Chapter 1

Emerging Vehicular Applications

Uichin Lee, Ryan Cheung, Mario Gerla
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095
{*uclee,rcheung,gerla*}*@cs.ucla.edu*

1.1 Introduction

Safe navigation support through wireless car-to-car and car-to-curb communications has become an important priority for car manufacturers as well as transportation authorities and communications standards organizations. New standards are emerging for car-to-car communications (DSRC and more recently IEEE 802.11p) [8]. There have been several well publicized testbeds aimed at demonstrating the feasibility and effectiveness of car-to-car communication safety; for instance, the ability to rapidly propagate accident reports to oncoming cars, the awareness of unsafe drivers in the proximity and the prevention of intersection crashes.

While safe navigation has always been the prime motivation behind vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, vehicular networks provide a promising platform for a much broader range of large scale, highly mobile applications. Given the automobile's role as a critical component in peoples' lives, embedding software-based intelligence into cars has the potential to drastically improve the user's quality of life. This, along with significant market demand for more reliability, safety and entertainment value in automobiles, has resulted in significant commercial development and support of vehicular networks and applications.

These emerging applications span many fields, from office-on-wheels to entertainment, mobile Internet games, mobile shopping, crime investigation, civic defense, etc. Some of these applications are conventional "mobile internet access" applications, say, downloading files, reading e-mail while on the move, etc. Others involve the discovery of local services in the neighborhood (e.g., restaurants, movie theaters, etc.) using the vehicle grid as an ad hoc network. Yet others demand close interaction among vehicles such as interactive car games.

To support more advanced services, new brands of functions must be deployed such as the creation/maintenance of distributed indices, "temporary" storage of sharable content, "epidemic" distribution of content and indices. Examples include the collection of "sensor data" in mobile vehicular sensor platforms, the sharing and streaming of files in a BitTorrent fashion, and the creation/maintenance of massively distributed databases with locally relevant commercial, entertainment, and cultural information (e.g., movies, hotels, museums, etc.). Typically, these applications are distributed and follow a P2P collaboration pattern.

In this chapter, we review such emerging applications. We address potential vehicular networking architectures by reviewing various wireless access methods such as DSRC, 3G, and WiMAX. We discuss the unique characteristics of vehicular communications and briefly analyze various VANET routing protocols that are essential to supporting applications. VANET applications are classified by the vehicle's role in managing data: as a data source, data consumer, source and consumer, or intermediary. Based on this, we review various emerging applications proposed in the research community.

1.2 Background

We overview various wireless communication methods available in vehicular networks, 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). Finally, we review various routing protocols to gain better insight into the application protocol design.

1.2.1 Wireless Access Methods in Vehicular Networks

DSRC/WAVE: Dedicated Short-Range Communication (DSRC) is a short to medium range communication technology operating in the 5.9 GHz range [8]. The Standards Committee E17.51 endorses a variation of the IEEE 802.11a MAC for the DSRC link. DSRC supports vehicle speeds up to 120mph, nominal transmission range of 300m (up to 1000m), and default data rate of 6Mbps (up to 27Mbps). 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 Vehicular 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 [14, 51]. Readers can find a detailed overview of the DSRC standards in [26].

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 9.6kbps. To provide higher rate data communications, GSM-based systems use GPRS (<171kbps) and EDGE (<384kbps), and IS-95-based CDMA systems use 1xRTT (<141kbps). Now 3G systems support much higher data rate.[1] UMTS/HSDPA provides maximum rates of 144kbps, 384kbps, and 2Mbps under high mobility, low mobility, and stationary environments respectively. CDMA2000 1xEvDO (Rev. A) provides 3Mbps and 1.8Mbps for down and up links respectively. The average data rate perceived by users is much lower in practice: <128kbps for GSM/EDGE and <512kbps 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 has evaluated by Qureshi et al. [44].[2] They reported that 1) the average RTT was consistently high (around 600ms) with high variance (ρ=350ms); 2) there were a small number of short-lived (<30s) disconnections during their experiments; 3) the download throughput varied, ranging from 100kbps to 420kbps, and the peak upload throughput was less than 140kbps; 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.

WiMAX/802.16e: 802.16e or WiMAX (Worldwide Interoperability for Microwave Access) aims at enabling the delivery of last mile wireless broadband access (<40Mbps) 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 (<60Km/h), and coverage (<10Km) required to deliver the Internet access to mobile clients. In its part, WiBro, developed in Korea based on 802.16e draft version 3, provides 1Km range communications at the maximum rate per user of 6Mbps and 1Mbps 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 application that requires a minimum guaranteed data rate. Han et al. [22] measured the performance of WiBro networks in a subway whose maximum speed is 90Km/h, and showed that 1) the average uplink and downlink speeds were 2Mbps and 5.3Mbps respectively, and 2) the average packet delay (half RTT) was less than 100ms, and almost all packets experienced delay below 200ms, except the case when handoffs happened (>400ms).

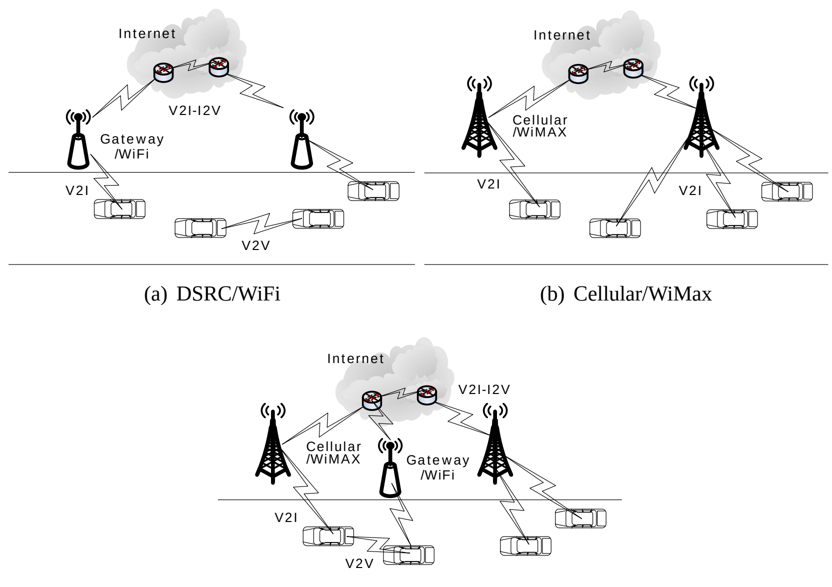
WLAN: WiFi or WLAN can also support broadband wireless services. 802.11a/g provides 54Mbps and has nominal transmission range of 38m (indoor) and 140m (outdoor). Despite its short radio range, its ubiquitous deployment makes WLAN an attractive method to support broadband wireless services. It has long 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

[1] `http://en.wikipedia.org/wiki/3G`

[2] Readers can find the evaluation of UMTS/HSDPA systems in a static environment [54].

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

[4] `http://en.wikipedia.org/wiki/Wardriving`



(a) DSRC/WiFi

(b) Cellular/WiMax

(c) DSRC/WiFi + Cellular/WiMax

Figure 1.1: Possible Wireless Vehicular Networking Scenarios

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 [21].

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 Figure 1.1(a). 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 [15]. 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 Internet as in Figure 1.1(b). For instance, people with iPhone 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 Figure 1.1(c). Researchers 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 [48]. Thus far, the third scenario has not yet received attention, but it has its potential to enable novel applications in the future.

[5] `http://meraki.com`

1.2.2 Characteristics of VANET 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, battlefield, etc.). These differences have a significant impact on application infrastructures.

- 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 gasoline or alternative fuel engine.

- 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 must account for easy access to the Internet during "normal" operations.

1.2.3 VANET Routing Protocols

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.

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 [56, 60]. Xu et al. [60] studied the impact of rapid repetition of broadcast messages on the packet reception failure in random access protocols. Torrent-Moreno et al. [56] showed channel access time and reception probability under deterministic and statistical channel models. Yin et al. [62] detailed the DSRC PHY layer model and incorporated the model into a VANET simulator to support generic safety application models. ElBatt et al. [10] 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) [28] 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) [55] and Batch Mode Multicast MAC (BMMM) [53] 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 [61].

