mobile networks with rapidly changing topologies. This largely limits WiFi Direct apps to close-range, static deployments between a handful of phones. Bluetooth is similarly limited to star topologies. The emerging LTE Direct [16] standard requires the use of the LTE base station as a centralized coordinator, and thus suffers from similar issues. It also requires modifications to the LTE base station, which may slow adoption.

Lack of seamless connectivity: WiFi Direct and Bluetooth require user interaction to initiate connections between devices, limiting their usefulness in networks where nodes may rapidly come and go.

Lack of pervasive support: The existence of several D2D standards has fragmented platform support. Android does not support ad-hoc WiFi, but does support WiFi Direct. iOS does not support WiFi Direct, and iOS Multipeer Connectivity is limited to iOS devices. Different mobile OSs often do not support the same Bluetooth profiles. LTE Direct is not available yet, and hinges on mobile provider's support.

These limitations render these standards unsuitable for networks with highly mobile nodes that result in a rapidly changing topology, and preclude the use of multi-hop mesh networking. Thus, there is no currently available mobile networking standard or programming interface that achieves the highly-flexible topology, rapid background connectivity, and fast response times required for next-generation ITS apps.

C. Convergence of vehicular and smartphone connectivity

Phones are now increasingly part and parcel of intelligent transportation services, enabling easily available services for mapping, navigation, transit prediction, ride sharing, and more.

Smartphone capabilities are also being merged into vehicles: car manufacturers are designing vehicles with the capability to interface to phones, offering features such as integrated information sharing and connectivity with mobile devices [17], built-in dashboard docks for smartphones [18], [19], and there exist aftermarket devices to bring cellular data connectivity and telematics to existing vehicles [20].

As vehicles and phones become more interconnected and interdependent in future transportation systems, we ask: can wireless vehicle and phone connectivity converge? Can V2X be integrated into mobile devices, resulting in "X2X" communications to connect the "Internet of Cars, Phones, and Everything Else"? The convergence of vehicular and smartphone connectivity affords several advantages:

Fast adoption path: By leveraging the pervasiveness and rapid upgrade cycle of mobile devices, we can quickly realize not just the growth of mobile D2D or the "Internet of Cars," but the "Internet of Things," too, as we evolve V2X to X2X, connecting everything to everything.

Extends transportation connectivity: X2X rapidly extends connectivity not just to automobiles, but also to pedestrians on sidewalks and passengers on buses, taxis, trains, shared mobility, cyclists and more.

Extends mobile connectivity: X2X offers alternative and complementary connectivity to cellular that is ultra-fast and infrastructure-less.

Together, a fabric of V2X-connected nodes will enable next-generation ITS apps and beyond.

II. APPLICATIONS: NEXT-GENERATION ITS APPS

The convergence of mobile and vehicular connectivity will enable new kinds of applications that lie beyond the safety applications currently indicated for V2X. Such apps that straddle the line between vehicular and mobile apps include:

- Collaborative info-sharing between vehicles and phones.
- Real-time traffic management.
- Participatory urban sensing and planning.
- Real-time multiplayer mobile gaming.
- Mobility-on-demand and ride-sharing.
- Crowd-counting and mass transit capacity prediction.

To demonstrate some of these new applications, we have developed several phone-based ITS applications that leverage device-to-device communications. These apps have also demonstrated the need for ultra-fast and longer-range V2X for next-generation ITS, as neither cellular communications nor existing D2D communications like ad-hoc WiFi suffice.

A. RoadRunner

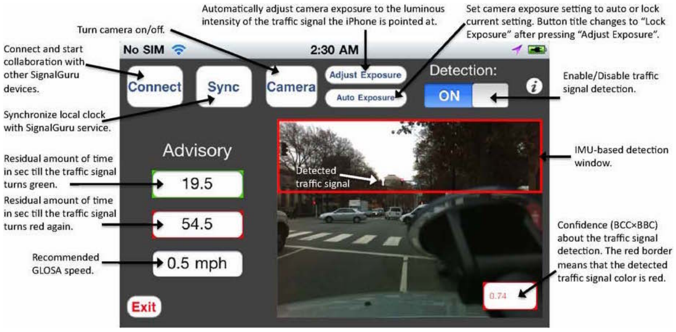
RoadRunner [21] leverages V2X communications to realize city-wide fine-grained traffic management. Roads or geographic areas are divided into regions, and some can be designated as congestion controlled. To enter such a region, a vehicle must posses a digital token. When it exits, it passes the token to another vehicle via V2V, or back to a fallback server if there are no nearby vehicles that need it.

