



A survey of urban vehicular sensing platforms

Uichin Lee*, Mario Gerla

Bell Labs, Alcatel-Lucent Computer Science Department, USA

ARTICLE INFO

Article history:

Available online 30 July 2009

Keywords:

Vehicular sensor networks
Survey
Comparative evaluation

ABSTRACT

Vehicular sensing where vehicles on the road continuously gather, process, and share location-relevant sensor data (e.g., road condition, traffic flow) is emerging as a new network paradigm for sensor information sharing in urban environments. Recently, smartphones have also received a lot of attention for their potential as portable vehicular urban sensing platforms, as they are equipped with a variety of environment and motion sensors (e.g., audio/video, accelerometer, and GPS) and multiple wireless interfaces (e.g., WiFi, Bluetooth and 2/3G). The ability to take a smartphone on board a vehicle and to complement the sensors of the latter with advanced smartphone capabilities is of immense interest to the industry. In this paper we survey recent vehicular sensor network developments and identify new trends. In particular we review the way sensor information is collected, stored and harvested using inter-vehicular communications (e.g., mobility-assist mobility-assisted dissemination and geographic storage), as well using the infrastructure (e.g., centralized and distributed storage in the wired Internet). The comparative performance of the various sensing schemes is important to us. Thus, we review key results by carefully examining and explaining the evaluation methodology, in the process gaining insight into vehicular sensor network design. Our comparative study confirms that system performance is impacted by a variety of factors such as wireless access methods, mobility, user location, and popularity of the information.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Vehicular Ad Hoc Networks (VANETs) are acquiring commercial relevance because of recent advances in inter-vehicular communications via the DSRC/WAVE standard, which stimulates a brand new family of visionary services for vehicles, from entertainment applications to tourist/advertising information, from driver safety to opportunistic intermittent connectivity and Internet access [1,2]. In particular, vehicular sensor networks (VSNs) are emerging as a new tool for effectively monitoring the physical world, especially in urban areas where a high concentration of vehicles equipped with onboard sensors is expected (see Fig. 1) [3,4]. In addition, the rising popularity

of smartphones with onboard sensors and *always-on* mobile Internet connections via 2/3G sheds lights on using smartphones as a platform for participatory vehicular sensing [5–7]. Thus, these kinds of vehicular sensing platforms will enable novel applications such as street-level traffic flow estimation, ride quality monitoring, and proactive urban surveillance.

Vehicles are typically not affected by strict energy constraints and can be easily equipped with powerful processing units, wireless transmitters, and sensing devices even of some complexity, cost, and weight (GPS, chemical spill detectors, still/video cameras, vibration sensors, acoustic detectors, etc.). VSNs represent a significantly novel and challenging deployment scenario, considerably different from more traditional wireless sensor network environments, thus requiring innovative solutions. In fact, differently from traditional wireless sensor nodes, vehicles usually exhibit constrained mobility patterns due to street

* Corresponding author.

E-mail addresses: uclee@bell-labs.com (U. Lee), gerla@cs.ucla.edu (M. Gerla).

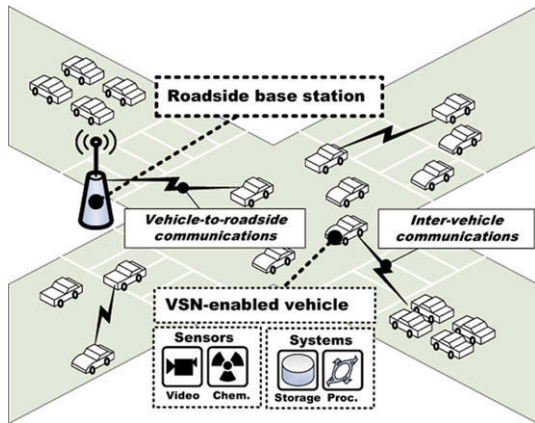


Fig. 1. Vehicular sensor network (VSN).

layouts, junctions, and speed limitations. In addition, they usually have no strict limits on processing power and storage capabilities.

In general, a vehicular sensor network (VSN) platform provides a means of collecting/processing/accessing sensor data. Vehicles continuously collect sensor data from urban streets (e.g., images, accelerometer data, etc), which are then processed to search for information of interest (e.g., recognizing license plates, or inferring traffic patterns). The architecture of information access in a vehicular sensor network is mainly dependent on the underlying wireless access methods in vehicular environments (see Fig. 2). If vehicles are only equipped with inter-vehicular communications devices, it should operate in an infrastructure-free mode. The sensor data has to be processed either locally or cooperatively, and information access is facilitated by Vehicle-to-Vehicle (V2V) communications [8–12]. If vehicles are also equipped with a broadband wireless access method such as 2/3G and WiMax, sensor data sharing can be implemented over the Internet. Mobile users can report the sensor data (or processed sensor data) to the Internet servers, and other users can access the information from those servers [4,13].

In this paper, we survey recent VSN proposals and report some of the key results by carefully examining the evaluation methodology. In Section 2, we overview various wireless communication methods and existing routing

protocols in vehicular networks; and we also present various mobile sensing proposals in the literature. In Section 3, we detail potential vehicular sensing applications such as traffic monitoring, urban surveillance and road surface monitoring. In Section 4, we review V2V-based sensor data sharing techniques using mobility-assisted dissemination [8–11], and geographic storage [12] and examine how vehicular mobility (e.g., speed, density, churning, location, etc.) influences the overall performance of the VSN platforms via simulations. In Section 5, we review infrastructure-based sensor data sharing techniques using centralized or distributed Internet servers [4,13] and illustrate how mobile nodes interact with Internet infrastructure to support the vehicular sensor networking services. Our survey shows that the vehicular sensor networking system performance is mainly influenced by several factors, including: wireless access methods, vehicle mobility (density, speed, and churning), geographic location, and popularity of information.

2. Background and related work

We overview various wireless access methods in vehicular environments, namely DSRC/WAVE, Cellular, WiFi, WiMAX, etc. We then outline key differences that distinguish the vehicular platform from the traditional mobile wireless ad hoc networks (MANETs) and review various routing protocols designed for vehicular networks. Finally, we present various mobile sensing proposals in the literature as mobile vehicular sensing belongs to a general category of mobile sensing.

2.1. Wireless access methods in vehicular environments

2.1.1. DSRC/WAVE

Dedicated Short-Range Communication (DSRC) is a short to medium range communication technology operating in the 5.9 GHz range [14]. The Standards Committee E17.51 endorses a variation of the IEEE 802.11a MAC for the DSRC link. DSRC supports vehicle speed up to 120 mph, nominal transmission range of 300m (up to 1000 m), and default data rate of 6 Mbps (up to 27 Mbps). This will enable operations related to the improvement of traffic flow, highway safety, and other Intelligent Transport System (ITS) applications in a variety of application environments called DSRC/WAVE (Wireless Access in a Vehicu-

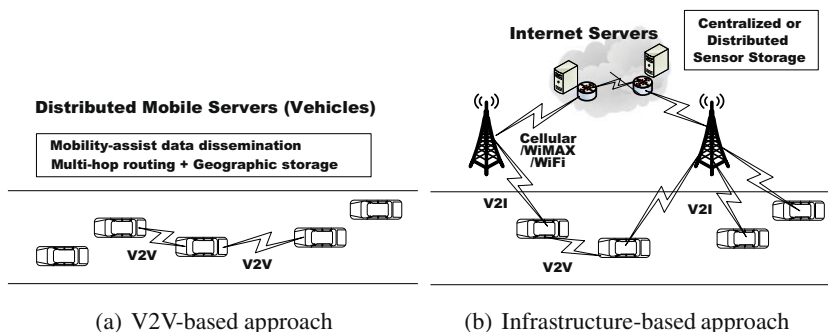


Fig. 2. Vehicular sensor networking scenarios.

lar Environment). DSRC has two modes of operations: (1) Ad hoc mode characterized by distributed multi-hop networking (vehicle–vehicle), (2) Infrastructure mode characterized by a centralized mobile single hop network (vehicle–gateway). Note that depending on the deployment scenarios, gateways can be connected to one another or to the Internet, and they can be equipped with computing and storage devices, e.g., Infostations [15,16]. Readers can find a detailed overview of the DSRC standards in [14].

2.1.2. Cellular networks

Cellular systems have been evolving rapidly to support the ever increasing demands of mobile networking. 2G systems such as IS-95 and GSM support data communications at the maximum rate of 9.6 kbps. To provide higher rate data communications, GSM-based systems use GPRS (<171 kbps) and EDGE (<384 kbps), and IS-95-based CDMA systems use 1xRTT (<141 kbps). Now 3G systems support much higher data rate.¹ UMTS/HSDPA provides maximum rates of 144 kbps, 384 kbps, and 2 Mbps under high mobility, low mobility, and stationary environments, respectively. CDMA2000 1xEvDO (Rev. A) provides 3 Mbps and 1.8 Mbps for down and up links, respectively. The average data rate perceived by users is much lower in practice: <128 kbps for GSM/EDGE and <512 kbps for 3G technologies. In the US, Verizon and Sprint provide 1xEvDO, and AT&T and T-mobile provide GSM/EDGE.

The behavior of 3G services (i.e., 1xEvDO) in a vehicular environment was evaluated by Qureshi et al. [17].² They reported that 1) the average RTT was consistently high (around 600ms) with high variance ($\sigma = 350$ ms); 2) there were a small number of short-lived (<30 ss) disconnections during their experiments; 3) the download throughput varied, ranging from 100 kbps to 420 kbps, and the peak upload throughput was less than 140 kbps; and 4) they found no correlation between the vehicle's speed and the achieved throughput, but geographic location is the dominant factor leading to variations.

2.1.3. WiMAX/802.16e

802.16e or WiMAX (Worldwide Interoperability for Microwave Access) aims at enabling the delivery of last mile wireless broadband access (<40 Mbps) as an alternative to cable and xDSL, thus providing wireless data over long distances. This will fill the gap between 3G and WLAN standards, providing the data rate (tens of Mbps), mobility (<60 km/h), and coverage (<10 km) required to deliver Internet access to mobile clients. For its part, WiBro, developed in Korea based on 802.16e draft version 3, provides 1Km range communications at the maximum rate per user of 6 Mbps and 1 Mbps for down and up links.³ It also supports several service levels including guaranteed QoS for delay sensitive applications, and an intermediate QoS level for delay tolerant applications that require a minimum guaranteed data rate. Han et al. [19] measured the performance of

WiBro networks in a subway whose maximum speed is 90 km/h, and showed that (1) the average uplink and downlink speeds were 2 Mbps and 5.3 Mbps respectively, and (2) the average packet delay (half RTT) was less than 100 ms, and almost all packets experienced delay below 200 ms, except the case when handoffs happened (>400 ms).

2.1.4. WLAN

WiFi or WLAN can also support broadband wireless services. 802.11a/g provides 54Mbps and has a nominal transmission range of 38 m (indoor) and 140 m (outdoor). Despite its short radio range, its ubiquitous deployment makes WLAN an attractive method to support broadband wireless services. It has long been used as a means of Internet access in vehicles, which is known as Wardriving.⁴ Also, open WiFi mesh networking has received a lot of attention; e.g., Meraki sells \$50 WiFi access points and provides Internet access for free by forming a mesh network over those access points.⁵ Readers can find a thorough evaluation of WiFi performance in a vehicular environment in [20].