Unicast routing: There are many MANET routing protocols: proactive routing (e.g., DSDV, OLSR) , 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 connectivity. 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. [5] 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 has demonstrated to achieve 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 [1], part of the UMass DieselNet project[6] 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, 3) packets with small hop-counts, and 4) packets with a high probability of being delivered through the other party. VADD [65] rests on the assumption that most node encounters happen in intersection areas. Effective decision strategies are proposed to reduce packet delivery failures and delay. Naumove et al. [41] 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.

Geocast: Applications for distributed data collection in VANET call for geographic dissemination strategies that deliver packets to all nodes belonging to target remote areas (or geo-casting), despite possibly interrupted paths [59, 52, 36]. MDDV [59] 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. [52] proposed several strategies based on virtual potential fields generated by propagation

[6] UMass' DieselNet. `http://prisms.cs.umass.edu/dome`

functions: 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 minima target zones are detected. Maihofer 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.

1.3 Vehicular Application Classification

The major departure of vehicle networks from conventional ad hoc networks is the opportunity to deploy, in addition to traditional applications, a broad range of innovative content sharing applications (typically referred to as Peer-to-Peer applications). While the popularity of P2P applications has been well documented, these applications have been thus far confined to the wired Internet (e.g., BitTorrent). The much increased storage and processing capacity of VANETs with respect to personal or sensor based ad hoc networks make such applications feasible. Moreover, the fact that car passengers are a captive audience provides incentive for content distribution and sharing applications at a scale that would be unsuitable to other ad hoc network contexts. We describe a representative set of VANET P2P applications and classify them by the vehicle's role in managing data: as a data source, data consumer, data source and consumer, or intermediary.

First, the vehicles provide an ideal platform for mobile data gathering especially in the context of monitoring urban environments (i.e., vehicular sensor networks) [32, 33, 31, 24, 12]. Each vehicle can sense events (e.g., images from streets or the presence of toxic chemicals), process sensed data (e.g., recognizing license plates), and route messages to other vehicles (e.g., forwarding notifications to other drivers or police officers). Because vehicular sensors have few constraints on processing power and storage capabilities, they can generate and handle data at a rate impossible for traditional sensor networks. These applications require persistent and reliable storage of data for later retrieval. In addition, they require networking protocols (including sophisticated query processing) to efficiently locate/retrieve data of interest (e.g., finding all the vehicles at a certain time and location).

Second, the vehicles can be significant consumers of content. Their local resources are capable of supporting high fidelity data retrieval and playback. For the duration of each trip, drivers and passengers make up a captive audience for large quantities of data. Examples include locality-aware information (map based directions) and content for entertainment (streaming movies, music and ads) [38, 35, 39, 4]. These applications require high throughput network connectivity and fast access to desired data.

In a third class of compelling applications, vehicles are both producers and consumers of content. Examples include services that report on road conditions and accidents, traffic congestion monitoring, and emergency neighbor alerts, e.g. my brakes are malfunctioning [7, 37, 34, 20, 43]. Also, interactive applications (e.g., voice over V2V and online gaming) belong to this category. These applications require location-aware data gathering/dissemination and retrieval. In particular, interactive applications require real-time communication among vehicles.

Finally, all of the above applications will need to rely on vehicles in an intermediary role. Individual vehicles in a mobile group setting can cooperate to improve the quality of the applicant experience for the entire network. Specifically, vehicles will provide temporary storage (caching) for others, as well as forwarding of both data and queries for data. In this capacity, they require reliable storage as well as efficient location of and routing to data sources and consumers.

The demands of these applications give us a list of requirements and challenges for vehicular applications. Note that we can leverage them to simplify the applications infrastructure.

- *Time sensitivity* – Time-sensitive data must be retrieved or disseminated to the desired location within a given time window. Failure to do so renders the data useless. This mirrors the needs of multimedia streaming across traditional networks, and we can leverage relevant research results from related areas.

- *Location awareness* – Both data gathered from vehicles and data consumed by vehicles are highly location-dependent. This property has direct implications on the design of data management and security components. Data caching and indexing should focus on location as a first order property; while data dissemination must be location-aware in order to maintain privacy and prevent tampering.

Most applications require methods of storing/retrieving such location/time sensitive information. As in MANETs, we can use structured approaches such as geographic hashing [45] and DHT [3], or structureless approaches such as epidemic dissemination [57]. However, it is non-trivial to maintain structure in VANETs due to the high mobility, non-uniform distribution of vehicles, and intermittent connectivity. Thus, most application protocols rely on variants of epidemic data dissemination such that the produced information disseminated to nodes in an area where the information is produced [7, 37, 4, 66, 34].

1.4 Data Source (Vehicular Sensor) Applications

Vehicular networks are emerging as a new network paradigm of primary relevance, for example for proactive urban monitoring using sensors and for sharing and disseminating data of common interest. In particular, we are interested in urban sensing for effective monitoring of environmental conditions and social activities in urban areas using vehicular sensor networks (VSNs). The major departure from traditional wireless sensor nodes is that vehicles are not strictly affected by the energy constraints and the size of sensor units. Thus, vehicles can easily be equipped with powerful processing units, wireless communication devices, GPS, and sensing devices such as chemical detectors, still/video cameras, and vibration/acoustic sensors. Figure 1.2 shows an application scenario.

1.4.1 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,

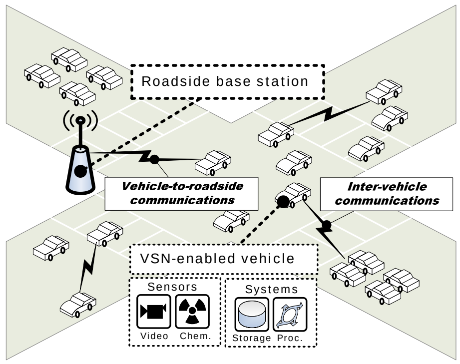


Figure 1.2: Vehicular Sensor Network(VSN)

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 police agents to find the trajectories of specific cars). 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 [25]), 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.

MobEyes is a novel middleware that supports VSN-based proactive urban monitoring applications [32, 33, 31]. 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 patrolling 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?

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 the 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-hop 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 needed diffusion.

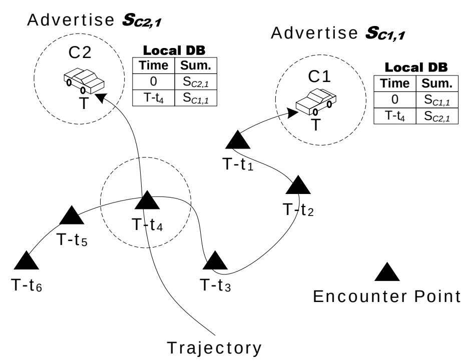


Figure 1.3: MobEyes single-hop passive diffusion

Figure 1.3 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 $S_{C1,1}$.

Meta-data Harvesting: In parallel with diffusion, meta-data harvesting can take place. A MobEyes police agent can request the collection of diffused meta-data packets by proactively querying its neighbor regular nodes. The ultimate goal is to collect all the meta-data packets generated in a given area. Obviously, a police agent is interested in harvesting meta-data packets it has not collected so far. To focus only on missing packets, a MobEyes 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 $h_1, \cdots, h_k$ to MobEyes 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.

Therefore, the MobEyes harvesting procedure consists of the following steps:

1. The police agent broadcasts a "harvest" request with its Bloom filter.

2. Each neighbor prepares a list of "missing" packets from the received Bloom filter.

3. One of the neighbors returns missing packets to the agent.

4. 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.

5. Steps 3 and 4 are repeated until there are no missing packets.

Note that Bloom filter membership checking is probabilistic. In particular, false positives may occur and induce MobEyes regular nodes not to send packets still missing to the agent. The probability of a false positive depends on m and n [13]. Nevertheless, in MobEyes, the agent can obtain a missing packet with high probability, because it is highly probable that other nodes have the packets as time passes, and the harvesting procedure is repeated as the agent moves. For example, in usual VSN deployment scenarios (e.g., with 10 neighbors on average), the probability of missing one packet due to false positives after repeating the procedure multiple times is very low.