The prototype deployment in Cambridge, MA, USA on 10 vehicles over 2 hours tested both ad-hoc WiFi and 802.11p for V2V token exchanges (see Figure 1). Due to its limited range, WiFi V2V offloaded only 5 token exchanges (6.8% of all token exchanges) from the cellular network, at a mean distance of 29.2 meters. With 802.11p V2V, 47 token exchanges (43% of all token exchanges) were offloaded, at a mean distance of 175.7 meters. WiFi's limited range also negatively impacted driver experience due to tokens getting stuck on cars, while 802.11p ensured effective congestion control.

B. SignalGuru

SignalGuru [23] runs on windshield-mounted iPhones that collaboratively sense and learn traffic signal transitions. It provides Green Light Optimal Speed Advisory (GLOSA) to drivers and was shown to accurately predict transitions to within 2s when deployed over 5 and 8 vehicles at Cambridge,

Fig. 1: Picture of deployment setup in each vehicle. From [22].



Automatically adjust camera exposure to the luminous intensity of the traffic signal the iPhone is pointed at.

Set camera exposure setting to auto or lock current setting. Button title changes to "Lock Exposure" after pressing "Adjust Exposure".

Turn camera on/off.

Connect and start collaboration with other SignalGuru devices.

Enable/Disable traffic signal detection.

Synchronize local clock with SignalGuru service.

IMU-based detection window.

Residual amount of time in sec till the traffic signal turns green.

Residual amount of time in sec till the traffic signal turns red again.

Confidence (BCC×BBC) about the traffic signal detection. The red border means that the detected traffic signal color is red.

Recommended GLOSA speed.

Fig. 2: Screenshot of SignalGuru. From [23].

MA, and Singapore respectively, and reducing fuel consumption by 20%. Figure 2 shows the app's interface. The app used ad-hoc WiFi to share observed traffic signal transitions between vehicles, but the very limited range of ad-hoc WiFi resulted in the need for a road-side node to relay sharing between vehicles. Longer-range V2V would eliminate the need for such infrastructure.

C. Panoramio

We implemented a clone of Panoramio, a popular location-based photo sharing app [24], that leverages D2D. A geographical area is divided into a grid of regions, and the phones within each region collaboratively host location-based services. Apps can access data in other regions via multi-hop networking between regions. We also implemented a baseline comparison app that used only the cellular network for comparison. The D2D-enabled app improved latency 10X versus the 3G baseline and 2X versus 4G, and offloaded cellular network accesses by 96%. The deployment with 20 devices was limited to relatively small regions and distances between phones, again due to the limited range of ad-hoc WiFi. Longer-range and faster V2X would allow much larger regions, and improve service responsiveness.

D. Bus Capacity Prediction

Bus Capacity Prediction (BCP) [25] is an app that predicts the number of passengers on incoming buses so that users at bus stops can make decisions like waiting for a less crowded bus or boarding alternative buses. It was deployed along a campus bus route in Singapore, on 8 phones, and uses statistical models for (dis)embarking passengers, collected via on-vehicle infrared sensors and cameras installed at bus stops.

E. Related Work

The apps presented above show that significant improvements in application performance can be realized by rethinking the current cloud-oriented model and moving towards a model that leverages V2X designed for ultra-fast response times and longer-range. Here we highlight prior works that explored alternative models and how V2X can help. Cabernet [9] is a vehicle content delivery network that uses WiFi to deliver data to moving vehicles, viable for non-interactive applications. V2X on phones would enable content delivery between vehicles and phones, sustaining communications beyond just a few seconds, enabling interactive applications. Sankaran et. al [26] developed a dynamic framework for deploying highly localized mobile web apps that communicate over a delay-tolerant network of phones using WiFi Direct or Bluetooth. V2X for phones could improve the range and responsiveness for all web apps utilizing this framework, and reduce the relatively high app deployment latency of several seconds. FireChat [27] is a popular iOS and Android app that sends chat messages via peer-to-peer mesh networking, operating over WiFi and Bluetooth. The limited range of WiFi and Bluetooth and store-and-forward behavior delays message delivery when there are few phones around. V2X on phones would allow FireChat to send messages further and faster.