2.1.5. Possible vehicular networking scenarios

Given the above wireless access methods, we now summarize possible vehicular networking scenarios. If vehicles are only equipped with DSRC, we can have an infrastructure-free mode (V2V only), infrastructure mode (V2I), and mixed mode (V2V and V2I), as shown in Fig. 3a. Note that this can be also done with commercial WiFi devices. The mixed mode has been extensively studied in the research communities in terms of routing and network capacity, and readers can find the details in [21]. If vehicles are only equipped with other broadband wireless access (i.e., cellular, WiMAX), we can have a scenario where vehicles can talk to each other via the Internet as in Fig. 3b. For instance, people with iPhones or other Smart Phones with Internet access can form a P2P overlay network via the Internet. Finally, when vehicles have both DSRC and other broadband wireless access methods, we can have a mixed access scenario as in Fig. 3c. Researchers have mostly focused on the first scenario, yet the second scenario has recently received a lot of attention due to the widespread usage of Smart Phones, or WiBro [22].

2.2. Characteristics of vehicular network environments

In designing protocols for the next generation vehicular network, we recognize that nodes in these networks have significantly different characteristics and demands from those in traditional wireless ad hoc networks deployed in infrastructureless environments (e.g. sensor field, battle-field, etc.). These differences have a significant impact on application infrastructure.

- Vehicles have much higher power reserves than a typical mobile computer. Power can be drawn from on-board batteries, and recharged as needed from a gas-line or alternative fuel engine.

¹ <http://en.wikipedia.org/wiki/3G>.

² Readers can find the evaluation of UMTS/HSDPA systems in a static environment [18].

³ The peak sector (or cell) throughput is 18 Mbps and 6 Mbps for downlink and uplink, respectively.

⁴ <http://en.wikipedia.org/wiki/Wardriving>.

⁵ <http://meraki.com>.

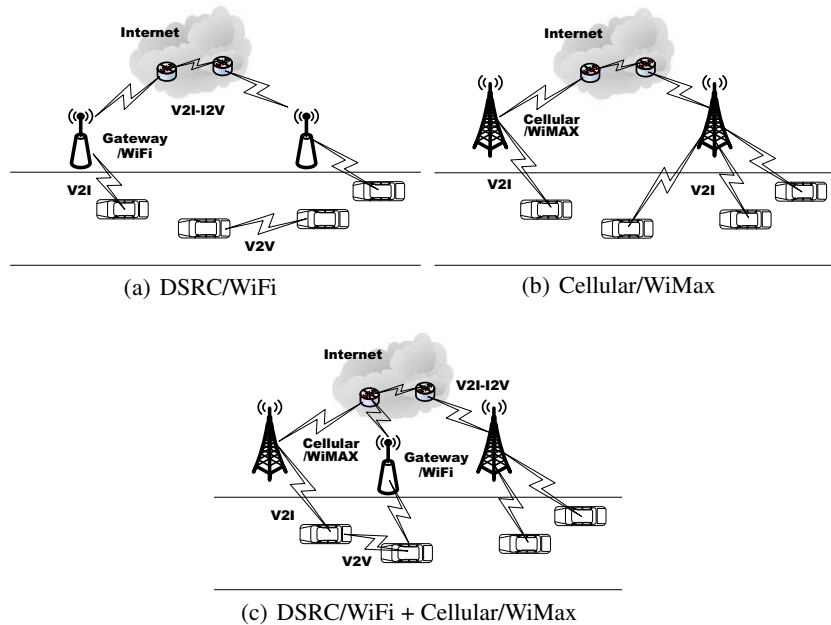


Fig. 3. Possible wireless vehicular networking scenarios.

- Vehicles are orders of magnitude larger in size and weight compared to traditional wireless clients, and can therefore support significantly heavier computing (and sensorial) components. This combined with plentiful power means vehicular computers can be larger, more powerful, and equipped with extremely large storage (up to Terabytes of data), as well as powerful wireless transceivers capable of delivering wire-line data rates.
- Vehicles travel at speeds up to one hundred miles per hour, making sustained, consistent vehicle-to-vehicle communication difficult to maintain. However, “existing statistics” of vehicular motion, such as tendencies to travel together or traffic patterns during commute hours, can help maintain connectivity across mobile vehicular groups.
- Vehicles in a grid are always a few hops away from the infrastructure (WiFi, cellular, satellite, etc.). Thus, network protocol and application design may depend on easy access to the Internet during “normal” operations.

2.3. Routing in vehicular networks

Several VANET applications critically rely on VANET routing protocols (unicast, broadcast, geocasting, etc.). These protocols originate from prior ad hoc network architectures but have been extensively redesigned by targeting the unique characteristics and needs of VANET scenarios and applications. We review the VANET routing protocols first as this offers an initial insight into VANET application characteristics.

2.3.1. Broadcasting

Safety related applications (e.g., forward/backward collision warnings, lane change assistance) call for the

delivery of messages to all nodes located close to the sender (reliable single/multi-hop broadcasting) with high delivery rate and short delay. Recent research addressed this issue by proposing reliable broadcasting strategies [23,2]. Xu et al. [2] studied the impact of rapid repetition of broadcast messages on the packet reception failure in random access protocols. Torrent-Moreno et al. [23] showed channel access time and reception probability under deterministic and statistical channel models. Yin et al. [24] detailed the DSRC PHY layer model and incorporated the model into a VANET simulator to support generic safety application models. ElBatt et al. [25] modeled Cooperative Collision Warning (CCW) applications that broadcast a fixed size packet at a certain rate. They measured the quality of reception using Packet Inter-Reception Time (IRT) that captures the effect of successive packet collisions on the perceived latency. Urban Multi-hop Broadcast (UMB) [26] supports directional broadcast in VANETs. UMB tries to improve reliability of broadcast by alleviating a hidden terminal problem through an RTS/CTS-style handshake, and broadcast storms through black-burst signals to select a forwarding node that is farthest from the sender using location information. Unlike UMB, Broadcast Medium Window (BMW) [27] and Batch Mode Multicast MAC (BMMM) [28] require all the receiving nodes to send back an ACK to the sender in order to achieve reliability. BMMM has also adapted to directional MAC in VANETs [29].

2.3.2. Unicast routing

There are many MANET routing protocols: proactive routing (e.g., DSDV, OLSR) or reactive routing (e.g., AODV, DSR), geographic routing (e.g., GPSR), and hybrid geographic routing (e.g., Terminode), and yet they cannot directly be used due to high mobility and non-uniform distribution of vehicles, which causes intermittent connec-

tivity. In VANETs, geographic or hybrid geographic routing protocols are often preferred. Also, the carry-and-forward strategy is used to overcome intermittent connectivity; when disruption happens, a node stores a packet in its buffer and waits until connectivity is available. Chen et al. [30] considered a “straight highway” scenario and evaluated two ideal strategies: pessimistic (i.e., synchronous), where sources send packets to destinations only as soon as a multi-hop path is available, and optimistic (i.e., carry-and-forward), where intermediate nodes hold packets until a neighbor closer to the destination is detected. In such a highway scenario, they showed that the latter scheme achieves a lower delivery delay. However, in more realistic situations (i.e., Manhattan-style urban mobility and buffer constraints), carry-and-forward protocols call for careful design and tuning. MaxProp [31], part of the UMass DieselNet project⁶ has a ranking strategy to determine packet delivery order where precedence is given in the following order: (1) packets destined to the neighboring nodes, (2) packets containing routing information, (3) acknowledgement packets of delivered data, (4) packets with small hop-counts, and (6) packets with a high probability of being delivered through the other party. VADD [32] rests on the assumption that most node encounters happen in intersection areas. Effective decision strategies are proposed to reduce packet delivery failures and delay. Naumov and Gross [33] proposed a hybrid geographic routing protocol, called Connectivity-Aware Routing (CAR). Route discovery finds a set of anchor points (i.e., junctions) to the destination via flooding. Geographic greedy forwarding is used to deliver packets over the anchored path.

2.3.3. Geocast

Applications for distributed data collection in a VANET call for geographic dissemination strategies that deliver packets to all nodes belonging to target remote areas (or geo-casting), despite possibly interrupted paths [34–36]. MDDV [34] exploits geographic forwarding to the destination region, favoring paths where vehicle density is higher. In MDDV, messages are carried by head vehicles, i.e., best positioned toward the destination with respect to their neighbors. As an alternative, Sormani et al. [35] proposed several strategies based on virtual potential fields generated by propagation functions that encode the target areas and the preferred paths to reach these areas. A node estimates its position in the field and retransmits packets until nodes placed in locations with lower potential values are found; this procedure is repeated until a packet reaches target zones whose potential values are below a certain threshold. Maihöfer et al. [36] proposed abiding geocast, a time stable geocast where messages are delivered to all nodes that are inside a destination region within a certain period of time and discussed design space, semantics, and strategies for abiding geocast.

2.3.4. Infrastructure-assisted hybrid routing

The benefits of using the infrastructure to enhance per node throughput capacity of an ad hoc network are well

documented in [37,38]. Most proposed hybrid (infrastructure and ad hoc) networks are to provide extended coverage of existing services, e.g., wireless LAN and 3G [39–41]. In addition to the extended coverage, a hybrid network can also route packets to a remote mobile node over the infrastructure. Miller et al. [42] proposed a hybrid network implementation where an Access Point (AP) restricts its wireless multi-hop service to k -hops for efficiency. A mobile node joins an AP after receiving a beacon by setting up a default path to that AP. The overall routing protocol shares the same idea of AODV, but their protocol allows the route discovery packet to travel over the APs. Mobile IP for ad hoc networks can be used to support communications with hosts on the Internet. WIANI [43] extended the previous approach and pursued load balancing among APs. To this end, an AP periodically sends a beacon with “local” load information; from this information, a mobile node makes a probabilistic join decision only if it finds an AP with load below a given threshold. A common problem of the previous approaches is that they propose to use RREQ flooding over APs to select and setup the rerouting path. Unfortunately, this does not scale as the number of mobile nodes and APs increases. To solve this problem, Gerla et al. proposed an Overlay Location Service (OLS) [21]. OLS maintains geographic locations of APs and mobile nodes, and allows mobile nodes to efficiently utilize geo-routing not only over the vehicular grid but also over the Internet. In addition, OLS provides a “global” view of AP congestion levels, thus leveraging efficient use of communication resources.

2.4. Mobile sensing and sensor storage

Traditionally, sensor networks have been deployed in static environments, with application-specific monitoring tasks. Recently, opportunistic mobile sensor networks have emerged, which exploit existing devices and sensors, such as cameras in mobile phones [44,45,12,6].

MetroSense [44] proposes a three tier architecture for mobile sensing: servers in the wired Internet are in charge of storing/processing sensed data; Internet-connected stationary Sensor Access Points (SAP) act as gateways between servers and mobile sensors (MS); MS move in the field opportunistically delegating tasks to each other, and “muling” [46,47] data to SAP. MetroSense requires infrastructure support, including Internet-connected servers and remotely deployed SAP. Similarly, Wang et al. proposed data delivery schemes in a Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN) for human-oriented pervasive information gathering [48]. The trade-off between data delivery ratio/delay and replication overhead is mainly investigated in the case of buffer and energy resource constraints. Application-level protocols for the resolution of queries to sensed data have been proposed by Riva and Borcea [45]; Contory abstracts the network as a database and can resolve declarative queries; Spatial Programming hides remote resources, such as nodes, under local variables, thus enabling transparent access; finally, Migratory Services are components that react to changing context, e.g., the target moving out of range by migrating to other nodes.

⁶ UMass’ DieselNet. <http://prisms.cs.umass.edu/dome>.