1.4.2 Related Mobile Sensor Platform Projects

In CarTel [24], users submit their queries about sensed data on a portal hosted on the wired Internet. Then, an intermittently connected database is in charge of dispatching queries to vehicles and of receiving replies when vehicles move in the proximity of open access points to the Internet. Eriksson et al. [12] proposed a system called Pothole Patrol (P^2) that uses the mobility of vehicles, opportunistically gathering data from vibration and GPS sensors, to access road surface conditions. Yoon et al. [64] proposed a method of identifying traffic conditions on surface streets using the GPS location traces collected from vehicles.

In general, a vehicular sensor network can be considered as a form of "opportunistic mobile sensing platform." The opportunistic mobile sensing area has been extremely productive recently, providing a wealth of related work. ZebraNet [27] 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). SWIM [51] 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 whenever they encounter an Infostation. Eisenman et al. [9] proposed a three tier architecture called MetroSense: servers in the wired Internet are in charge of storing/processing sensed data; Internet-connected stationary Sensor Access Points (SAPs) act as gateways between servers and mobile sensors (MSs); MSs move in the field opportunistically delegating tasks to each other, and "muling" [49] data to SAP. MetroSense requires infrastructure support, including Internet-connected servers and remotely deployed SAP. Wang et al. [58] proposed data delivery schemes in Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN) for human-oriented pervasive information gathering. The trade-off between data delivery ratio/delay and replication overhead is mainly investigated in terms of buffer and energy resource constraints. CENS' Urban Sensing project [2] addresses "participatory" sensing where people of the same interest participate in an urban monitoring campaign. Intel IrisNet [16] and Microsoft SenseWeb [40] investigate the integration of heterogeneous sensing platforms in the Internet via a common data publishing architecture. Urbanet [47] proposes application programming models for opportunistic sensing.

1.5 Data Consumer Applications

1.5.1 Content Distribution

Content distribution to vehicles ranges from multi-media files to road condition data and to updates/patches of software installed in the vehicle. Nandan et al. [38] proposed SPAWN, a BitTorrent-like file swarming protocol in a VANET. In SPAWN, a file is divided into pieces and is uploaded into an Internet server. Each file has a unique ID (e.g., hash value of the file content), and each piece has a unique sequence number. Users passing by the Access Points (APs) download parts of the file. Once out of the range of APs, they cooperatively exchange missing pieces as in Figure 1.4.

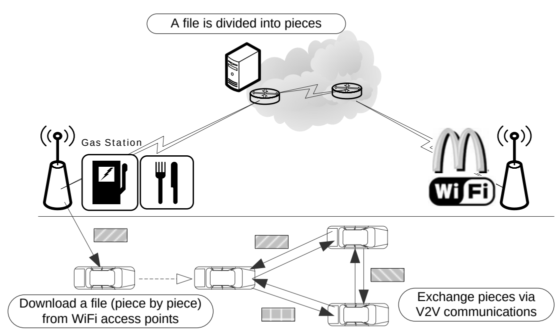


Figure 1.4: Cooperative file downloading in a VANET

SPAWN is composed of the following components: peer/content discovery, and peer/content selection. Due to intermittent presence of APs, SPAWN cannot use a centralized server as in BitTorrent that keeps track of all the peers. Instead, SPAWN uses a decentralized "gossiping" mechanism for peer/content discovery that leverages the broadcast medium of the wireless networks. A gossip message of a node contains a file ID, a list of pieces that the node has, a hop-count, etc. For efficient gossiping, SPAWN uses the following gossiping methods, namely probabilistic spawn and rate-limited spawn. In the probabilistic spawn, nodes forward gossip messages with a certain probability, whereas in the rate-limited spawn, nodes forward gossip messages in their buffer with a certain rate, e.g., forwarding a random gossip message in the buffer every 2 seconds. The hop-count of a gossip message is incremented, whenever a gossip message is forwarded. For a given file, there are three types of users in the network: those who are interested in downloading the files, those who are uninterested in downloading the files, and those who do not understand the SPAWN protocol. These roles are considered in the gossiping. For instance, interested users may have a higher probability of packet forwarding than uninterested users.

After the peer/content discovery, a node has to select a peer to download a piece. Given that TCP connections spanning fewer hops perform better in multi-hop wireless networks, SPAWN uses proximity-driven piece/peer selection strategies where the proximity is estimated by the hop-count in the gossip messages: (1) Rarest-Closest First chooses the rarest piece among all the peers in one's peer list, and breaks the tie based on proximity, and (2) Closest-Rarest First selects the rarest piece among all the closest peers. Recall that BitTorrent uses a rarest piece first selection strategy where the rarest piece among all the peers in its list is selected. After peer selection, the node finally downloads pieces by setting up a TCP connection. Any routing protocol such as AODV and DSR can be used for this purpose.

By simplifying SPAWN, Lee et al. [29] proposed CarTorrent. Given that proximity is the key factor of peer selection, CarTorrent uses k-hop limited probabilistic gossiping and Closest-Rarest First is used 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. [35] proposed CodeTorrent, a network coding based content distribution protocol. Recall that BitTorrent-like protocols suffer from a coupon collection problem; i.e., as a node collects more pieces, it will take progressively longer time to collect a new piece. It is known that network coding can mitigate this problem [17, 6].

Eriksson et al. [11] proposed techniques to improve data delivery throughput. QuickWiFi, a streamlined WiFi client, reduces the end-to-end link establishment delay to a WiFi access point, and Cabernet Transport Protocol (CTP) improves the data throughput by differentiating congestion in wired links and packet loss in wireless links. Recently, Yoon et al. [63] proposed Mobile Opportunistic Video-on-demand (MOVi), a mobile peer-to-peer video-on-demand application. Since switching WiFi modes (between infrastructure and ad hoc modes) takes time, MOVi exploits the opportunistic mixed usage of roadside WiFi access points and direct peer-to-peer communications using Direct Link Service (DLS) in 802.11 standards that enables direction communications between nodes within a single BSS.

1.5.2 Location-aware Advertisements

Advertisements are one of the most important sources of revenue for Internet-based companies. Similarly, cars with wireless communications will become lucrative targets for advertisements. As an extension of the physical billboards, Nandan et al. [39] proposed AdTorrent that delivers location sensitive commercial advertisements to the vehicles using digital billboards (or Ad Stations). With AdTorrent, business owners in the vicinity of the billboards can subscribe to digital billboards. Advertisements include simple text-based ads or multimedia ads, for example, trailers of movies playing at the nearby theater, virtual tours of hotels in a 5 mile radius, or conventional television advertisements relevant to local businesses.

AdTorrent aims at allowing drivers to download the advertisements of interest. So, it has a location aware distributed mechanism to *search, rank* and *deliver* relevant advertisements. Each advertisement has meta-data information (e.g., keywords, advertisement ID). For keyword search, a client builds its own inverted index that links keywords to advertisements. Since it is expensive to disseminate the raw inverted index, AdTorrent uses a special hash table that is based on a Bloom filter, a space efficient membership checking data structure. Keywords are stored in a Bloom filter, and the resulting hash table is disseminated to k-hop neighbors. After receiving a set of Bloom filters, a node aggregates them by performing the logical AND operation on Bloom filters. To resolve a query, a node first searches its local index (i.e., its aggregated Bloom filter). When failing to retrieve ℓ advertisements (where ℓ is set by the query originator), the node tries to search more results via m-hop scoped flooding. After collecting ℓ matched advertisements, the node downloads all the meta-data information of those advertisements. The node ranks each advertisement based on its relevance, location, stability, etc., and it starts downloading the best advertisement using CarTorrent, a content distribution protocol in VANETs.

Note that some of the products advertised to cars may be very dependent on car navigation and management. Caliskan et al. [4] proposed a parking spot information dissemination protocol where infrastructure (e.g., parking meters) periodically broadcasts parking spot information, and vehicles disseminate this information via periodic single-hop broadcasting. For efficient dissemination, they used spatio-temporal characteristics of parking spot information. Related to this is the advertising of "lanes" (on a time shared basis) to vehicles interested in bidding to the service.

1.6 Data Producer/Consumer Applications

1.6.1 Emergency Video Streaming

Vehicle-to-Vehicle live Video (V3) Streaming Architecture

V3 supports location-aware video streaming so that users can watch videos originating from remote regions of interest [20]. V3 assumes that vehicles are equipped with on-board computing, wireless communication devices, and GPS devices to keep track of their locations; and some of the vehicles have video cameras and have enough storage to buffer videos.