III. HARDWARE: REALIZING V2X ON MOBILE PHONES

The longer-range and ultra-fast response time of V2X communications holds promise to enable new ITS apps on mobile phones, but existing V2X radios, such as the 802.11p radios used in RoadRunner, suffer from relatively high power consumption and large form factor. In this section, we demonstrate that it is possible to realize 802.11p V2X within the tight *size* and *power* budget of a mobile phone [28].

Compared to 802.11a WiFi that already exists on phones, 802.11p doubles the OFDM modulation time domain parameters to overcome the high mobility and severe fading of vehicular environments (Table I). Thus, an existing 802.11a WiFi baseband at half clock frequency can support 802.11p (as our system prototype demonstrates), and existing digital and ADC/DAC components can be shared to readily support 802.11p by adding a RF front-end, as shown in Figure 3. For further integration, a single RF front-end could support both 802.11a and 802.11p if the maximum carrier frequency range were extended from 5.875GHz to 5.925GHz. In short, 802.11p can be realized on phones with the engineering of a RF front-end that meets 802.11p specifications.

This is non-trivial though. The RF front-end of a communications system is one of the most critical components, with multiple packages on the mainboard taking up a significant footprint in existing smartphones. III-V semiconductor devices (e.g. GaN) show improved power density and efficiency compared to CMOS devices for FEMs and will readily meet the high transmit power (and corresponding long range) requirements of 802.11p, but RF transceiver circuitry is still typically in CMOS. To realize 802.11p within the size constraints of

Characteristic	***802.11p***	***802.11a***
User mobility	Vehicular (high)	Personal (low)
Operating frequencies	5.85–5.925GHz	5.15–5.825GHz
Channel bandwidth	10MHz	20MHz
Max. output power	760mW	40/200mW
Data rate	3–27Mbps	6–54Mbps
Modulation	BPSK–64QAM	BPSK–64QAM
OFDM symbol duration	8us	4us
Guard time	1.6us	0.8us
Preamble duration	32us	16us
Subcarrier spacing	156kHz	312kHz

TABLE I: Comparison of 802.11p vs. 802.11a [28].

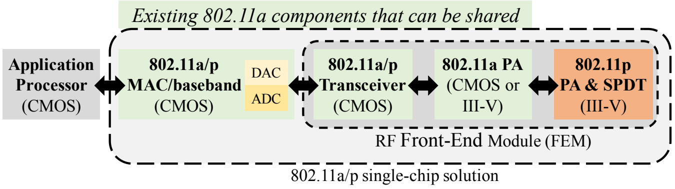


Fig. 3: Proposed 802.11a/p FEM integration [28].

a mobile phone, we leverage a unique LEES (Low Energy Electronics Systems) process, detailed in [28], to fabricate CMOS and III-V devices on a *single* die.

This new device and process technology, combined with a novel circuit design and application-level adaptive gain control, makes low-power and small form-factor 802.11p V2X radios in mobile phones feasible.

A. System prototype

The transmit mode dominates power consumption, so we designed and implemented an entire 802.11p transmit chain to validate feasibility and evaluate power savings. A USB-Ethernet adapter connects an Android smartphone via USB On-the-Go to a Xilinx XUPV5 FPGA board, which is running the Airblue 802.11a baseband [29] at half clock rate to achieve 802.11p compatibility.

To demonstrate RF front-end functionality before the LEES process was ready, we designed and fabricated a reference front-end circuit using a standard commercial $0.18\,\mu\mathrm{m}$ CMOS and $0.25\,\mu\mathrm{m}$ GaN-on-SiC processes on separate dies. After calibration, the CMOS transmitter achieves -52dB single-sideband (SSB) rejection and -36.5dB EVM at the output using 802.11p baseband I/Q signals. A novel circuit design for the GaN PA (detailed in [28]) achieves -30.5dB EVM and 22% drain efficiency across one decade of output power up to the maximum of 28.8 dBm for 802.11p.

B. System prototype evaluation

Our circuit's characteristic of good power efficiency across a wide range of transmit power (unlike a conventional PA whose efficiency exponentially decreases as output power drops) supports application-level adaptive power control (ALAPC) to fine-tune the radio's transmit power and power consumption to a desired V2X communication distance. This joint hardware-software optimization fits the system within the power budget.