Regarding dissemination of sensed data through peers, we can mention two solutions from the naturalistic environment, namely ZebraNet [49] and SWIM [16]. ZebraNet addresses remote wildlife tracking, e.g., zebras in Mpala Research Center in Kenya, by equipping animals with collars that embed wireless communication devices, GPS, and biometric sensors. As GPS-equipped animals drift within the park, their collars opportunistically exchange sensed data, which must make its way to the base station (the ranger's truck). ZebraNet proposes two dissemination protocols: a flooding-based approach where zebras exchange all the data within their buffers (either locally generated or received from other animals) with neighbors, and a history-based protocol where data is uploaded only to zebras with good track record of base station encounters. SWIM [16] addresses sparse mobile sensor networks with fixed Infostations as collecting points. Sensed data is epidemically disseminated via single-hop flooding to encountered nodes and offloaded when Infostations are in reach.

Internet-based approaches for *generic* sensor data sharing have a simple multi-tier structure, e.g., ArchRock [50], SensorBase [51], and SensorMap [52], or a semi-hierarchical structure, e.g., IrisNet [53], and Global Sensor Networks (GSN) [54]. In ArchRock [50] and SensorBase [51], sensor data from a sensor network is aggregated at the local gateway and is published to the front-end server through which users can share the data. In SensorBase, back-end servers (called republishers) further process sensor data to enable sensor data searching. SensorMap [52] is a web portal service that provides mechanisms to archive and index data, process queries, and aggregate and present results on geocentric Web interfaces such as Microsoft Virtual Earth. In IrisNet [53], each organization maintains database servers for its own sensors, and a global naming service is provided for information access. Instead of using a global naming service, GSN [54] uses structureless P2P communications for query resolution.

3. Vehicular sensing applications

3.1. Street-level traffic flow estimation

Most traffic information systems currently analyze data from closed-circuit cameras and sensors installed on the roads to estimate traffic conditions. This information is then shared via radio broadcasts, traffic reports on the Internet (e.g., Google Traffic), and en route traffic displays and signals. The coverage of those systems, however, is extremely limited (e.g., mainly highways) due to high installation and maintenance costs. To overcome such limitations, researchers have proposed to use vehicles as sensors to collect GPS measurements using various wireless connectivity such as roadside WiFi Access Points (APs), vehicle-to-vehicle communications, and cellular communications (2/3G). This mobile sensor approach greatly extends coverage, thus enabling *street-level traffic flow estimation*. CarTel [4] uses roadside WiFi APs, and TrafficView [55] and MobEyes [8] use car-to-car communications. Also, cellular communications have been recently considered: TruTraffic by Dash Express's SatNav system

and a smartphone-based system by Nokia Research. In particular, TruTraffic collects GPS measurement data from SatNav users using 2/3G and provides a real-time traffic information service such as dynamic route guidance and congestion notification.

3.2. Proactive urban surveillance

Vehicular sensing can be used for proactive urban monitoring services [8] where vehicles continuously sense events from urban streets, maintain sensed data in their local storage, autonomously process them, e.g., recognizing license plates, and possibly route messages to vehicles in their vicinity to achieve a common goal, e.g., to permit the police to track the movements of specified cars. Vehicular sensing could be an excellent complement to the deployment of fixed cameras/sensors. This completely distributed and opportunistic cooperation among sensor-equipped vehicles has the “deterrent” effect of making it harder for potential attackers to disable surveillance. Another less sensational but relevant example is the need to track the movements of a car, used for a bank robbery, in order to identify thieves, say. It is highly probable that some vehicles have spotted the unusual behavior of thieves' cars in the hours before the robbery, and might be able to identify the threat by “opportunistic” correlation of their data with other vehicles in the neighborhood. It would be much more difficult for the police to extract that information from the massive number of multimedia streams recorded by fixed cameras.

3.3. Vehicular safety warning services

Safe navigation support through wireless inter-vehicular communications has become an important priority and new standards are emerging such as IEEE DSRC/WAVE [14]. For instance, it can be used for forward collision warning and advisories to other vehicles about road perils (e.g., ice on bridge, congestion ahead). As an alternative, a safety warning service can be supported over 2/3G. Due to the large RTT, it is difficult to implement *time-critical* real-time safety warning messaging that requires less than several hundred millisecond responses such as notifying braking events to vehicles behind, yet non-time critical real-time safety warning messages could be delivered in a timely manner over 2/3G networks.

3.4. Ride quality monitoring

Municipalities around the world spend millions of dollars to keep roads in good ride quality [56]. Ride quality is mainly measured by the pavement roughness of a road surface that is an expression of the surface irregularity. The roughness causes vertical vibration of vehicles and thus reducing ride quality. There are several main sources of roughness: steps, dips, humps, bumps, and defects (cracks, potholes). It also causes the stress and fatigue damage of both vehicle and road structures. Municipalities have been profiling roads using non-contact profiling devices mounted on the vehicles that use GPS, accelerometer/laser sensors [57]. This approach is still quite

expensive due to the operation costs and limited availability of specialized profiling devices. Researchers have recently considered using less expensive commodity sensors; e.g., pothole detection using off-the-shelf GPS and accelerometer sensors [58]. Also smartphones can be treated as a good profiling platform as demonstrated in [7]. They may have less accurate sensors, yet this can be overcome, because a large number of people can participate in the monitoring project.

3.5. Location-aware micro-blogging

Micro-blogging is a form of multimedia blogging that allows users to send brief text updates or micromedia such as photos or audio/video clips and publish them, either to be viewed by anyone or by a restricted group which can be chosen by the user (e.g., Twitter) [59]. The content of a micro-blog differs from a traditional blog in that it is typically more topical, smaller in aggregate file size (e.g. text, audio or video). Gaonkar et al. [60] recently proposed location-aware micro-blogging where micro-blogs are superimposed on physical space based on device's location awareness (e.g., GPS). For instance, an individual's micro-blogs about a restaurant may float near the restaurant. Mobile users can sift through micro-blogs to find information of interest. Micro-blogging would be extremely useful in vehicular environments, providing means of sharing useful information while mobile users are behind the wheel; e.g., pictures of accidents, construction, or malfunctioning traffic signals, etc.

4. V2V-based VSN platforms

We review V2V-based VSN platforms, namely MobEyes [8–10], FleaNet [11], and VITP [12]. Both MobEyes and FleaNet use mobility-assisted data dissemination to facilitate information access. VITP uses the concept of geographic storage; i.e., the sensor information is stored in an area where it is generated, and mobile users pull it by sending a request to the area of interest.

4.1. MobEyes: proactive urban monitoring services

MobEyes aims at provisioning proactive urban monitoring services where vehicles continuously monitor events from urban streets, maintain sensed data in their local storage, process them (e.g. recognizing license plate numbers), and route messages to vehicles in their vicinity to achieve a common goal (e.g., to allow the police to find the trajectories of specific cars) [8–10]. However, this requires the collection, storage, and retrieval of massive amounts of sensed data. In conventional sensor networks, sensed data is dispatched to “sinks” and is processed for further use (e.g., Directed Diffusion [61]), but that is not practical in VSNs due to the sheer size of generated data. Moreover, it is impossible to filter data a priori because it is usually unknown which data will be of use for future investigations. Thus, the challenge is to find a completely decentralized VSN solution, with low interference to other services, good scalability, and tolerance to disruption caused by mobility and attacks.

4.1.1. MobEyes protocols

In MobEyes, each sensor node performs event sensing, processing/classification of sensed data, and periodically generates *meta-data* of extracted features and context information such as timestamps and positioning coordinates. Meta-data are then disseminated to other regular vehicles, so that mobile agents, e.g., police patrol cars, move and opportunistically harvest meta-data from neighbor vehicles. As a result, agents can create a low-cost opportunistic index which enables agents to query the completely distributed sensed data storage. This enables us to answer questions such as: (1) which vehicles were in a given place at a given time?; (2) which route did a certain vehicle take in a given time interval?; and (3) which vehicles collected and stored the data of interest?

4.1.1.1. Meta-data diffusion. Any regular node periodically advertises a packet with a set of newly generated meta-data to its current neighbors. Each packet is uniquely identified (generator ID + locally unique sequence number). This advertisement to neighbors provides more opportunities for mobile agents to harvest the meta-data packets. Note that the duration of periodic advertisement is configured to fulfill the desired latency requirements, because harvesting latency depends on it. Neighbors receiving a packet store it in their local meta-data databases. Therefore, depending on node mobility and encounters, packets are opportunistically diffused into the network. MobEyes is usually configured to perform “passive” diffusion: only the packet source advertises its packets. Two different types of passive diffusion are implemented in MobEyes: single-hop passive diffusion (packet advertisements only to single-hop neighbors) and k -hop passive diffusion (advertisements travel up to k -hops as they are forwarded by j -hop neighbors with $j < k$). MobEyes can also adopt other diffusion strategies, for instance single-hop “active” diffusion, where any node periodically advertises all packets (generated and received) in its local database at the expense of a higher traffic overhead. In a usual urban VANET, it is sufficient for MobEyes to exploit the lightweight k -hop passive diffusion strategy, with very small k values, to achieve the needed diffusion.

Fig. 4 depicts the case of a VSN node C1 encountering other VSN nodes while moving (for the sake of readability, only C2 is explicitly represented). Encounters occur when two nodes exchange meta-data, i.e., when they are within their radio ranges and have a new meta-data packet to advertise. In the figure dotted circles and timestamped triangles represent respectively radio ranges and C1 encounters. In particular, the figure shows that C1 (while advertising $S_{C1,1}$) encounters C2 (advertising $S_{C2,1}$) at time $T - t_4$. As a result, after $T - t_4$ C1 includes $S_{C2,1}$ in its storage, and C2 includes $S_{C1,1}$.

4.1.1.2. Meta-data harvesting. In parallel with diffusion, meta-data harvesting can take place. A mobile agent, e.g., a police patrol car, can request the collection of diffused meta-data packets by proactively querying its neighbor nodes. The ultimate goal is to collect all the meta-data packets generated in a given area. Obviously, a mobile agent is interested in harvesting meta-data packets it has

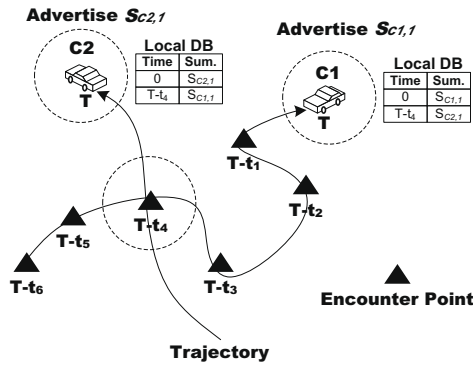


Fig. 4. MobEyes single-hop passive diffusion.

not collected so far. To focus only on missing packets, a mobile agent compares its already collected packets with the packet list at each neighbor (a set difference problem), by exploiting a space-efficient data structure for membership checking, i.e., a Bloom filter. A Bloom filter for representing a set of n elements consists of m bits, initially set to 0. The filter applies k independent random hash functions h_1, \dots, h_k to packet identifiers and records the presence of each element into the m bits by setting k corresponding bits. To check the membership of the element x , it is sufficient to verify whether all $h_i(x)$ are set. Thus, the harvesting procedure consists of the following steps. First, the police agent broadcasts a “harvest” request with its Bloom filter. Second, each neighbor prepares a list of “missing” packets from the received Bloom filter. Third, one of the neighbors returns missing packets to the agent. Fourth, the agent sends back an acknowledgment with a piggybacked list of just received packets. Upon listening or overhearing this, neighbors update their missing packet lists for the agent. Steps 3 and 4 are repeated until there are no missing packets.