V3 is composed of a video triggering sub-system and a video transmission sub-system. The video triggering sub-system is responsible to forward video trigger message to the destination region. The video transmission sub-system sends video data back to the receiver. In Figure 1.5, a vehicle R that wants to receive streaming video from the region A, sends a triggering message to that region. A trigger message contains the requester ID, query time, destination region information, deadline, etc. The trigger message signals vehicles in a region of interests to start capturing videos and to send back the captured video to the requestor. In the figure, two vehicles observing the accidents stream the captured videos to the receiver using multi-hop routing.

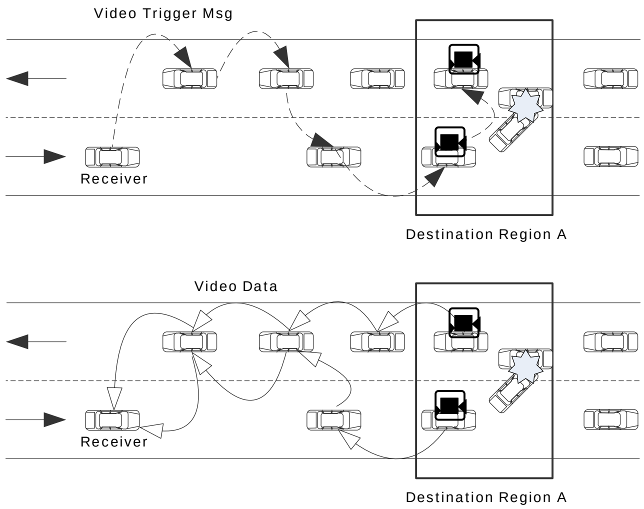


Figure 1.5: An Illustration of a typical V3 service process

One of the key challenges of implementing V3 is to overcome intermittent connectivity due to the dynamic nature of a VANET (i.e., high mobility, non-uniform vehicle distribution) and the low penetration ratio of V3-enabled vehicles. For triggering message delivery, V3 uses a *directional* flooding method based on a trigger message forwarding zone (TMFZone). For a given node v, TMFZone defines a set of potential forwarders that are closer to the destination region than node v or are moving towards the destination region. After a request is disseminated to nodes in the destination region, a video source vehicle must be selected. V3 selects a node that will possibly stay in the destination region the longest. Note that to handle the case that a source vehicle moves away from the region, V3 uses continuous trigger methods that forward the trigger message to the incoming vehicles to the region. For video transmission, V3 mainly uses a store-carry-and-forward approach to overcome intermittent connectivity. Like TMFZone, a node v selects a set of candidate forwarders that are closer to the requester or are moving towards the requester. Since it may not be efficient to send a packet to all of the candidate forwarders, the node v selects a subset of candidate forwarders, based on the amount of knowledge it has.

Reliable Video Streaming using Network Coding

Park et al. [43] proposed a *network coding* based emergency video streaming protocol. Unlike V3, it "pushes" urgent video streams regarding emergency situations such as natural disaster, traffic accidents, and terrorist attacks in order to help drivers effectively avert the danger, and random linear network coding is used to provide reliable and robust video streaming.

Suppose a multimedia data source generates a stream of frames $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\cdots$ where subscripts denote unique and consecutive sequence numbers (See Figure 1.6). A

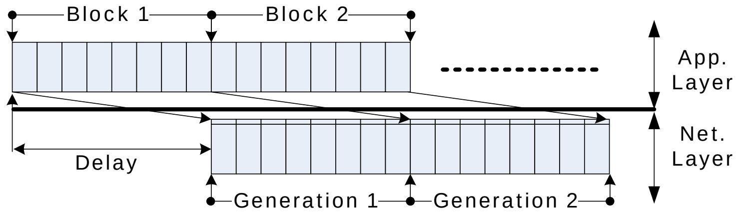


Figure 1.6: Multimedia streaming: each block has 8 frames

tuple ($blockid$, $blocksize$) is used to indicate a block of frames with sequence numbers greater than or equal to $blockid$ and smaller than ($blockid$ + $blocksize$) (i.e., $\mathbf{p}_{blockid}$, $\cdots$, $\mathbf{p}_{blockid+blocksize-1}$) belong to the block. A *coded packet* $\mathbf{c}_{(blockid,blocksize)}$ is a linear combination of frames in ($blockid$, $blocksize$). That is, $\mathbf{c}_{(blockid,blocksize)} = \sum_{k=1}^{blocksize} e_k \mathbf{p}_{(k-1+blockid)}$ where e_k is an element in a Galois field $\mathbb{F}$. Here, every arithmetic operation (i.e., addition and multiplication) is over the Galois field $\mathbb{F}$. Data frames $\mathbf{p}$'s and coded packets $\mathbf{c}$'s are also regarded as vectors over the field. In the header of a coded packet, $\mathbf{e} = [e_1 \ \cdots \ e_{blocksize}]$ is stored along with $blockid$ and $blocksize$ for the purpose of *decoding* packets on receivers. When generating a $\mathbf{c}$, each e_k is chosen randomly from $\mathbb{F}$, which is in general referred to as the random linear coding.

The reliable delivery service agent (or layer) residing on the multimedia data source generates and transmits code packets to the receivers. Since a block of frames is required to generate a coded packet, the agent residing on the video source collects frames generated by the application and buffers them. On reception of a coded packet $\mathbf{c}_{(blockid,\, blocksize)}$, every node stores the packet in its local memory for later decoding and forwarding. To recover $blocksize$ original frames belonging to ($blockid$, $blocksize$), a node should collect a $blocksize$ number of coded packets tagged with $(blockid,\ blocksize)$ and encoding vectors that are linearly independent of each other. Once collected, the reliable delivery service agent recovers the $blocksize$ original data frames and deliver them to the upper layer. Let $\mathbf{c}_k$ be a coded packet labeled $(blockid,\ blocksize)$ in a node's local memory, $\mathbf{e}_k$ be the encoding vector prefixed to $\mathbf{c}_k$, and $\mathbf{p}_{blockid+k-1}$ be an original data frame to be recovered where $k = 1, \cdots,$ $blocksize$. Further, let $\mathbf{E}^{\mathtt{T}} = [\mathbf{e}_1^{\mathtt{T}} \ \cdots \ \mathbf{e}_{blocksize}^{\mathtt{T}}]$, $\mathbf{C}^{\mathtt{T}} = [\mathbf{c}_1^{\mathtt{T}} \ \cdots \ \mathbf{c}_{blocksize}^{\mathtt{T}}]$, and $\mathbf{P}^{\mathtt{T}} = [\mathbf{p}_{blockid}^{\mathtt{T}} \cdots \ \mathbf{p}_{blockid+blocksize-1}^{\mathtt{T}}]$, then conceptually $\mathbf{P} = \mathbf{E}^{-1}\mathbf{C}$, which corresponds to the original data frames where superscripts $\mathtt{T}$ to denote the transpose operation. Note that all $\mathbf{e}_k$'s must be linearly independent to be able to invert $\mathbf{E}$.

When a node receives a coded packet with a new tuple ($blockid$, $blocksize$), it sets up a timer for the tuple ($blockid$, $blocksize$) expiring in $blocktimeout$ seconds. When the timer expires it broadcasts one coded packet $\acute{\mathbf{c}}_{(blockid,blocksize)}$ after local re-encoding to its neighbors. The local re-encoding is through the same process that the data source has undergone to generate a coded packet, i.e., a random linear combination of packets with the same ($blockid$, $blocksize$) available in local memory as shown in Figure 1.7. The timer for ($blockid$, $blocksize$) is reset on expiration unless a decodable set of packets is collected for the tuple ($blockid$, $blocksize$). On the expiration of

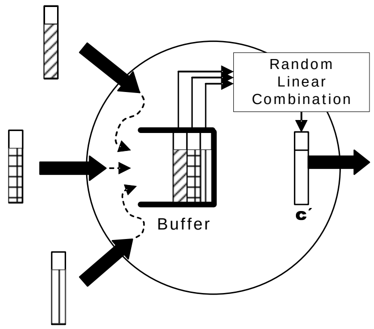


Figure 1.7: Re-encoding at an intermediate node

the timer for ($blockid$, $blocksize$), even though there are less than $blocksize$ number of packets of ($blockid$, $blocksize$) in the local memory, a node has to generate and transmit a coded packet using packets available in memory. The number of frames/packets that are combined to yield a coded packet is recorded in the field $rank$ in the header of the coded packet. Since a coded packet $\mathbf{c}_{(blockid,\, blocksize)}$ with $rank$ smaller than $blocksize$ indicates that the sender of **c** is in need of more coded packets tagged with ($blockid$, $blocksize$) to recover original frames, on reception of such packets, a node transmits another coded packet to help the sender of **c** collecting more coded packets.