We use the location and communications traces of RoadRunner and SignalGuru (Section II), from prior deployments which originally used off-the-shelf V2X radios or adhoc WiFi, to estimate achievable system power savings. We obtain a table that corresponds power level to distance from experimental measurements of the GaN PA, using 64-QAM coding.

1) RoadRunner: In the original RoadRunner deployment, every V2X token exchange is conducted at full radio power (our baseline). We assume that with our adaptive power control, they would instead be transmitted at the minimum power required to reach the other vehicle. Our new PA design with ALAPC reduces power by 47%.

2) SignalGuru: Whenever a vehicle broadcasts a SignalGuru packet, we calculate the power level required to reach the nearest other vehicle, ranging from 19.8 to 28.8 dBm. In our baseline, we assume every broadcast is transmitted at maximum power (28.8 dBm). With our system, SignalGuru broadcasts use 56% less power.

With ALAPC only on a conventional CMOS PA, the power savings of RoadRunner and SignalGuru are only 5% and 25%, respectively, underscoring the importance of the new PA's improved efficiency across a wide range of transmit power. To put our power reductions of 1.6W (RoadRunner) and 1.9W (SignalGuru) in context, we measured a Samsung Galaxy S4 smartphone's power consumption to range between 1 W (screen on, idle) and 11 W (running a CPU-intensive benchmark) using a Monsoon Solutions power monitor.

In summary, with a full system prototype consisting of fabricated GaN and CMOS circuits, off-the-shelf CMOS ADC/DACs, an 802.11p baseband on an FPGA, and an Android phone with custom software, we demonstrate the feasibility of supporting 802.11p-based V2X communications within the stringent power and area constraints of a phone.

IV. SYSTEM SOFTWARE DESIGN: PROGRAMMING NEXT-GEN APPS THAT USE V2X

Programming mobile apps to utilize V2X communications is still a challenge. Realizing the vision of converged mobile and automotive connectivity for new ITS apps will require a way to easily program a collection of mobile devices (whether phones, vehicles, or anything else) interconnected via V2X.

Mobile apps today still use the conventional client-server model, with a lightweight front-end on the phone delegating compute-intensive and data storage tasks to servers in the cloud via LTE or WiFi. This model is widely used for simplicity, but for high mobility and real-time ITS apps, incurs the same disadvantages as described in Section I-A for vehicles. Additionally, cellular data transmission drains energy [30], a limited resource for mobile phones. And even if mobile app developers choose to use existing D2D interfaces described in Section I-B, the app developer must explicitly manage connectivity between individual devices, data partitioning, and distributed computation. These challenges make it difficult to motivate the use of D2D and V2X for mobile apps instead of a relatively simple client-server / cloud-oriented model.

Research into new programming abstractions will help ease the transition from today's cloud-oriented model to a holistic model that intelligently interweaves resources across both the cloud and edge devices such as mobile phones and vehicles.

A. DIPLOMA

One approach we propose is to translate the shared memory model from parallel computing to a collection of mobile nodes, and address the unique challenges that mobility and unreliable wireless networking present in achieving consistency and coherence. Application developers see a single global shared memory space, while the system hides the underlying mobility of nodes and takes care of coordinating V2X between devices.

DIPLOMA (**DI**stributed **P**rogramming **L**ayer **O**ver **M**obile **A**gents) is a programming layer and distributed shared memory system that provides coherent relaxed-consistency access to data residing on different mobile phones across a large geographic area that is divided into regions. We extend Virtual Nodes (VNs) [31] into Virtual Cores (VCores) to abstract a collection of unreliable mobile nodes into a stationary reliable virtual node, while addressing practical issues arising from unpredictable mobility and unreliable networking by judiciously using the cloud for occasional coordination.

Each VCore contributes to a global shared memory space, communicating across VCores via multi-hop. A programming API provides atomic read and write operations (Atoms) that always execute once or fail completely, like a critical section or an *acquire-release* block in Release Consistency [32]. By default, we guarantee relaxed consistency [32] and allow Atoms to be reordered by the unpredictable network, but programmers can optionally enforce stricter ordering with a memory fence primitive. To improve performance, we also design and implement a novel *write update, snoopy (broadcast-based) cache coherence protocol resilient to unreliable networking*, building on timestamp snooping [33] and INSO [34], which are multiprocessor broadcast-based protocols that achieve ordering on unordered networks.