4.1.2. Performance analysis

We evaluate MobEyes protocols via extensive ns-2 [62] simulations [8–10]. This section summarizes the most important results, with the goal of investigating MobEyes performance from the following perspectives: meta-data diffusion/harvesting speed in urban environments, and vehicle tracking application. In the simulations, we consider vehicles moving in a fixed region of size $2400\text{ m} \times 2400\text{ m}$. The default mobility model is Real-Track (RT) that models realistic vehicle motion in urban environments [63]. In RT, nodes move along virtual tracks, representing real accessible streets on an arbitrary loaded roadmap. For this set of experiments, we used a map of the Westwood area in the vicinity of the UCLA campus, as obtained by the US Census Bureau data for street-level maps [64] (Fig. 5). At any intersection, each vehicle randomly selects the next track it will run through; speed is periodically changed (increase or decrease) by a quantity uniformly distributed in the interval $[0, \pm \Delta s]$. Our simulations consider number of nodes $N = 100, 200, 300$, moving with average speed of $v = 5, 15, 25\text{ m/s}$. The meta-data advertisement period of regular nodes and the harvesting request period are kept constant as 3 s. We use the follow-

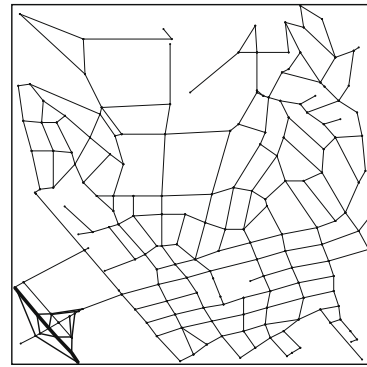


Fig. 5. Map of Westwood area in vicinity of UCLA campus.

ing communication model: MAC protocol IEEE 802.11, transmission band 2.4 GHz, bandwidth 11 Mbps, nominal radio range equal to 250 m, and Two-ray Ground propagation model.

4.1.2.1. Diffusion and harvesting performance. We investigate the regular node collection (diffusion) and mobile agent harvesting processes. We evaluate the cases with the number of nodes $N = 100/300$, and the average speed $v = 5/25$. Fig. 6 plots the cumulative distribution of meta-data collected by regular nodes as a function of time. The process highly depends on the average node speed; in fact, the speed determines to a large extent how quickly nodes “infect” other participants with their own meta-data. The results do not depend on node density. Fig. 7 plots the cumulative distribution of meta-data harvested by a police agent as a function of time. The results are mainly dependent on the speed, but unlike passive harvesting, the density plays an important role in active harvesting. Intuitively, if there are more neighbors, the agent has a higher chance of collecting an arbitrary meta-datum. In Fig. 8, we also plot the cumulative distribution of meta-data harvested by 1, 3 agents ($a \neq$) with $k = 1, 3$ relay hops. This estimation is useful to decide the tuning of the parameters (k -hop relay scope and number of agents) to address application requirements. Fig. 8 shows how the number of agents, the choice of the number of relaying hops k , and the average speed v of the nodes influence the process. In the

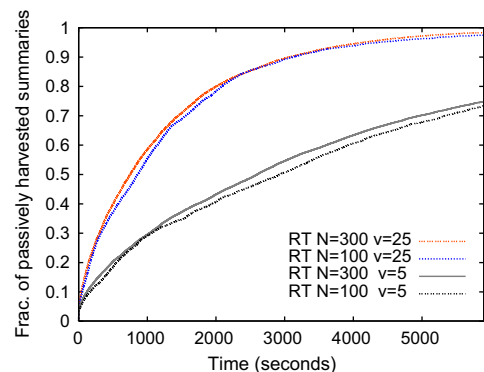


Fig. 6. Meta-data diffusion.

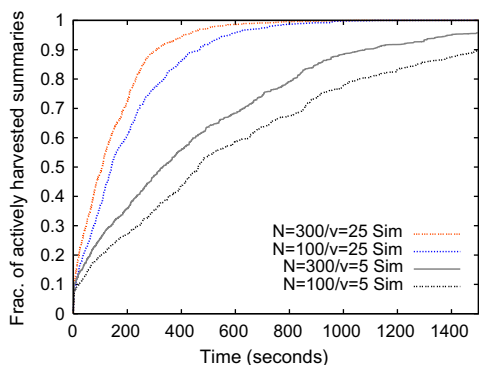


Fig. 7. Meta-data harvesting.

case of multiple agents, the harvesting process considers the union of the summary sets harvested by agents. The figure clearly shows that k -hop relay scope and multiple agents highly impact harvesting latency.

4.1.2.2. *Tracking application.* In the vehicle tracking application, an agent reconstructs a trajectory of an arbitrary node by exploiting the collected meta-data. To show this, we configure the regular node to generate new meta-data every $T = 120$ s and to continuously advertise the latest meta-data. Every meta-data includes the license plate number and position of the vehicle nearest to the meta-data originator at the generated time, tagged with a timestamp. The tracking application exploits the MobEyes diffusion protocol with $k = 1$ to spread the meta-data and deliver as much information as possible to a single agent scouting the ground. As the agent receives the meta-data, it extracts the information about node plates and positions, and tries to reconstruct node trajectories within the area. Its effectiveness is illustrated in Fig. 9 that visualizes the reconstructed trajectory.

4.2. Virtual information exchange bazaar

FleaNet [11] operates on the vehicular networks and provides an excellent method for people to communicate with each other as information traders and to efficiently

find matches of interest. Vehicles as well as static roadside Advertisement Stations (or Adstations) generate and propagate queries which can be either user generated content using on-board sensors (e.g., images, video clips, etc.) or commercial advertisements (e.g., special offers). Thus, a vehicle can be both a publisher and a subscriber of information. For example, a mobile user who takes pictures of an accident (publisher) can propagate the information to other vehicles nearby (subscriber); and a pizzeria (publisher) could advertise its special pizza offer to vehicles passing by and a driver (subscriber) who received the advertisement could place an order.

4.2.1. FleaNet query dissemination protocol

FleaNet utilizes mobility assisted query dissemination where the query “originator” periodically advertises its query only to 1-hop neighbors. Each neighbor then stores the query in its local database without any further relaying; thus, the query spreads only because of vehicle motion. Upon receiving a query, a node tries to resolve it locally in its database; in case of success, the originator will be automatically informed. A match only happens in its neighbors and thus, there is no redundant notification. Since this match could lead to a further interaction (e.g., downloading a file stored in the remote node), FleaNet provides a mechanism to deliver data using multi-hop inter-vehicular communications. A user could see multiple matches for a given query. Based on a user’s own criterion (either on distance from the current location), a mobile user selects the best one and sends a request message. Then, the target user responds with the reply after seeing the request. To support this, FleaNet uses Last Encounter Routing (LER) [65]. LER is based on geo-routing and combines location and routing services. In FleaNet the query packet includes the originator geo-coordinates, and thus, LER does not incur any initial flood search routing cost.

4.2.2. Performance analysis

We evaluate FleaNet performance through extensive simulations using ns-2 [11]. Each node is considered to have 802.11a connectivity with a bandwidth of 11 Mbps, 250 m radio range, and two-ray ground reflection model for radio propagation. For realistic mobility generation, we use VanetMobiSim that simulates macro- and micro-mobility

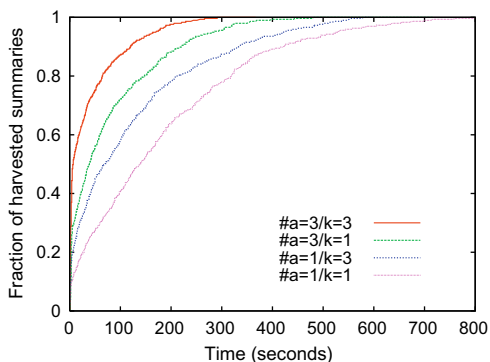


Fig. 8. Meta-data harvesting with multi-agents.

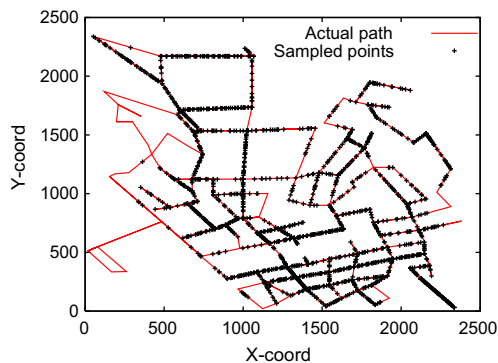


Fig. 9. Actual node trajectory (the unbroken line) vs. reconstructed sampled points.

patterns in urban environments [66]. Macro-mobility deals with road topology/structure and traffic signs (stop signs, traffic lights, speed limits), and micro-mobility models the speed and acceleration of each vehicle. We use the Westwood topology from Tigermap (TGR06037, Los Angeles) that represents the area in the vicinity of the UCLA campus (see Fig. 5). The simulations consider a vehicular network with the number of nodes between 100 and 400 in a 2400 m × 2400 m area. Vehicles travel with an average speed between 5 m/s and 25 m/s. To evaluate the performance, we mainly use the average latency of notification completion. For a given query, the *matching latency* measures the time for a node to receive a match, which happens when it encounters a node that carries a matching query. Then, a data request packet is sent to the query originator by using the Last Encounter Routing (e.g., from a publisher to subscriber, or vice versa) and the *routing latency* measures the time to deliver the notification to the destination. In our simulations, we assume that a user immediately sends a request to the other party. Thus, the total latency is the sum of matching and routing latency. The latency is dependent on many parameters, specifically on density/speed, query popularity, and mobility.

4.2.2.1. Impact of density/speed. Fig. 10 shows the latency as a function of density and speed in each node density with a single publisher. For each node density, a node is randomly selected as a subscriber to evaluate how node density affects the matching latency depending only on the speed. From the figure, we see that both the density and speed of vehicles are important factors in determining the latency. As the average speed or the number of nodes increases, the matching latency decreases. This is not surprising since as the average speed or the number of nodes increases, a node has a higher chance of meeting other nodes, which translates into more rapid dissemination of the query and also a higher chance to find a match.

4.2.2.2. Impact of query popularity. The overall latency is heavily dependent on the query popularity. We can easily see that if many people are interested in specific information, notification will quickly happen. To show this, we plot the latency as a function of popularity in a single-subscriber, *k*-publisher case in Fig. 11. We increase the ratio of publishers in the network from 5% to 25% with a gap

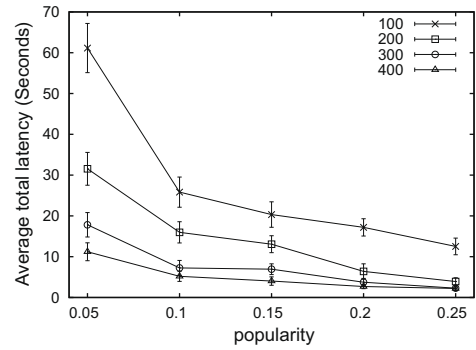


Fig. 11. Impact of query popularity.

of 5% on the X-axis and vary the number of nodes (i.e., 100–400 nodes). Given a single subscriber, *x*% publishers are randomly selected and we measure the latency for notification. We limit ourselves to the single subscriber case to clearly see the impact of query popularity on the latency. The figure clearly confirms our intuition about the impact of popularity on the latency: as the popularity increases, the latency decreases.