Owing to this recovery process in combination of buffering of packets, the protocol can deliver packets efficiently and reliably across partitions. Suppose that a vehicle encounters a platoon of other vehicles carrying data that the vehicle does not have. The vehicle runs the recovery process and it collects data from the platoon. In the recovery process, a vehicle sends out a coded data packet tagged with ($blockid$, $blocksize$), and $rank$ where $rank < blocksize$ and in response to the *help request* packet, neighbors of the vehicles sends appropriate coded data packets. If a vehicle in seek of help has no data, then it just sends out header only packets with $rank$ = 0. In fact, the help request and responses handshaking is not necessarily to be done block by block (or generation by generation) if a vehicle wants to collect consecutive blocks of frames.

Tavarua: Video Streaming over 3G Services

Qureshi et al. [44] proposed Tavarua, a novel real-time multimedia communications sub-system designed to support mobile telemedicine applications that require high bandwidth (e.g., $>$500kbps) and QoS. As shown in Section 1.2.1, they evaluated the behavior of 3G services (i.e., 1xEvDO) in a vehicular network and found that 1xEvDO has relatively high latency (around 600ms), and low upload bandwidth (e.g., $<$140kbps). To support adequate bandwidth and QoS, Tavarua uses multiple simultaneous 3G connections. Tavarua has the following components: Tribe which provides the lowest level connection between Tavarua and the network interfaces; Horde which provides the network striping layer, including congestion control; and a video services subsystem. Prototype results show that Tavarua significantly mitigates the impact of packet loss on video quality and provides sufficient upstream bandwidth to transmit high quality video data.

1.6.2 FleaNet: A Virtual Market Place in VANETs

Wireless communications in vehicles will guide us into a new era of pervasive computing in which seamless access to information sources is provided. When traveling or shopping, for instance, one can search the *web* to get directions or locate specific products. In fact, not only do such devices empower us with ubiquitous Internet access but they also create a new environment of vehicular social networking where opportunistic cooperation can emerge among users with shared interests/goals, such as drivers exchanging safety related information, shoppers/sellers trading goods, etc. [46]

Following this model, Lee et al. [34] considered a "virtual flea market" in urban vehicular networks called FleaNet. FleaNet operates on the vehicular "ad hoc grid" without any infrastructure support and provides an excellent method for people to communicate with each other as buyers and sellers of goods (or information) and to efficiently find matches of interest, potentially leading to transactions. Figure 1.8 shows an illustration of FleaNet scenarios. Vehicles as well as static roadside Advertisement Stations (or Adstations) generate and propagate queries. Adstations can be stores advertising their products. For example, a pizzeria could advertise its special pizza offer to vehicles passing by and a driver who received the advertisement could place an order.

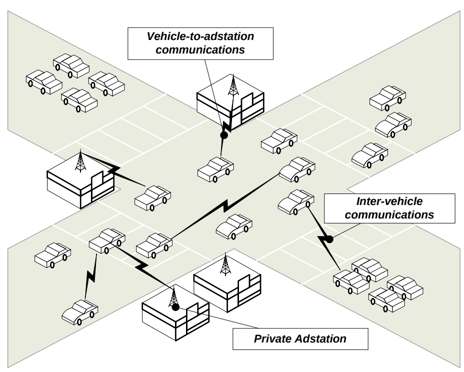
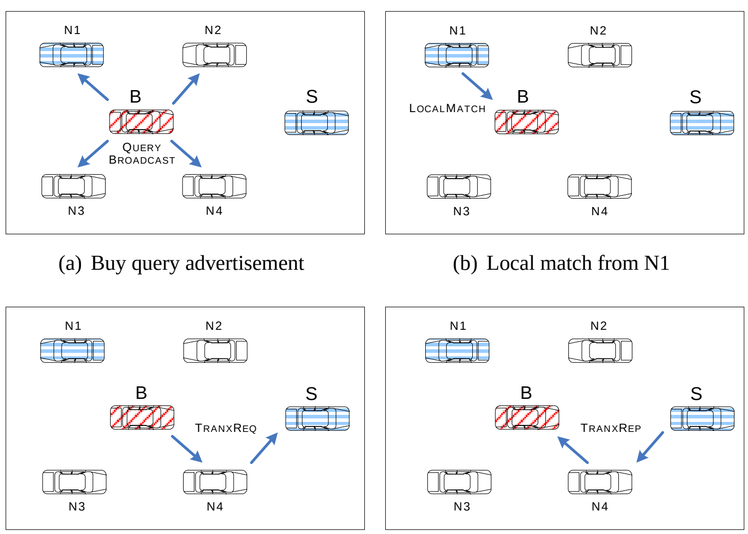


Figure 1.8: FleaNet Scenario

To illustrate FleaNet, consider the following examples. One day Joe Bruin wants to sell some of his items, but he is too busy with his work to do a garage sale. In this situation, FleaNet helps him to sell the items while he is behind the wheel (i.e., mobile garage sale!). He inputs details of the items using FleaNet software to create queries of items; for example, "Consumer Electronics/MP3 Players/Apple iPod Mini, 4G." Since he is commuting between downtown LA and west LA, he wants to find buyers near that area. Using a digital map provided by FleaNet software, he can easily set the area of interest to which his queries will be disseminated. For some items, he wants to see multiple matches, say 5, to make the best deal by simply setting the "number of matches" field. He also wants to sell the items while he is commuting, which takes about half an hour, and thus, he sets the expiration time accordingly. As a result, this query will be advertised and is spreading near his commuting path through vehicular networks using the FleaNet query dissemination protocol. Some time later, the query will be responded with a match message (i.e., a sell query of a ticket). Joe Bruin will then send a transaction request message to sell his item, and in the end, he will receive a transaction response from the originator of the matched query.

FleaNet utilizes mobility assisted query dissemination where the query "originator" periodically advertises his query only to one hop neighbors. Each neighbor then stores the advertisement (i.e., 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 match notification. Since this match could lead to an actual transaction, FleaNet provides a mechanism that routes the transaction request/reply. A user could see multiple matches for a given query. Based on his own criterion (either on distance from his current location or on the offered price), he selects the best one and sends the transaction request. Then, the target user responds with the transaction reply after seeing the request. For this purpose, FleaNet uses Last Encounter Routing (LER) [19]. LER is based on geo-routing and combines location service and routing service. In FleaNet the query packet includes the originator geo-coordinates, and thus, LER does not incur any initial flood search routing cost.



(a) Buy query advertisement

(b) Local match from N1

(c) Transaction request

(d) Transaction response

Figure 1.9: Match and transaction notification

Figure 1.9 shows an example of match notification. Let us assume that node B and node S are advertising sell and buy queries respectively, and $N1$ is carrying the sell query of node S since node $N1$ has already met S. In Figure 1.9(a), node B

advertises its buy query to its neighbors. Node $N1$ then finds a local match and sends LOCALMATCH to node B as shown in Figure 1.9(b). As a result, node B makes its final decision by sending TRANXREQ to node S and thus, S will respond with TRANXREQ to node B as shown in Figure 1.9(c) and Figure 1.9(d).

Readers can find an extensive evaluation of the FleaNet protocols via both analysis and simulation [34]. A random query can be resolved, in most cases, within a tolerable amount of time and with minimal bandwidth, storage and processing overhead; and if the advertiser is stationary (e.g., Adstation), the query resolution time is critically dependent on its location.