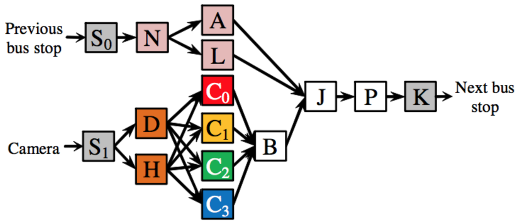
Using DIPLOMA, we implemented a D2D-leveraging app, Panoramio-clone, and show concrete benefits (see Section II).

B. MobiStreams

Distributed Stream Processing Systems (DSPSs) can also hide the complexity of fault tolerance, workload balancing, and connectivity management while providing scalable computing power across devices. A typical DSPS might stream in live data from an end-user device (such as sensors on a smartphone or vehicle) and distribute processing tasks to servers in a datacenter, which then send results back to the user devices.

MobiStreams [25] is a DSPS that runs directly on smartphones interconnected by D2D via ad-hoc WiFi, whereas other DSPSs typically run on clusters of reliable servers interconnected by high-speed Ethernet. To overcome the unreliability of using smartphones, it builds up a reliable DSPS using check-pointing, with two new approaches to reduce overhead: 1) Token-triggered checkpointing: source operators can generate tokens that flow down and trigger checkpointing at each downstream operator, avoiding the redundant data saves that occur in prior fault tolerance schemes. 2) Broadcast-based checkpointing: checkpoint data transmission is split into multiple phases, and uses unreliable UDP broadcasts in the first phases to avoid redundant transmission.

MobiStreams was implemented as a middleware running on iOS, with two example applications from Section II to evaluate its performance: 1) SignalGuru and 2) Bus Capacity Prediction (BCP). Figure 4 shows the stream graph for BCP. In these two apps, MobiStreams increased throughput by up to 43X and reduced latency by 10 to 95% versus server-based DSPSs.



S_0: data from previous bus stop, **N**: noise filter, **A**: prediction model for bus arrival time, **L**: prediction model for alighting passengers, **S_1**: camera data source, **D**: dispatcher, **H**: motion detection (passerby filter), **C**: counter (counting people in images), **B**: predication model for boarding passengers, **J**: join, **P**: prediction model for bus capacity, **K**: sink (to next bus stop).

Fig. 4: The DSPS for each bus stop in the BCP app. Operators with the same color run on the same node. From [25].

C. Related work

There is substantial work in wireless sensor networks on programming frameworks [35] and system software [36], [37], but they are not compatible with phone programming. Other works have investigated mobile P2P frameworks, e.g. for content delivery [38] or social networking [39], but do not holistically address D2D connectivity management, distributed data storage and computation, and programming interfaces.

V. EVALUATION AND TEST INFRASTRUCTURE

The evaluation of connected mobile and vehicular system prototypes is hampered by logistical hurdles in large deployments, which have been difficult and time-intensive to execute, not to mention safety and regulation constraints. The apps described have been limited to small pilot deployments of several vehicles, which provided mobility traces that were useful for evaluating new hardware, such as the development of 802.11p radios for phones (Section III), and evaluating system-level performance, such as latency improvements and cellular offload benefits. But these small deployments are insufficient to measure city-scale implications such as traffic congestion, bus capacities, and travel times.

Large-scale simulation infrastructure can enable evaluation of city-scale implications of V2X-enabled ITS apps. SimMobility is an integrated simulation platform that enables development and evaluation of future ITS scenarios, with integrated models of human and commercial activities, land use, transportation, environmental impacts, and energy use [40]. SimMobility short-term (SimMobility ST) is an agent-based, multimodal micro-simulator, suitable for simulating individual people, vehicles, and devices. Drivers, pedestrians and passengers are modeled as agents whose movements are captured at millisecond scale, including behaviors such as lane changing, vehicle accelerating and braking, walking and driving route decisions, riders boarding buses, etc.