4.2.2.3. Impact of location. Nodes can be static, e.g., Adstation, in FleaNet. In this section, we show how a stationary node affects the average notification latency, i.e., the impact of its location. Intuitively, since the average relative speed of two nodes is higher if both move, a mobile node has a higher chance of meeting more nodes than a stationary node, which results in faster query dissemination. Restricted mobility in the Westwood mobility model makes the situation of a stationary node worse, because nodes tend to stay longer in the area where roads are densely clustered together. To understand how the position of a stationary node affects the latency, we use a scenario with 100 nodes moving at the average speed of 25 m/s. The 100 nodes are randomly selected in the map. Given the set of nodes, we generate 100 scenarios by adding a stationary node as a subscriber at the initial coordinate of each node. Thus, for a given subscriber (i.e., a scenario file) we perform 100 trials to get the average latency, by selecting each one of the other 100 nodes as a publisher. Queries are broadcast within three-hop neighbors, and no additional query adver-

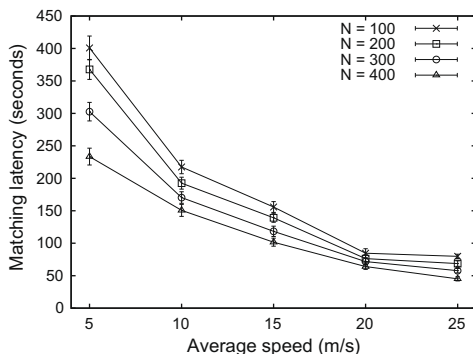


Fig. 10. Impact of density/speed.

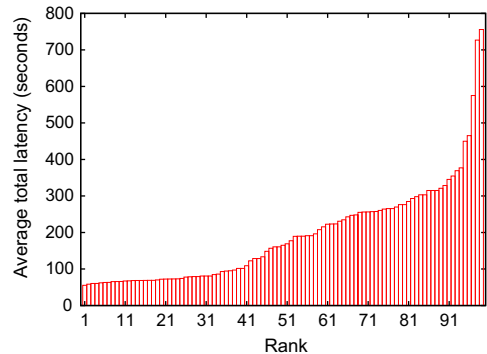


Fig. 12. Impact of location.

tiser is applied on the scenario. In Fig. 12, the total latency distribution is presented in ascending order with the rank of the delay on the X-axis. The i th index of the X-axis represents a node with i th largest latency. For example, the smallest latency (rank 1) is 55.4 s, while the largest latency (rank 100) is 755.9 s. As we can see, the largest latency is 13.6 times higher than the smallest one, and thus, the latency heavily depends on the location. By examining the location of the static node in each experiment we found that the first 10% stationary nodes (Rank 1 to 10) are located on the south west side of the map where the entrance to the highway is located. In fact, there is high traffic around the high entrance and the loop to the highway in the scenario. Therefore, it is likely that the stationary node in the area has the highest probability of encounter with other nodes. On the other hand, the last 10% stationary nodes (Ranks 91–100) are placed either at the border of the map or in the northwest area where roads are sparse.

4.2.2.4. Impact of churning. We measure the latency of stationary nodes in presence of churning where nodes move out of the network area (or out of the region of interest). When a node reaches the border area of the network (100 m width), we reset a node's buffer with probability p . The higher the probability, the higher is the churning rate. We vary the reset probability p from 0 to 0.6 with a gap of 0.2 to show the impact of churning. Fig. 13 shows the delay distribution using a box-and-whisker plot. As shown in the graph, the median value increases as the reset probability increases. The median increases from 165.34 s to 319.79 s as the reset probability increases to 0.6. Similarly, the maximum delay with 0.6 reset probability is about 196 s larger than that without churning. By manually examining the results, we find that some stationary nodes located at the border area have a noticeable delay increment of 22.6%. In our earlier section, we show that stationary nodes with high delay are placed either at the border of the map or in the northwest area where roads are sparse. Nodes visit these sites less frequently than the other sites due to the road layout. Moreover, those nodes traveling near the border area are more likely to suffer from churning. Thus, stationary nodes near the border area have much higher delay as churning increases.

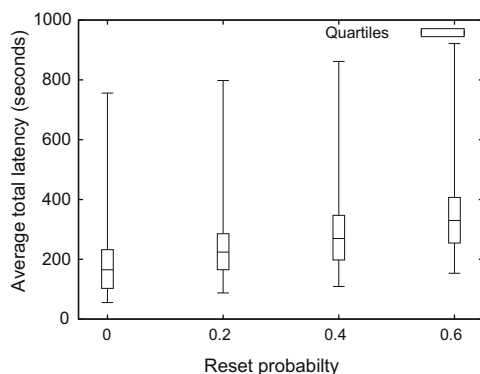


Fig. 13. Impact of churning.

4.3. Other related works

There are several other related works, namely geographic storage, and content distribution. In geographic storage, publishers store information in a geographic region from which mobile users can pull information on-demand. Efficient content distribution is necessary when the size of sensor data is too large and it has to be distributed to a number of users.

4.3.1. Geographic storage

Vehicular Information Transport Protocol (VITP) [12] uses the concept of geographic storage to support various on-demand location aware services such as traffic conditions (e.g., congestion, traffic flows), traffic alerts (e.g., accidents), and roadside service directories (e.g., location/menu of a local restaurant). In geographic storage, the information can be retrieved in a geographic area where it is generated, which is different from a Geographic Hash Table (GHT) where a data item is hashed and is stored in an arbitrary position [67]. VITP is an application-layer, stateless communication protocol that specifies the syntax and semantics of messages carrying location-sensitive queries and replies between the nodes of a VANET. One of the key features of VITP is that it allows nodes to aggregate (or summarize) location sensitive information and to report the summarized results to the requester. Fig. 14 shows the illustration of VITP operations. Node V wants to know the average vehicle speed near the gas station, and it sends the query Q to the destination location (i.e., dispatch phase). VITP-enabled peers around that destination location cooperatively resolve the query (i.e., computation phase) and return a reply R to the requester (i.e., reply-delivery phase).

4.3.2. V2V content distribution

For efficient content distribution, Nandan et al. [1] proposed SPAWN, a BitTorrent-like file swarming protocol in a VANET. In SPAWN, a file is divided into pieces and is uploaded into a road side stationary server or a mobile node. Each file has a unique ID (e.g., hash value of the file content), and each piece has a unique sequence number. Nodes cooperatively exchange missing pieces. By extending SPAWN, Lee et al. [68] proposed CarTorrent. Given that proximity is the key factor of peer selection, CarTorrent uses k -hop limited probabilistic gossiping and Closest-Rarest First for piece/peer selection. CarTorrent uses a cross-layer approach in that route discovery of underlying on-demand protocols is utilized for gossiping. Lee et al. [69] proposed CodeTorrent, a network coding based content distribution protocol based on the fact that network coding can mitigate the rare piece problem [70,71].

5. Infrastructure-based VSN platforms

In this section, we review Internet-based VSN platforms, namely Senster [13] and CarTel [4].

5.1. Senster: A mobile platform for scalable vehicular sensing

It is a promising option to realize vehicular sensing using smartphones now on the market, such as the iPhone

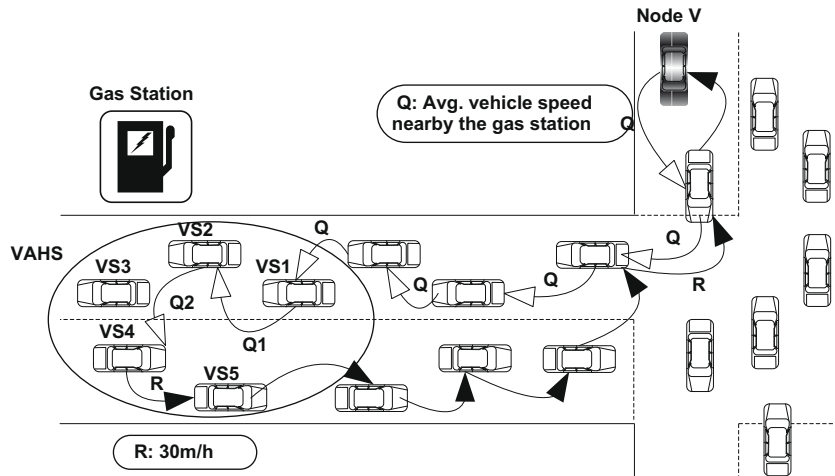


Fig. 14. An Illustration of VITP.

and Nokia N95. These phones are equipped with onboard sensors (e.g., voice, image, video, accelerometer, GPS) and have “always-on” mobile Internet connections via 2.5/3G. External sensors wirelessly interface to phones via Bluetooth or WiFi. Given the fact that there were 64 million 3G devices in the US in June 2008 [72], millions of users could simultaneously use the system. Thus, there is a need for large scale “distributed” storage that facilitates information sharing among millions of mobile users via always-on 2/3G connections.

A promising design option is to use a mobile-to-mobile overlay network of 2/3G users [22]. However, P2P connections between mobile devices are not always possible due to Network Address Translation (NAT). NATing is a commonly used technique in the mobile operator’s domain due to lack of IP addresses [73]. Although there are several known techniques to overcome NATing [74], they are generally not applicable to mobile clients due to complex packet processing in the mobile operator’s domain [73]. For P2P, we need additional services such as Session Initiation Protocol (SIP) or P2P proxies [73,75]. Another potential problem is that 2/3G cellular networks suffer from very large round trip delay (around several hundred milliseconds [17]); protocol operations in a large-scale P2P overlay may cause an intolerable delay in highly mobile vehicular environments.

In [13], we proposed a two-tier sensor storage Senster that exploits the Internet infrastructure. We assume that users install Senster clients in both PCs/laptops and smart phones. Internet clients on PCs/laptops provide a distributed P2P sensor storage over the Internet, through which mobile clients can publish/access sensor data (See Fig. 15). Senster considers the following design goals: localized publish/lookup to minimize the communication overhead; dynamic load balancing to handle heterogeneous vehicular density; complex query support such as range queries, and; data access among mobile users.

5.1.1. Senster protocols

5.1.1.1. SenterKBR: Key Based Routing. Senster uses Mercury DHT [76] for Key Based Routing. Mercury DHT relies

on Symphony DHT [77] that guarantees logarithmic properties of DHT operations by maintaining k random long links, inspired by the small world phenomenon. Recall that CAN [78] requires us to increase the dimension of key space to $\Theta(\log n)$ in order to guarantee logarithmic properties. However, this is inappropriate to our 2D vehicular sensing scenario because the lookup/update cost is $O(\sqrt{n})$.⁷

As shown in Fig. 16, we consider the 2D urban area for vehicular sensing. In Senster, we predefine the size of a grid and discretize the 2D space, by linearizing the space using a Hilbert space filling curve (i.e., each grid point is numbered). We assume that Senster-enabled nodes have the Hilbert space mapping a priori. For any geographic location, a node can find the associated geographic grid ID (or geo-ID) on the Hilbert curve. Unlike traditional DHTs where “consistent hashing” is used to map both node ID and keywords to a key space (e.g., CAN, Chord), Senster uses a geographic ID in the Hilbert curve as a key space [79,76]. Every sensor data generated is tagged with the geo-ID, and is stored on the Internet server that handles the corresponding geographic ID. This enables us to efficiently support range queries without maintaining other data structures as in [80].