The use of FleaNet is not limited to vehicular ad hoc networks. It can be extended to other networks such as Personal Area Networks formed by pedestrians with PDAs or SmartPhones. Also, FleaNet can be associated with infrastructure. For instance, mobile users' advertisements can be uploaded in the Internet (e.g., Craigslist), and similarly, Internet users' advertisements can be posted in the vehicular networks. Besides, it is very important to provide incentives and security mechanisms to deal with non-cooperative or/and malicious users. Lee et al. [30] proposed Signature-Seeking Drive (SSD), a secure incentive framework for commercial ad dissemination in VANETs where a PKI (Public Key Infrastructure) is leveraged to provide secure incentives for cooperative nodes.

1.6.3 Vehicular Information Transfer Protocol (VITP)

VITP [7] aims 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). 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.

Figure 1.10 shows the illustration of VITP operations. Node V wants to know the average vehicle speed nearby the gas station, and it sends the query Q to the destination location (i.e., dispatch phase). VITP-enabled peers in that destination location cooperatively resolve the query (i.e., computation phase) and return a reply R to the requester (i.e., reply-delivery phase).

In VITP, locations are represented as a (roadID, segmentID) tuple where roadID is a unique key representing a road, and segmentID is a number representing a specific segment of the road. The VITP-enabled peers in a road segment become Virtual Ad-Hoc Servers (VAHS), and these peers cooperatively answer the query. This can be best explained using the above example; i.e., node V wants to know the average speed of the vehicles nearby the gas station. The query reaches to node $VS1$ in the destination zone (i.e., an oval area). After node $VS1$ performs a local computation of average speed and updates the query with its partial result, it sends the updated query $Q1$ to node $VS2$. Similarly, node $VS2$ performs the updates, and passes the updated query $Q2$ to $VS4$. In this way, on-the-fly updates continue until the query resolution process satisfies a return condition specified by the requester (i.e., the average value of at least three vehicles). VITP provides best-effort query resolution: any available VITP-enabled

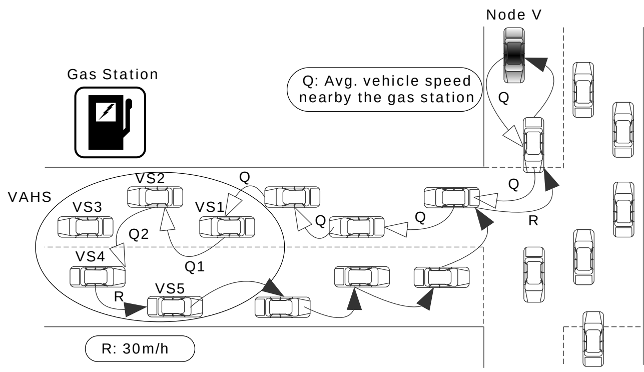


Figure 1.10: An Illustration of VITP

peers in a target road segment can participate in query resolution, and it is possible that peers can move out of the area before the query resolution completes.

1.6.4 Infrastructure-based P2P Traffic Information Systems

Rybicki et al. [48] designed a distributed Traffic Information System (TIS) that allows vehicles to exchange information about the current traffic status, say in a city. The major shortcoming of the VANET is its low penetration ratio which makes long distance communications hard (or opportunistic delay tolerant data exchange at best). Under current circumstances, they claimed that infrastructure-based mobile wireless access methods such as 3G, GPRS, WiMAX are more amenable. For instance, UMTS/HSDPA has been getting popular in many countries, and WiMAX has already deployed and used commercially.

To realize such a service, vehicles must be able to store and retrieve traffic information. Infrastructure allows using a “structured” overlay network, or Distributed Hash Table (DHT) such as Chord and CAN. Recall that DHTs provide a lookup service similar to a hash table such that (name, value) pairs, and any participating node can efficiently retrieve the value associated with a given name. However, there are still several challenges. Unlike Internet-based DHT where data tends to be static, sensed data on the roads are highly dynamic. Frequent handoffs could incur a short intermittent connectivity. Moreover, the system load is much heavier since it can be potentially used by tens of thousands vehicles at the same time. Thus, the system must be scalable/reliable and must well balance the system load. Yet, some of the issues could be efficiently handled using conventional techniques such as locality based load balancing and redundancy/coding based reliability solutions.

Given that this traffic information system shares the same concept of the publish/subscribe paradigm, Rybicki et al. [48] proposed to build a DHT-based publish/subscribe system for TIS. Vehicles publish traffic information to DHT and it can be delivered to the subscribers based on subscription information. For instance, a user who is interested in accessing traffic conditions during his commuting hours can automatically receive such information. However, there are still several "open" challenging issues. First, the road condition must be constantly monitored and reported in a timely manner. Second, the significant update of driving routes or other changes must be updated timely, so that users can receive information without any interruption. Third, the system must be able to reliably deliver information (e.g., having multiple distribution trees). Finally, the system must be able to preserve privacy, coordinate traffic patterns, and serve specialized requests (e.g., customized data delivery).

1.6.5 Interactive Applications

RoadSpeak: Voice Chatting in Vehicular Social Networks

Smaldone et al. [50] presented a framework for building virtual mobile communities they call Vehicular Social Networks (VSNs), to allow commuters to communicate with each other. RoadSpeak, their VSN-based system, allows drivers to form Voice Chat Groups (VCGs) and communicate on the road via voice chat messages. RoadSpeak uses a centralized server to manage users, and clients connect to the server using infrastructure that provides Internet connections (e.g., 3G services, WiFi access points, WiMAX, etc.).

Users need to first download the client and create an account in the system, which is required to log in and access groups. During the account creation process, the client will generate an asymmetric key pair (using PKI) and register its public key (K_{PUB}) with the server. At this point, they are free to join or create their own groups. To join a group, a user submits their profile, and if the Group Admission Manager validates and accepts the user, the group key (K_G) is transmitted back to the user.

Once on the road, the user chooses one group to listen to and connects to the voice chat server. The Connection Handler spawns an Admission Control Handler (ACH) to perform admission control. The client will then send the user's name, profile, and SHA1 hash of the group key to the server, which is encrypted using the client's private key. After validating the information, the ACH passes the connection to the Channel Handler for the VCG. The Channel Handler maintains one thread per client connected and handles delivery of messages to clients. The client itself keeps three queues: send, receive, and control queues. The first two buffer outgoing and incoming voice messages, respectively, and the third is used for flow control, allowing the server to send pause and resume messages.

To illustrate RoadSpeak, consider the following example. Joe Bruin downloads the RoadSpeak client onto his smart phone, and joins a politics discussion group that is active between 8 and 10 AM on the 10 Freeway between west LA and downtown LA, consistent with his daily commute. The next day, as Joe merges onto the 10, his GPS enabled smart phone detects his position and contacts the RoadSpeak server. Joe receives an alert notifying him that he has joined the group, and begins to listen to the on going discussions. When Joe speaks, his RoadSpeak client transmits his message to the server, which is distributed to all other members of the group. As Joe nears the end of his commute, his smart phone detects his position and the client notifies him of his imminent departure from the group.

Online "Passenger" Games

Interactive online games have been mainly based on the Internet infrastructure. As the wireless networking technology becomes ubiquitous in mobile platforms (e.g., Smart Phones and vehicles), wireless and mobile online gaming will soon emerge. In particular, vehicles will provide an ideal platform for wireless and mobile gaming. Vehicular entertainment systems are ever gaining popularity – more than millions of TV/DVD systems are sold every year. The convergence of VANETs and entertainment will bring online passenger gaming to reality. Given this, say, a family or a group of friends driving multiple vehicles in a caravan to a distant place can spend their time engaging in online game sessions with each other through wireless P2P communications. Internet connectivity via access points or 3G services can also support this application, yet these alternatives are less attractive given that wireless access points cannot guarantee quality of service due to intermittent connectivity, and 3G services require a subscription fee.

One of the key challenges is to broadcast game events with high bandwidth and low latency to all the players, because high interactivity is a fundamental feature for online games [42]. For instance, the tolerable delay of fast paced games is about 150-200ms for ordinary players and 50-100ms for professional players respectively. Moreover, these events must be delivered via the shared wireless medium. Fairness among online players is another key issue.