While agent-based simulators that model the transportation environment in detail have been available for decades, the ability to simulate the interactions between human agents and mobile ITS apps was lacking. This is critical for evaluating the implications of new ITS apps at the city scale. Ideally, it should be possible to evaluate unmodified ITS mobile apps (such as Android software like RoadRunner or DIPLOMA), without requiring reimplementation of the app logic inside the simulator. This requires simulation infrastructure to meet the needs of ITS designers and mobile app developers in evaluating pervasive mobile apps using V2X. It must emulate smartphones associated to agents, with simulated human interaction and environmental sensing feeding information into apps, as well as simulated V2X connectivity.

Similitude [41] combines SimMobility with emulated Android devices and the ns-3 network simulator. The Android apps require some rudimentary modifications, but Similitude provides a Java library, *simmobility4android*, to ease integration and provide shims for functionality like location updates, timing, and message passing. A RoadRunner Android app was modified and run in a simulation of a large area (6 x 2.5 km) with a randomized road network to test Similitude's performance. Results showed that Similitude can scale to 288 connected clients, the maximum supported by the test server cluster. Beyond that, SimMobility itself scaled well up to 100,000 agents, and the Android emulators scaled well at 1.0 to 1.5 emulated Android devices per CPU in the cluster. ns-3 proved to be the main bottleneck above 20,000 agents.

As an example of the potential of large-scale evaluation infrastructure like SimMobility to measure city-scale impact, we were able to simulate 74,904 vehicle trips over 24 hours of simulated time for RoadRunner [21], compared to 10 vehicles for 2 hours in the deployment. The simulation showed that RoadRunner improves vehicle travel speed during peak congestion compared to the baseline ERP system by eliminating peak congestion slowdowns, and the slowest 5-minute-average travel speed improves from 45.5 to 49.0 km/h.

VI. CONCLUSION

The convergence of automotive and mobile connectivity will enable new mobile ITS apps and realize cross-industry benefits. We propose that a well-adopted automotive V2X standard, IEEE 802.11p, will be suitably flexible and pervasive for this task. We demonstrated examples of next-generation ITS apps in Section II. We showed V2X is realizable on phones in Section III, and presented new programming abstractions to easily develop V2X-leveraging apps in Section IV. Finally, we present a simulation infrastructure for large-scale evaluation of these apps in Section V. We see pervasive V2X holding promise beyond ITS to connect cyber-physical systems and the Internet of Things.

REFERENCES

[1] "CAR 2 CAR Communication Consortium," car-to-car.org.

[2] D. Hübner and G. Riegelhuth, "A new system architecture for cooperative traffic centres-the simtd field trial," in *ITS World Congress*, 2012.

[3] "Toyota to introduce V2X communications on 2015 Japanese models," www.traffictechnologytoday.com/news.php?NewsID=64353, Traffic Technology Today, Nov. 2014.

[4] "Transportation Sec. Foxx announces steps to accelerate road safety innovation," www.nhtsa.gov/About+NHTSA/Press+Releases/2015/nhtsa-will-accelerate-v2v-efforts, U.S. DOT NHTSA, May 2015.

[5] "Cisco, NXP Invest in Cohda Wireless to Enable the Connected Car," newsroom.cisco.com/press-release-content?articleId=1119826, 2013.

[6] P. Papadimitratos *et al.*, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE Comm. Mag.*, vol. 47, no. 11, pp. 84–95, 2009.

[7] G. Araniti *et al.*, "LTE for vehicular networking: a survey," *IEEE Comm. Mag.*, vol. 51, no. 5, pp. 148–157, 2013.

[8] J. Patterson, "Reality check: Whither unlimited data?" RCR Wireless News Reality Check, Jul. 2015.

[9] J. Eriksson *et al.*, "Cabernet: vehicular content delivery using wifi," in *ACM MobiCom*, 2008.

[10] A. Benslimane *et al.*, "Dynamic clustering-based adaptive mobile gateway management in integrated vanet–3g heterogeneous wireless networks," *Selected Areas in Comm.*, vol. 29, no. 3, Mar. 2011.

[11] Y. Ge *et al.*, "Hierarchical olsr-a scalable proactive routing protocol for heterogeneous ad hoc networks," in *WiMob*. IEEE, 2005.

[12] L. Zhou *et al.*, "Context-aware middleware for multimedia services in heterogeneous networks," *IEEE Intel. Sys.*, vol. PP, no. 99, p. 1, 2009.