Unicast routing (or geographic unicast routing) is quite straight-forward using Mercury DHT. Since SensterDB may receive a geographic region query (e.g., average speed in a certain region), SensterKBR provides geographic multicast as well. Given the region information, each node can translate the shape and can find a set of ordered linear segments in the Hilbert curve space. Geographic multicast can be efficiently supported by hopping through the linear segments. The search costs vary depending on the size of covered space. When the covered space is lower than

⁷ Not that Content Addressable Network (CAN) [78] dynamically divides d -dimensional Cartesian coordinate space among all the nodes in the system such that every node possesses at least one distinct zone within the overall space. CAN routes a packet in $O(dn^{1/d})$ hops; logarithmic hop can only be achieved when $d = \log_2 n/2$. When we directly map 2D space ($d = 2$), the routing cost becomes $O(\sqrt{n})$, failing to guarantee logarithmic routing.

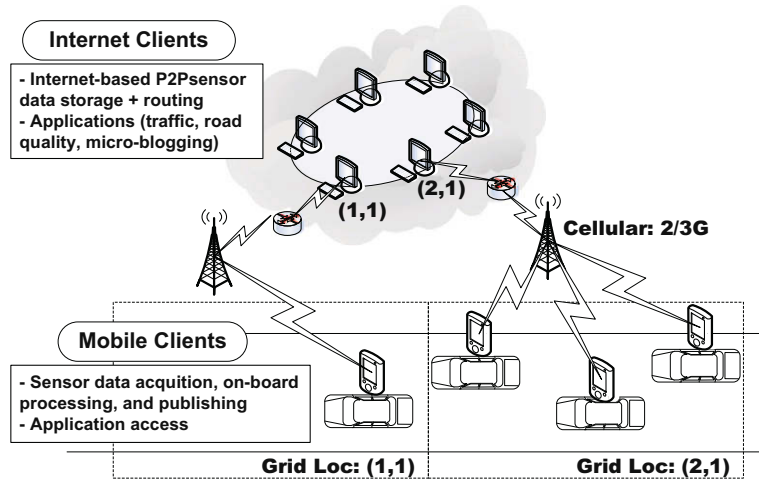


Fig. 15. Senster, a two-tier sensor storage framework.

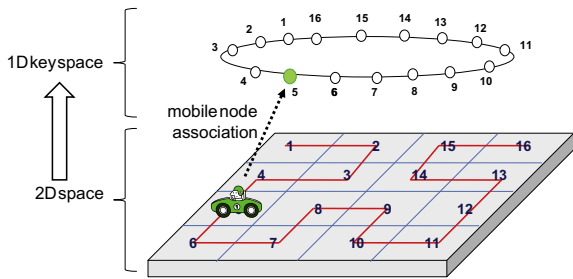


Fig. 16. Hilbert space filling example.

$1/\Theta(\log n)$ fraction of the key space, we can prove that the multicast cost is bounded by $O(\log n)$.

5.1.1.2. SensterDB: distributed database over SensterKBR. SensterDB manages the sensor data indexed by a certain key range to allow expressiveness and optimization using underlying SensterKBR. SensterDB supports a relational database model and a declarative query language (SQL). The current SensterDB prototype uses HSQLDB as a lightweight database [81]. The database can be extended to support stream data processing (e.g., periodic traffic information reporting). The key difference from conventional DBMS is that due to Internet scale database processing, we relax ACID requirements and perform best-effort (or approximate) query processing as in [82]. The data access patterns can be categorized as either spatial or temporal filtering. Spatial filtering occurs as spatial range queries are frequent, for example, finding traffic patterns along the commuting path. Temporal filtering occurs in cases where time is used for filtering values from a given area. These queries can also be one-shot or continuous. Simple approximate continuous query functionality can be provided by issuing a one-shot query at regular time intervals. SensterDB interacts with SensterKBR for key space management. Whenever there is a change in the key space (due to node join/leave), SensterKBR invokes

an UPDATE upcall to notify the change to all the applications, including SensterDB. As in KBR [83], SensterKBR exports the routing status information (e.g., current key range, neighbor set) to the upper layer via several access functions.

5.1.1.3. SensterMobile: mobile client. SensterMobile provides a common software architecture to interact with SensterKBR and to enable sensor data acquisition from different mobile platforms. In Fig. 17, we present the design of the SensterMobile architecture that allows efficient control of and transparent access to sensing resources. For each service, we define a set of filter applications running on the mobile client. For instance, traffic information services may require installing filters for reporting the average speed per road segment, and the braking pattern information (by processing the accelerometer data). After processing, the filters publish the summary data via the SensterKBR mobile client that provides transparent data delivery using standard send/receive interfaces. These filters acquire sensor data using a SQL-like sensor access

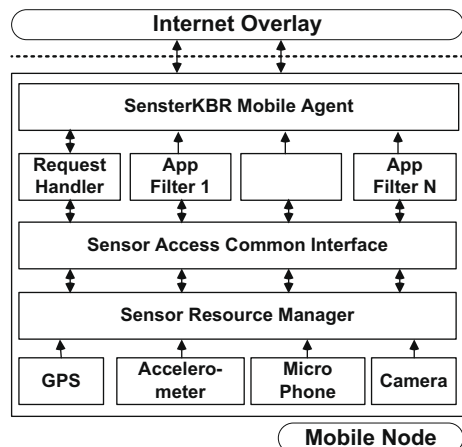


Fig. 17. SensterMobile software framework.

query language similar to the one defined in [4]. For instance, a filter requires access to sensor 1 and sensor 2 under specific conditions of sensor1 data value less than 10 and sensor 2 value greater than 30 for every 5 s; e.g., *select * from sensor1, sensor2 where sensor1 < 10 and sensor2 > 30 interval 5 s*. SQL provides predicates that allow detailed sensor configuration depending on the sensor type. The sensor resource manager takes these SQL sensor access queries and configures the sensors. Raw sensor data are then stored in its local storage, and query results are reported to the filters according to their requested rates. External nodes can also access the raw data from SensterDB. A remote request is delivered to the remote request manager. It then sends the request to the resource manager and returns the results via the SensterKBR mobile client.

5.1.2. Performance analysis

We evaluate the performance of content-based routing using extensive simulations. In particular, we evaluate the following: (1) the importance of load balancing in a very large-scale network scenario, (2) the locality of content-based routing, (3) the impact of different region sizes on the routing cost, and (4) the impact of mobility patterns and grid sizes on hand-off overhead. We implement an event-driven discrete-time simulator where each application level hop takes a unit time. For the sake of a large network simulation, our simulator does not model any queueing delay at intermediate nodes or packet loss on links. The simulator is implemented in C# (.NET framework 3.5) and provides a GUI for dynamic node generation/join/leave, load balancing, and multicasting features.

The number of long links in the Mercury DHT is set to five, as recommended in Symphony DHT [77]. We discretize the network area into grids for the Hilbert curve-based linearization: from 64×64 to 256×256 . For realistic mobility generation, we use VanetMobiSim that simulates macro- and micro-mobility patterns in urban environments [66]. The network area size ranges from $3200 \text{ m} \times 3200 \text{ m}$ and $12,800 \text{ m} \times 12,800 \text{ m}$. The Westwood topology from Tigermap (TGR06037, Los Angeles) represents the area in the vicinity of the UCLA campus. We use the random trip generation module in VanetMobiSim: each vehicle randomly picks starting and ending points in the map and moves along the shortest path, yet considering traffic congestion.

5.1.2.1. Load balancing. Mobile nodes publish sensor data to the overlay nodes. We study how heterogeneous distribution of vehicles influences the overall load imbalance. We use the Los Angeles map to extract road topology information. The area size is $12,800 \text{ m} \times 12,800 \text{ m}$, centered at the UCLA campus. The northern parts of the area are residential (low road density), whereas the southern parts are commercial districts (high road density). We use the grid size of $50 \text{ m} \times 50 \text{ m}$. The area is composed of 256×256 grids. We simulate different numbers of mobile clients from 1000 to 5000 with a gap of 1000 nodes. The number of overlay nodes is fixed to 1000 nodes. We measure the total published data size per node and draw a box-plot in Fig. 18. The results show that the total data size increases linearly, as the number of mobile clients in-

creases. Also, the case without load balancing shows much higher variation as opposed to the case with load balancing. There are still minor variations in the case with load balancing. This is because the system load of SensterKBR becomes balanced after several iterations of leave/join-based load balancing operations. We then vary the number of overlay nodes from 1000 to 5000 nodes. The number of mobile clients is fixed to 10,000 nodes.

5.1.2.2. Locality of content-based routing. Content-based routing in Senster exploits the geographic locality, on top of logarithmic routing performance. The construction of the Hilbert curve shows that packet forwarding happens within the area of interest (usually, a small fraction of the entire key space). To clearly show this, we place a node at a grid point (0,0) and measure the cost of a remote query with the square area of size 3×3 , by changing its location from (0,0) to (253,253). The number of overlay nodes is set to 1000. In Fig. 19, we plot routing hop counts over the 256×256 grids. The figure shows that as distance from (0,0) increases, the hop counts also increases. The reason why it shows non-contiguous colors is that we lose 50% of locality, yet locality is preserved at the higher level thanks to the recursive construction property of the Hilbert curve.

5.1.2.3. Impact of different region sizes. We show the sensitivity of routing cost with different geocasting area sizes: 2×2 , 3×3 , 4×4 , and 8×8 . We vary the number of overlay nodes from 1000 to 5000 with a gap of 1000 nodes. In Fig. 20, we plot the average number of hop counts as a function of the number of overlay nodes. The average value comes from the previous simulation setting, by averaging the hop counts of all scenarios; i.e., by covering all the query regions from (0,0) to $(255-M, 255-M)$ where M is the side length of a region. The figure shows that as the number of overlay nodes increases, the average hop count also increases (logarithmically). The query size does not change the overall cost significantly. For small area queries (e.g., 2×2 and 3×3), it is likely that a single node covers the queried area, whereas for a slightly large query 8×8 , a few overlay nodes may be required, thereby increasing hop counts.

5.1.2.4. Hand-off overhead analysis. For hand-off analysis, we consider dense scenarios: a network area of size

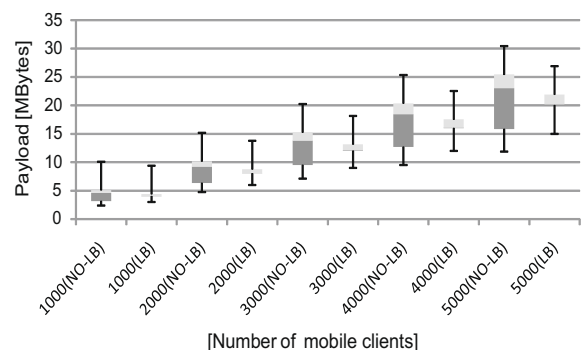


Fig. 18. Total published data size per overlay node.

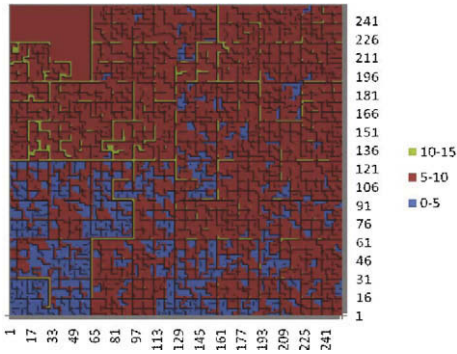


Fig. 19. Locality of content-based routing: routing cost from (0,0) to (X,Y).