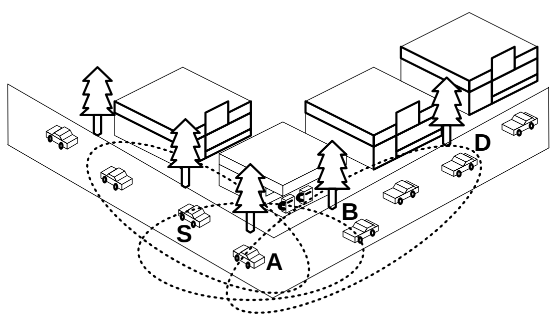


Figure 1.11: An Illustration of FMBP

To this end, Palazzi et al. [42] designed an efficient multi-hop broadcast scheme called Fast Multi-Broadcast Protocol (FMBP). Most broadcasting protocols use a random jitter (or back-off) to prevent collision. Also, those protocols force the nodes that are located furthest to forward packets in order to minimize the hop count and thus, the average delay. This can be implemented by setting one's back-off window inversely proportional to the distance from the originator: i.e., $\tau(1 - d/R)$ where d is the distance from the sender, R is the maximum transmission range, and τ is a fixed time slot value. For instance, if a node's distance is R, it will immediately send out a packet as soon as one receives a packet. In reality, however, the transmission range is not fixed due to road layouts and obstacles in vehicular networks. In Figure 1.11, vehicle A has a shorter communication range due to road layouts and obstacles. Thus, in FMBP nodes estimates its maximum distance in both forward and backward directions. In the example, vehicle B set the backward distance as $|P_B - P_A|$ and the forward distance as $|P_D - P_B|$. A node selects R value based on the direction of the broadcast message. The simulation results in [42] show that FMBP can effectively reduce the delay compared to the other schemes.

1.6.6 Other Applications and Related Works

TrafficView [37] disseminates, or pushes (through flooding), information about the vehicles on the road, thus providing real-time road traffic information such as speed of vehicles to drivers. To alleviate broadcast storms, this work focuses on data aggregation/fusion based on distance from the source. Gradinescu et al. [18] proposed adaptive traffic lights where a wireless controller installed in an intersection determines the optimum values for the traffic light phases. The wireless controller collects the volume of vehicles using TrafficView. Zhou et al. [66] proposed EZCab that discovers and books free cabs using V2V communications. A client (e.g., a node located at a taxi stand) first sends a BookSM packet to find free cabs using flooding or probabilistic forwarding, and free cabs send back ReportSM packets to the client. Given this, the client chooses a cab and makes a reservation by sending a ConfirmSM packet to the cab. Similarly, Huang et al. [23] investigated financial and technical feasibility of a taxi dispatching application.

1.7 Conclusion

In this chapter we surveyed the emerging vehicular applications, ranging from vehicular sensors to entertainment. We outlined a potential vehicular network architecture based on various wireless protocols and access methods such as DSRC, 3G, and WiMAX. We described these protocols and discussed the unique characteristics of vehicular communications. Given this, we proceeded to classify a representative set of VANET P2P applications based on the vehicle's role in managing data: as source, consumer, source/consumer, or intermediary. For the data source scenario, we reported the state-of-the-art vehicular sensing applications; for the consumer scenario, we summarized the contention distribution applications such as multi-media files or advertisements; and for the source/consumer scenarios, we reviewed (emergency) P2P video streaming, virtual flea markets, vehicular information services, and interactive applications (e.g., chatting/games).

Bibliography

[1] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[2] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory Sensing. In *ACM WSW*, Boulder, CO, USA, Oct.-Nov. 2006.

[3] M. Caesar, M. Castro1, E. B. Nightingale, G. O'Shea, and A. Rowstron. Virtual Ring Routing: Network routing inspired by DHTs. In *SIGCOMM'06*, Pisa, Italy, Sept. 2006.

[4] M. Caliskan, D. Graupner, and M. Mauve. Decentralized Discovery of Free Parking Places . In *ACM VANET*, Los Angeles, CA, USA, Sept. 2006.

[5] Z. D. Chen, H. Kung, and D. Vlah. Ad Hoc Relay Wireless Networks over Moving Vehicles on Highways. In *ACM MOBIHOC*, Long Beach, CA, USA, Oct. 2001.

[6] D. M. Chiu, R. W. Yeung, J. Huang, and B. Fan. Can Network Coding Help in P2P Networks? In *NetCod'06*, Boston, MA, Apr. 2006.

[7] M. D. Dikaiakos, S. Iqbal, T. Nadeem, and L. Iftode. VITP: An Information Transfer Protocol for Vehicular Computing . In *ACM VANET*, Cologne, Germany, Sept. 2005.

[8] Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Sept. 2003.

[9] S. B. Eisenman, G.-S. Ahn, N. D. Lane, E. Miluzzo, R. A. Peterson, and A. T. Campbell. MetroSense Project: People-Centric Sensing at Scale. In *ACM WSW*, Boulder, CO, USA, Oct.-Nov. 2006.

[10] T. ElBatt, S. K. Goel, G. Holland, H. Krishnan, and ayendra Parikh. Cooperative Collision Warning Using Dedicated Short Range Wireless Communications . In *ACM VANET'06*, Los Angeles, CA, USA, Sept. 2006.

[11] J. Eriksson, H. Balakrishnan, and S. Madden. Cabernet: A Content Delivery Network for Moving Vehicles. Technical Report TR-2008-003, MIT-CSAIL, 2008.

[12] J. Eriksson, L. Girod, B. Hull, R. Newton, H. Balakrishnan, and S. Madden. The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring . In *MobiSys'08*, Breckenridge, Colorado, June 2008.

[13] L. Fan, P. Cao, and J. Almeida. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. In *ACM SIGCOMM*, Vancouver, Canada, Aug.-Sept. 1998.

[14] R. Frenkiel, B. Badrinath, J. Borras, and R. D. Yates. The Infostations Challenge: Balancing Cost and Ubiquity in Delivering Wireless Data. *IEEE Personal Communications*, 7-2:66–71, Apr. 2002.

[15] M. Gerla, B. Zhou, Y.-Z. Lee, F. Soldo, U. Lee, and G. Marfia. Vehicular Grid Communications: The Role of the Internet Infrastructure. In *WICON'06*, Boston, NY, Aug. 2006.

[16] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan. IrisNet: An Architecture for a Worldwide Sensor Web. *IEEE Pervasive Computing*, 2(4):22–33, Oct.-Dec. 2003.

[17] C. Gkantsidis and P. Rodriguez. Network Coding for Large Scale Content Distribution. In *INFOCOM'05*, Miami, FL, USA, Mar. 2005.

[18] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode. Adaptive Traffic Lights Using Car-to-Car Communication. In *IEEE Vehicular Technology Conference*, Dublin, Ireland, Apr. 2007.

[19] M. Grossglauser and M. Vetterli. Locating Nodes with EASE: Mobility Diffusion of Last Encounters in Ad Hoc Networks. In *IEEE INFOCOM*, San Francisco, CA, USA, Mar.-Apr. 2003.

[20] M. Guo, M. H. Ammar, and E. W. Zegura. V3: A Vehicle-to-Vehicle Live Video Streaming Architecture. In *PerCom'05*, Mar. 2005.

[21] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal. Vehicular Opportunistic Communication Under the Microscope. In *MobiSys'07*, San Juan, Puero Rico, June 2007.

[22] M. Han, S. Moon, Y. Lee, K. Jang, and D. Lee. Evaluation of VoIP Quality over WiBro. In *Passive and Active Measurement Conference (PAM)*, Cleveland, OH, Apr. 2008.

[23] E. Huang, W. Hu, J. Crowcroft, and I. Wassell. Towards Commercial Mobile Ad Hoc Network Applications: A Radio Dispatch System. In *MobiHoc'05*, Urbana-Champaign, IL, USA, May 2005.

[24] B. Hull, V. Bychkovsky, K. Chen, M. Goraczko, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *ACM SenSys*, Boulder, CO, USA, Oct.-Nov. 2006.

[25] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: a Scalable and Robust Communication Paradigm for Sensor Networks. In *ACM MOBICOM'00*, Boston, MA, USA, 2000.

[26] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich. Design of 5.9GHz DSRC-based Vehicular Safety Communication. *IEEE Wireless Communications*, 13(5):36–43, Oct. 2006.