[13] H. Fathi *et al.*, "Optimization of mobile ipv6-based handovers to support voip services in wireless heterogeneous networks," *IEEE Trans. Vehicular Technology*, vol. 56, no. 1, pp. 260–270, Jan 2007.

[14] "Wi-Fi Direct," wi-fi.org/Wi-Fi_Direct.php, Wi-Fi Alliance, 2015.

[15] "Multipeer connectivity framework reference," developer.apple.com/library/ios/documentation/MultipeerConnectivity/Reference/MultipeerConnectivityFramework/.

[16] L. Lei *et al.*, "Operator controlled device-to-device comm. in lte-advanced networks," *IEEE Wireless Comm.*, vol. 19, no. 3, p. 96, 2012.

[17] S. Greengard, "Automotive systems get smarter," *Commun. ACM*, vol. 58, no. 10, pp. 18–20, Sep. 2015.

[18] P. Shelly, "Addressing challenges in automotive connectivity: Mobile devices, technologies, and the connected car," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 8, no. 2015-01-0224, pp. 161–169, 2015.

[19] S. Elmer, "Volkswagen ibeetle unveiled with built-in iphone dock," www.autoguide.com/auto-news/2013/04/volkswagen-ibeetle-unveiled-with-built-in-iphone-dock.html, AutoGuide.com, Apr. 2013.

[20] "Automatic: Connect your car to your digital life," www.automatic.com.

[21] J. Gao and L. Peh, "RoadRunner: Infrastructure-less vehicular congestion control," in *ITS World Congress*, 2014.

[22] J. Gao *et al.*, "DIPLOMA: Consistent and coherent shared memory over mobile phones," in *IEEE ICCD*, 2012.

[23] E. Koukoumidis *et al.*, "SignalGuru: leveraging mobile phones for collaborative traffic signal schedule advisory," in *ACM MobiSys*, 2011.

[24] J. C. Abela and E. M. Aguilar, "Panoramio," www.panoramio.com.

[25] H. Wang and L.-S. Peh, in *IPDPS*.

[26] K. Sankaran *et al.*, "Dynamic framework for building highly-localized mobile web dtn applications," in *ACM CHANTS*, 2014, pp. 43–48.

[27] "Firechat," opengarden.com/firechat/, Open Garden, 2015.

[28] P. Choi *et al.*, "A case for leveraging 802.11p for direct phone-to-phone communications," in *ACM/IEEE ISLPED*, 2014.

[29] M. C. Ng *et al.*, "Airblue: A system for cross-layer wireless protocol development," in *ACM/IEEE ANCS*, 2010.

[30] N. Balasubramanian *et al.*, "Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications," in *ACM IMC*, Nov. 2009.

[31] S. Dolev *et al.*, "Brief announcement: Virtual stationary automata for mobile networks," in *ACM PODC*, 2005.

[32] S. V. Adve and K. Gharachorloo, "Shared memory consistency models: A tutorial," *Computer*, vol. 29, pp. 66–76, Dec. 1996.

[33] M. M. K. Martin *et al.*, "Timestamp snooping: an approach for extending smps," in *ACM ASPLOS*, 2000.

[34] N. Agarwal *et al.*, "In-network snoop ordering (INSO): Snoopy coherence on unordered interconnects," in *HPCA*, 2009.

[35] M. Kushwaha *et al.*, "Oasis: A programming framework for service-oriented sensor networks," in *IEEE COMSARE*, 2007.

[36] P. Levis *et al.*, "TinyOS: An operating system for sensor networks," in *Ambient Intelligence*. Springer, 2005, pp. 115–148.

[37] A. Dunkels *et al.*, "Contiki–a lightweight and flexible operating system for tiny networked sensors," in *IEEE Local Computer Networks*, 2004.

[38] H. J. Kang and C. G. Kang, "Mobile device-to-device (d2d) content delivery networking: A design and optimization framework," *IEEE Comm. and Networks*, Oct 2014.

[39] S.-W. Seong *et al.*, "Prpl: a decentralized social networking infrastructure," in *ACM MCS*, 2010.

[40] K. Basak *et al.*, "Modeling reaction time within a traffic simulation model," in *IEEE ITSC*, 2013.

[41] S. N. Hetu *et al.*, "Similitude: Interfacing a traffic simulator and network simulator with emulated android clients," in *IEEE VTC*, 2014.