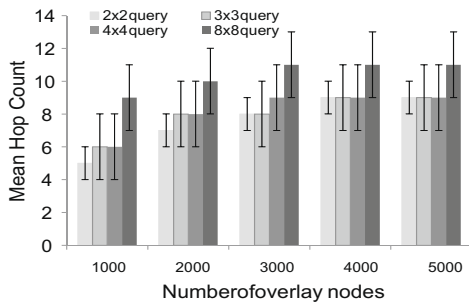


Fig. 20. Average hop counts with different region sizes.

3200 m × 3200 m with 1000, 2000, 3000 mobile clients. We set the maximum node speed to 10 m/s. We use grid sizes of 256 × 256 and the number of overlay nodes is fixed to 1000. Fig. 21 shows the average number of hop counts with different mobility patterns. As expected, the hop count increases as the number of mobile clients increases. The figure also shows the handoff cost is independent of the grid size, but depends on the number of overlay nodes. The handoff cost is mainly determined by the average key space per node.

5.2. CarTel: A distributed mobile sensor computing system

CarTel is a mobile sensor computing system designed to collect, process, deliver, and visualize data from sensors located on mobile units such as automobiles [4]. Unlike Sensor, each vehicle gathers and processes sensor data locally

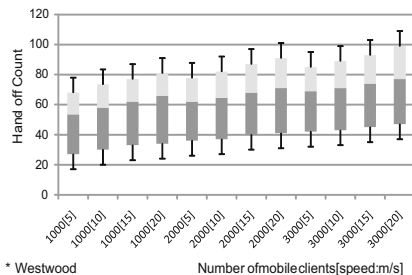


Fig. 21. Handoff overhead analysis.

before delivering them to a central portal, where the data is stored in a database for further analysis and visualization. CarTel provides a simple query-oriented programming interface that can handle large amounts of heterogeneous data from sensors. CarTel nodes rely primarily on opportunistic wireless (e.g., Wi-Fi hotspots) connectivity to the Internet. For delay tolerant data delivery, mobile nodes receive the data packets, hold them in storage, and wait for opportunities to transfer the data to remote destinations (called CafNet protocol). CarTel applications run on an Internet portal that uses a delay tolerant continuous query processor, called ICEDB, to specify how the mobile nodes should summarize, filter, and dynamically prioritize data. Fig. 22 shows the overall system architecture. Several applications were proposed such as road traffic analysis (e.g., commute time calculation), wide-area WiFi measurement, and automotive diagnostic using OBD-II output.

5.2.1. Intermittently connected DB (ICEDB)

ICEDB has a server component (Internet portal side) and a client component (mobile side). ICEDB server maintains a list of continuous queries submitted by applications where queries are pushed to mobile nodes using CafNets, a delay tolerant data delivery protocol. ICEDB client processes sensed data and returns the query results using CafNet. Results from ICEDB clients are stored in RDBMS of the portal. ICEDB uses an adaptor that defines a sensor type and schema. For example, CarTel has adapters for GPS, OBD-II, Wi-Fi, and cameras. The key innovation of ICEDB is that since delivering data in FIFO order is suboptimal in bandwidth-constrained vehicular networks, ICEDB implements prioritization. To be precise, local prioritization determines the priority within a given query buffer. For global prioritization, ICEDB clients send the summary results to the portal; and the portal applies a customized order function to order results and return the customized prioritization back to the clients.

5.2.2. Carry-and-forward network (CafNet)

CafNet is a general-purpose network stack for delay tolerant communications that enables applications to send

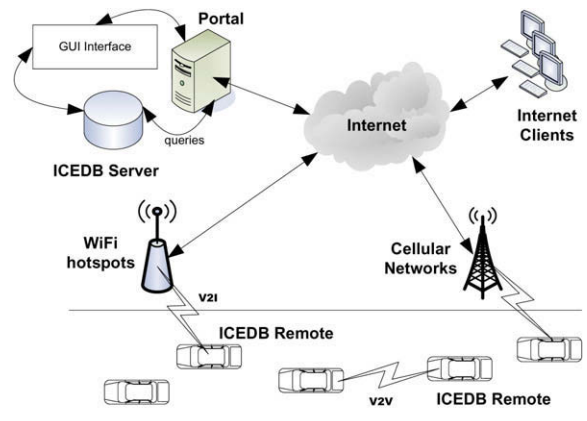


Fig. 22. CarTel system architecture.

messages across an intermittently connected network. CafNet offers a UDP-style message-oriented data transmission and reception API to applications. The major distinction from the traditional socket is that the transport layer does not have a buffer because the FIFO style buffer ordering may not be well suited for delay tolerant data transfer. The transport layer only informs the applications when connectivity is available or when network conditions change. Thus, applications can decide what data to send whenever there is a data transfer opportunity. CafNet defines a three-layer protocol stack, namely CafNet Transport Layer (CTL), CafNet Network Layer (CNL), and Mule Adaptation Layer (MAL). The transport layer, CTL, provides connectivity change notification to the application. CTL API defines a call back function; i.e., *cb_get_adu()* causes the application to synchronously return application data for immediate transfer. The network layer, CNL, currently implements simple static routing and flooding-based routing. Located beneath CNL, the MAL supports connectivity discovery such as WiFi and Bluetooth and notifies connectivity changes to the upper layers by invoking a callback function.

6. Conclusion

We surveyed recent vehicular sensor network proposals and reviewed some of the key results by carefully examining their evaluation methodologies to gain better insight into vehicular sensor network design and evaluation. We showed that underlying vehicular wireless access methods (e.g., DSRC, 2/3G, mixture) mainly determine the VSN architecture, which can be classified as either V2V-based or infrastructure-based techniques. Next, we reviewed V2V-based techniques such as mobility-assisted dissemination (MobEyes, FleaNet) and geographic storage (VITP), and; infrastructure-based techniques that facilitate distributed mobile node access via centralized Internet storage (CarTel) and distributed Internet storage (Senster). Our survey showed that the VSN system performance is mainly influenced by several factors, including: wireless access methods, vehicle mobility (density, speed, and churning), location of stationary users, and popularity of information.

References

- [1] A. Nandan, S. Das, G. Pau, M. Gerla, M.Y. Sanadidi, Co-operative downloading in vehicular ad-hoc wireless networks, in: WONS'05, St. Moritz, Switzerland, 2005.
- [2] Q. Xu, T. Mak, J. Ko, R. Sengupta, Vehicle-to-vehicle safety messaging in DSRC, in: VANET'04, Philadelphia, PA, USA, 2004.
- [3] U. Lee, J.-S. Park, E. Amir, M. Gerla, FleaNet: a virtual market place on vehicular networks, in: V2VCOM'06, San Francisco, CA, USA, 2006.
- [4] B. Hull, V. Bychkovsky, K. Chen, M. Goraczko, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, S. Madden, CarTel: a distributed mobile sensor computing system, in: SenSys'06, Boulder, CO, USA, 2006.
- [5] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M.B. Srivastava, Participatory sensing, in: WSW'06, Boulder, CO, USA, 2006.
- [6] M. Mun, S. Reddy, K. Shilton, N. Yau, P. Boda, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, PEIR, the personal environmental impact report, as a platform for participatory sensing systems research, in: MobiSys'09, Karaków, Poland, 2009.
- [7] P. Mohan, V. Padmanabhan, R. Ramjee, Nericell: rich monitoring of road and traffic conditions using mobile smartphones, in: SenSys'08, Raleigh, NC, 2008.
- [8] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, A. Corradi, MobEyes: smart mobs for urban monitoring with vehicular sensor networks, IEEE Wireless Communications 13 (5) (2006) 51–57.
- [9] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, A. Corradi, Dissemination and harvesting of urban data using vehicular sensor platforms, IEEE Transaction on Vehicular Technology 58 (2) (2009) 882–901.
- [10] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, P. Lio, K.-W. Lee, Bio-inspired multi-agent data harvesting in a proactive urban monitoring environment, Elsevier Ad Hoc Networks Journal, Special issue on Bio-Inspired Computing and Communication in Wireless Ad Hoc and Sensor Networks 7 (4) (2009) 725–741.
- [11] U. Lee, J.-S. Park, E. Amir, M. Gerla, FleaNet: a virtual market place on vehicular networks, in: V2VCOM'06, San Jose, CA, 2006.
- [12] M.D. Dikaiakos, S. Iqbal, T. Nadeem, L. Iftode, VITP: an information transfer protocol for vehicular computing, in: VANET'05, Cologne, Germany, 2005.
- [13] J.H. Ahn, U. Lee, H.J. Moon, M. Gerla, Senster: scalable smartphone based vehicular sensor networking systems, in: HotMobile'09, Santa Cruz, CA, 2009.
- [14] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, R. Herrtwich, Design of 5.9 GHz DSRC-based vehicular safety communication, IEEE Wireless Communications 13 (5) (2006) 36–43.
- [15] R. Frenkiel, B. Badrinath, J. Borras, R.D. Yates, The infostations challenge: balancing cost and ubiquity in delivering wireless data, IEEE Personal Communications 7–2 (2002) 66–71.
- [16] T. Small, Z.J. Haas, The shared wireless infostation model – a new ad hoc networking paradigm (or Where There is a Whale, There is a Way), in: MobiHoc'03, Annapolis, Maryland, USA, 2003.
- [17] A. Qureshi, J. Carlisle, J. Guttag, Tavarua: Video streaming with WWAN striping, in: Multimedia'06, Santa Barbara, CA, 2006.
- [18] W.L. Tan, O. Yue, Measurement-based performance model of IP traffic over 3G networks, in: TENCON 2005 IEEE Region 10, Melbourne, Australia, 2005.
- [19] M. Han, S. Moon, Y. Lee, K. Jang, D. Lee, Evaluation of VoIP quality over WiBro, in: Passive and Active Measurement Conference (PAM), Cleveland, OH, 2008.
- [20] D. Hadaller, S. Keshav, T. Brecht, S. Agarwal, Vehicular opportunistic communication under the microscope, in: MobiSys'07, San Juan, Puerto Rico, 2007.
- [21] M. Gerla, B. Zhou, Y.-Z. Lee, F. Soldo, U. Lee, G. Marfia, Vehicular grid communications: the role of the internet infrastructure, in: WICON'06, Boston, NY, 2006.
- [22] J. Rybicki, B. Scheuermann, W. Kiess, C. Lochert, P. Fallahi, M. Mauve, Challenge: peers on wheels – a road to new traffic information Systems, in: MobiCom'07, Quebec, Canada, 2007.
- [23] M. Torrent-Moreno, D. Jiang, H. Hartenstein, Broadcast reception rates and effects of priority access in 802.11-based vehicular ad-hoc networks, in: VANET'04, Philadelphia, PA, USA, 2004.
- [24] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, Performance evaluation of safety applications over DSRC vehicular ad hoc networks, in: VANET'04, Philadelphia, PA, USA, 2004.
- [25] T. ElBatt, S.K. Goel, G. Holland, H. Krishnan, J. Parikh, Cooperative collision warning using dedicated short range wireless communications, in: VANET'06, Los Angeles, CA, USA, 2006.
- [26] G. Korkmaz, E. Ekici, F. Ozguner, U. Ozguner, Urban multi-hop broadcast protocols for inter-vehicle communication systems, in: VANET'04, Philadelphia, PA, USA, 2004.
- [27] K. Tang, M. Gerla, MAC reliable broadcast in ad hoc networks, in: MILCOM'01, Washington, DC, 2001.
- [28] M.-T. Sun, L. Huang, A. Arora, T.-H. Lai, Reliable MAC layer multicast in IEEE 802.11 wireless networks, in: ICCP'02, Vancouver, 2002.
- [29] R.M. Yadumurthy, A. Chimalakonda, M. Sadashivaiah, R. Makanaboyina, Reliable MAC broadcast protocol in directional and omni-directional transmissions for vehicular Ad Hoc networks, in: VANET'05, 2005.
- [30] Z.D. Chen, H. Kung, D. Vlah, Ad hoc relay wireless networks over moving vehicles on highways, in: MobiHoc'01, Long Beach, CA, USA, 2001.
- [31] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, MaxProp: routing for vehicle-based disruption-tolerant networks, in: INFOCOM'06, Barcelona, Spain, 2006.
- [32] J. Zhao, G. Cao, VADD: vehicle-assisted data delivery in vehicular ad hoc networks, in: INFOCOM'06, Barcelona, Spain, 2006.
- [33] V. Naumov, T.R. Gross, Connectivity-aware routing (CAR) in vehicular ad hoc networks, in: INFOCOM'07, Anchorage, Alaska, 2007.
- [34] H. Wu, R. Fujimoto, R. Guensler, M. Hunter, MDDV: a mobility-centric data dissemination algorithm for vehicular networks, in: VANET'04, Philadelphia, PA, USA, 2004.