[27] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In *ACM ASPLOS-X*, San Jose, CA, USA, Oct. 2002.

[28] G. Korkmaz, E. Ekici, F. Ozguner, and U. Ozguner. Urban Multi-Hop Broadcast Protocols for Inter-Vehicle Communication Systems. In *ACM VANET*, Philadelphia, PA, USA, Oct. 2004.

[29] K. C. Lee, S.-H. Lee, R. Cheung, U. Lee, and M. Gerla. First Experience with CarTorrent in a Real Vehicular Ad Hoc Network Testbed. In *MOVE'07*, Anchorage, Alaska, May 2007.

[30] S.-B. Lee, G. Pan, J.-S. Park, M. Gerla, and S. Lu. Secure Incentives for Commercial Ad Dissemination in Vehicular Networks. In *MobiHoc'07*, Quebec, Canada, Sept. 2007.

[31] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, P. Lio, and K.-W. Lee. Bio-inspired Multi-Agent Data Harvesting in a Proactive Urban Monitoring Environmen. *Elsevier Ad Hoc Networks Journal, Special Issue on Bio-Inspired Computing and Communication in Wireless Ad Hoc and Sensor Networks*, 2008.

[32] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi. MobEyes: Smart Mobs for Urban Monitoring with Vehicular Sensor Networks. *IEEE Wireless Communications*, 13(5):51–57, Sept.-Oct. 2006.

[33] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi. Dissemination and Harvesting of Urban Data using Vehicular Sensor Platforms. *IEEE Transaction on Vehicular Technology*, 2008.

[34] U. Lee, J.-S. Park, E. Amir, and M. Gerla. FleaNet: A Virtual Market Place on Vehicular Networks. In *V2VCOM'06*, San Jose, CA, July 2006.

[35] U. Lee, J.-S. Park, J. Yeh, G. Pau, and M. Gerla. CodeTorrent: Content Distribution using Network Coding in VANETs. In *MobiShare'06*, Los Angeles, CA, Sep. 2006.

[36] C. Maihöfer, T. Leinmüller, and E. Schoch. Abiding Geocast: Time-Stable Geocast for Ad Hoc Networks. In *ACM VANET'05*, Cologne, Germany, Sept. 2005.

[37] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode. TrafficView: Traffic Data Dissemination using Car-to-Car Communication. *ACM Mobile Computing and Communications Review (MC2R)*, 8(3):6–19, July 2003.

[38] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Y. Sanadidi. Co-operative Downloading in Vehicular Ad-Hoc Wireless Networks. In *IEEE/IFIP WONS*, St. Moritz, Swiss, Jan. 2005.

[39] A. Nandan, S. Tewari, S. Das, G. Pau, M. Gerla, and L. Kleinrock. AdTorrent: Delivering Location Cognizant Advertisements to Car Networks. In *IEEE/IFIP WONS*, Les Menuires, France, Jan. 2006.

[40] S. Nath, J. Liu, and F. Zhao. Challenges in Building a Portal for Sensors World-Wide. In *ACM WSW*, Boulder, CO, USA, Oct.-Nov. 2006.

[41] V. Naumov and T. Gross. Connectivity-Aware Routing (CAR) in Vehicular Ad Hoc Networks. In *INFOCOM'07*, Anchorage, Alaska, May 2007.

[42] C. E. Palazzi, M. Roccetti, G. Pau, and M. Gerla. Online Games on Wheels: Fast Game Event Delivery in Vehicular Ad-hoc Networks. In *V2VCOM'07*, Istanbul, Turkey, June 2007.

[43] J.-S. Park, U. Lee, S. Y. Oh, M. Gerla, and D. Lun. Emergency Related Video Streaming in VANETs using Network Coding. In *ACM VANET'06*, Los Angeles, CA, USA, Sept. 2006.

[44] A. Qureshi, J. Carlisle, and J. Guttag. Tavarua: Video Streaming with WWAN Striping. In *ACM Multimedia 2006*, Santa Barbara, CA, Oct. 2006.

[45] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A Geographic Hash Table for Data-Centric Storage. In *WSNA'02*, Atlanta, GA, USA, Sept. 2002.

[46] H. Rheingold. *Smart Mobs: The Next Social Revolution*. Basic Books, 2003.

[47] O. Riva and C. Borcea. The Urbanet Revolution: Sensor Power to the People! *IEEE Pervasive Computing*, 6(2), 2007.

[48] J. Rybicki, B. Scheuermann, W. Kiess, C. Lochert, P. Fallahi, and M. Mauve. Challenge: Peers on Wheels - A Road to New Traffic Information Systems. In *MobiCom'07*, Quebec, Canada, Sept. 2007.

[49] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks. *Elsevier Ad Hoc Networks Journal*, 1(2-3):215–233, Sept. 2003.

[50] S. Smaldone, L. Han, P. Shankar, and L. Iftode. RoadSpeak: Enabling Voice Chat on Roadways using Vehicular Social Networks. In *SocialNets'08*, Glasgow, Scotland, UK, Apr. 2008.

[51] T. Small and 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 *ACM MOBIHOC*, Annapolis, Maryland, USA, June 2003.

[52] D. Sormani, G. Turconi, P. Costa, D. Frey, M. Migliavacca, and L. Mottola. Towards Lightweight Information Dissemination in Inter-vehicular Networks . In *ACM VANET'06*, Los Angeles, CA, USA, Sept. 2006.

[53] M. T. Sun, L. Huang, A. Arora, and T. H. Lai. Reliable MAC Layer Multicast in IEEE Wireless Networks. In *ICCP'02*, Vancouver, Aug. 2002.

[54] W. L. Tan and O. Yue. Measurement-based Performance Model of IP Traffic over 3G Networks. In *TENCON 2005 IEEE Region 10*, Melbourne, Australia, Nov. 2005.

[55] K. Tang and M. Gerla. MAC Reliable Broadcast in Ad Hoc Networks. In *IEEE MILCOM'01*, Washington, D.C., Oct. 2001.

[56] M. Torrent-Moreno, D. Jiang, and H. Hartenstein. Broadcast Reception Rates and Effects of Priority Access in 802.11-based Vehicular Ad-hoc Networks. In *ACM VANET*, Philadelphia, PA, USA, Oct. 2004.

[57] A. Vahdat and D. Becker. Epidemic Routing for Partially-Connected Ad Hoc Networks. Technical Report CS-200006, Duke University, Apr. 2000.

[58] Y. Wang and H. Wu. DFT-MSN: The Delay/Fault-Tolerant Mobile Sensor Network for Pervasive Information Gathering. In *INFOCOM'06*, Barcelona, Spain, Apr. 2006.

[59] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. MDDV: a Mobility-centric Data Dissemination Algorithm for Vehicular Networks. In *ACM VANET*, Philadelphia, PA, USA, Oct. 2004.

[60] Q. Xu, T. Mak, J. Ko, and R. Sengupta. Vehicle-to-Vehicle Safety Messaging in DSRC. In *ACM VANET*, Philadelphia, PA, USA, Oct. 2004.

[61] R. M. Yadumurthy, A. C. H., M. Sadashivaiah, and R. Makanaboyina. Reliable MAC Broadcast Protocol in Directional and Omni-directional Transmissions for Vehicular Ad Hoc Networks. In *VANET'05*, Sept. 2005.

[62] J. Yin, T. ElBatt, G. Yeung, B. Ryu, and S. Habermas. Performance evaluation of safety applications over dsrc vehicular ad hoc networks. In *ACM VANET'04*, Philadelphia, PA, USA, October 2004.

[63] H. Yoon, J. Kim, F. Tan, and R. Hsieh. On-demand Video Streaming in Mobile Opportunistic Networks. In *PerCom'08*, Hong Kong, China,, Mar. 2008.

[64] J. Yoon, B. Noble, and M. Liu. Surface Street Traffic Estimation. In *MobiSys'07*, San Juan, Puero Rico, June 2007.

[65] J. Zhao and G. Cao. VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks. In *IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[66] P. Zhou, T. Nadeem, P. Kang, C. Borcea, and L. Iftod. EZCab: A Cab Booking Application Using Short-Range Wireless Communication. In *IEEE PerCom'05*, Kauai Island, Hawaii, USA, Mar. 2005.