- [35] D. Sormani, G. Turconi, P. Costa, D. Frey, M. Migliavacca, L. Mottola, Towards lightweight information dissemination in inter-vehicular networks, in: VANET'06, Los Angeles, CA, USA, 2006.
- [36] C. Maihöfer, T. Leinmüller, E. Schoch, Abiding Geocast: time-stable geocast for ad hoc networks, in: VANET'05, Cologne, Germany, 2005.
- [37] U.C. Kozat, L. Tassiulas, Throughput capacity of random ad hoc networks with infrastructure support, in: MobiCom'03, San Diego, 2003.
- [38] B. Liu, Z. Liu, D. Towsley, On the capacity of hybrid wireless networks, in: INFOCOMM'03, San Francisco, CA, USA, 2003.
- [39] S. Lee, S. Banerjee, B. Bhattacharjee, The case for a multi-hop wireless local area network, in: INFOCOM'04, Hong Kong, 2004.
- [40] H. Luo, R. Ramjee, P. Sinha, L. Li, S. Lu, UCAN: a unified cellular and ad-hoc network architecture, in: MobiCom'03, San Diego, CA, USA, 2003.
- [41] M. Kodialam, T. Nandagopal, Characterizing the capacity region in multi-radio multi-channel wireless mesh networks, in: MobiCom'05, New York, USA, 2005.
- [42] M.J. Miller, W.D. List, N.H. Vaidya, A Hybrid Network Implementation to Extend Infrastructure Reach, Technical Report, UIUC CSD, January 2003.
- [43] J. Chen, S. Li, S.-H.G. Chan, J. He, WIANI: wireless infrastructure and ad-hoc network integration, in: ICC'05, Korea, 2005.
- [44] S.B. Eisenman, G.-S. Ahn, N.D. Lane, E. Miluzzo, R.A. Peterson, A.T. Campbell, MetroSense project: people-centric sensing at scale, in: WSW'06, Boulder, CO, USA, 2006.
- [45] O. Riva, C. Borcea, The urbanet revolution: sensor power to the people!, IEEE Pervasive Computing 6 (2) (2007) 41–49.
- [46] R.C. Shah, S. Roy, S. Jain, W. Brunette, Data MULEs: modeling a three-tier architecture for sparse sensor networks, Elsevier Ad Hoc Networks Journal 1 (2–3) (2003) 215–233.
- [47] Q. Li, D. Rus, Sending messages to mobile users in disconnected ad-hoc wireless networks, in: MOBIKOM'00, Boston, MA, USA, 2000.
- [48] Y. Wang, H. Wu, DFT-MSN: the delay/fault-tolerant mobile sensor network for pervasive information gathering, in: INFOCOM'06, Barcelona, Spain, 2006.
- [49] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, D. Rubenstein, Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet, in: ASPLOS'02, San Jose, CA, USA, 2002.
- [50] A. Woo, A new embedded web services approach to wireless sensor networks, in: SenSys'06, Boulder, Colorado, 2007.
- [51] S. Reddy, G. Chen, B. Fulkerson, S.J. Kim, U. Park, N. Yau, J. Cho, M. Hansen, J. Heidemann, Sensor-internet share and search – enabling collaboration of citizen scientists, in: DSI'07, Cambridge, MA, 2007.
- [52] S. Nath, J. Liu, F. Zhao, SensorMap for wide-area sensor webs, Computer 40 (7) (2007) 90–93.
- [53] P.B. Gibbons, B. Karp, Y. Ke, S. Nath, S. Seshan, IrisNet: an architecture for a worldwide sensor web, IEEE Pervasive Computing 2 (4) (2003) 22–33.
- [54] K. Aberer, M. Hauswirth, A. Salehi, A Middleware for fast and flexible sensor network deployment, in: VLDB'06, Seoul, Korea, 2006.
- [55] T. Nadeem, S. Dashtinezhad, C. Liao, L. Iftode, TrafficView: traffic data dissemination using car-to-car communication, ACM Mobile Computing and Communications Review (MC2R) 8 (3) (2003) 6–19.
- [56] Street Resurfacing and Pothole Repair Projects, <http://www.sfgov.org/site/budanalyst_page.asp?id=52980>.
- [57] Pavement Interactive Core: Roughness, <<http://pavementinteractive.org/index.php?title=Roughness>>.
- [58] J. Eriksson, L. Girod, B. Hull, R. Newton, H. Balakrishnan, S. Madden, The Pothole patrol: using a mobile sensor network for road surface monitoring, in: MobiSys'08, Breckenridge, Colorado, 2008.
- [59] Micro-blogging, <<http://en.wikipedia.org/wiki/Micro-blogging>>.
- [60] S. Gaonkar, J. Li, R.R. Choudhury, L. Cox, A. Schmidt, Micro-Blog: sharing and querying content through mobile phones and social participation, in: MobiSys'08, Breckenridge, Colorado, 2008.
- [61] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in: MobiCom'00, Boston, MA, USA, 2000.
- [62] ns-2 (The Network Simulator), <<http://www.isi.edu/nsnam/ns/>>.
- [63] B. Zhou, K. Xu, M. Gerla, Group and swarm mobility models for ad hoc network scenarios using virtual tracks, in: MILCOM'04, Monterey, CA, USA, 2004.
- [64] U.S. Census Bureau, TIGER, TIGER/Line and TIGER-Related Products, <<http://www.census.gov/geo/www/tiger/>>.
- [65] M. Grossglauser, M. Vetterli, Locating nodes with EASE: mobility diffusion of last encounters in ad hoc networks, in: INFOCOM'03, San Francisco, CA, USA, 2003.
- [66] J. Häiri, M. Fiore, F. Fethi, C. Bonnet, VanetMobiSim: Generating realistic mobility patterns for VANETs, in: VANET'06, Los Angeles, CA, 2006.
- [67] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, S. Shenker, GHT: a geographic hash table for data-centric storage, in: WSN'02, Atlanta, GA, USA, 2002.
- [68] K.C. Lee, S.-H. Lee, R. Cheung, U. Lee, M. Gerla, First experience with cartorrent in a real vehicular ad hoc network testbed, in: MOVE'07, Anchorage, Alaska, 2007.
- [69] U. Lee, J.-S. Park, J. Yeh, G. Pau, M. Gerla, CodeTorrent: content distribution using network coding in VANETs, in: MobiShare'06, Los Angeles, CA, 2006.
- [70] C. Gkantsidis, P. Rodriguez, Network coding for large scale content distribution, in: INFOCOM'05, Miami, FL, USA, 2005.
- [71] D.M. Chiu, R.W. Yeung, J. Huang, B. Fan, Can network coding help in P2P networks?, in: NetCod'06, Boston, MA, 2006.
- [72] Even Before iPhone, 3G Adoption Sharply Rose, <<http://blogs.eweek.com/applewatch/content/iphone/>>.
- [73] D. Kessens, T. Savolainen, 3G and IPv6 impact on battery life, in: French IPv6 Worldwide Summit, Cannes, France, 2006.
- [74] B. Ford, P. Srisuresh, D. Kegel, Peer-to-peer communication across network address translators, in: USENIX'05, Anaheim, CA, 2005.
- [75] S. Liu, W. Jiang, J. Li, Architecture and performance evaluation for P2P application in 3G mobile cellular systems, in: WiCom'07, 2007.
- [76] A.R. Bharambe, M. Agrawal, S. Seshan, Mercury: supporting scalable multi-attribute range queries, in: SIGCOMM'04, Portland, OR, 2004.
- [77] G.S. Manku, M. Bawa, P. Raghavan, Symphony: distributed hashing in a small world, in: USITS'03, Seattle, WA, 2003.
- [78] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, A scalable content-addressable network, in: SIGCOMM'01, San Diego, CA, USA, 2001.
- [79] N.J.A. Harvey, M.B. Jones, S. Saroiu, M. Theimer, A. Wolman, SkipNet: a scalable overlay network with practical locality properties, in: USITS'03, Seattle, WA, 2003.
- [80] Y. Chawathe, S. Ramabhadran, S. Ratnasamy, A. LaMarca, S. Shenker, J. Hellerstein, A case study in building layered DHT applications (PHT), in: SIGCOMM'05, Philadelphia, PA, 2005.
- [81] HSQLDB, <<http://hsqldb.org/>>.
- [82] R. Huesch, J.M. Hellerstein, N. Lanham, B. Thau, L.S. Shenker, I. Stoica, Querying the Internet with PIER, in: VLDB'03, Berlin, 2003.
- [83] F. Dabek, B. Zhao, P. Druschel, J. Kubiatowicz, I. Stoica, Towards a common API for structured peer-to-peer overlays, in: IPTPS'03, Berkeley, CA, 2003.



Uichin Lee received the B.S. degree in computer engineering from Chonbuk National University, Jeonju, Korea, in 2001, the M.S. degree in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2003, and the Ph.D. degree in computer science from the University of California at Los Angeles, in 2008.

He is currently a Member of Technical Staff with Bell Labs, Alcatel-Lucent, Holmdel, NJ. His research interests include distributed systems, mobile wireless networking systems,

and performance modeling/evaluation.



Mario Gerla received the Engineering degree from Politecnico di Milano, Milan, Italy, and the Ph.D. degree from the University of California at Los Angeles (UCLA).

From 1973 to 1976, he was with Network Analysis Corporation, Glen Cove, NY, where he helped transfer Advanced Research Projects Agency Network (ARPANET) technology to government and commercial networks. In 1976, he joined the UCLA Faculty. He is currently a Professor in computer science with UCLA, where he was part of the team that

developed the early ARPANET protocols under the guidance of Prof. L. Kleinrock. He has also designed and implemented network protocols, including ad hoc wireless clustering, multicast On-Demand Multicast Routing Protocol (ODMRP) CodeCast, and Internet transport (TCP West-

wood). He has lead the \$12 million six-year Office of Naval Research (ONR) MINUTEMAN project, designing next-generation scalable airborne Internet for tactical and homeland defense scenarios. He is now leading two advanced wireless network projects under Army and IBM funding.

His team is developing a vehicular testbed for safe navigation, urban sensing, and intelligent transport. A parallel research activity explores personal communications for cooperative networked medical monitoring (see <http://www.cs.ucla.edu/NRL> for recent publications).