Computer Communications 34 (2011) 1159–1168

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A new paradigm for urban surveillance with vehicular sensor networks

Xu Li [a,*], Hongyu Huang [b], Xuegang Yu [c], Wei Shu [e], Minglu Li [d], Min-You Wu [d]

[a] Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, USA
[b] High Performance Networking Research Center, Chongqing University, Chongqing, China
[c] College of Computer Science and Technology, Jilin University, China
[d] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[e] Department of Electrical and Computer Engineering, The University of New Mexico, NM, USA

ARTICLE INFO

Article history:
Received 15 December 2009
Received in revised form 2 November 2010
Accepted 26 November 2010
Available online 3 December 2010

Keywords:
Vehicular sensor networks
Cooperative storage
Mobile surveillance
Intelligent transportation system

ABSTRACT

We consider a new application paradigm of vehicular sensor networks (VSN). Currently, vehicles are equipped with forward facing cameras to assist forensic investigations of events by proactive image-capturing from streets and roads. Due to content redundancy and storage imbalance in this in-network distributed storage system, how to maximize its storage capacity becomes a nontrivial challenge. In other words, how to maximize the average lifetime of sensory data (i.e., images generated by cameras) in the network is a fundamental problem to be solved. This paper presents, VStore, a cooperative storage solution in vehicular sensor networks for mobile surveillance, which has been designed to support redundancy elimination and storage balancing throughout the network. Compared with existing works, we propose a novel storage architecture for urban surveillance and deal with challenges in a mobile scenario. Field testing was carried out with a trace-driven simulator, which utilized about 500 taxis in Shanghai. The testing results showed that VStore can largely prolong the average lifetime of sensory data by cooperative storage.

1. Introduction

Conventional infrastructures have involved using static sensors for surveillance in city urban areas, which are limited in spatial coverage. Recently, a new infrastructure for mobile surveillance has been proposed, named *vehicular sensor networks* (*VSN*) [10], which is a network of mobile sensors equipped on vehicles, such as taxis and buses. VSNs facilitate collection of surveillance data over a wider area than fixed infrastructure. At the same time, unlike traditional sensor nodes, vehicular sensors are typically not affected by strict energy constraints and vehicles can be equipped with powerful processing units and wireless transmitters.

Currently, with inclusion of forward facing cameras mounted on vehicles, there is an increasing interest in proactive urban sensing, e.g., vehicles can continuously capture images from streets and maintain sensory data (i.e., images generated by cameras) in their local storage. It may assist the scene reconstruction of crimes, and more generally, the forensic investigations of events monitored by VSN, such as traffic accidents [12], terrorist attacks, as shown in Fig. 1. Actually, this architecture can also be regarded as an in-network distributed storage system [7]. Different from conventional wireless sensor networks, it is infeasible to deliver image data to sink because it may incur considerable communication costs. In particular, such costs are not necessary because no one can predict which data will be useful for future investigations. In this system, fresh image data will replace the oldest data by FIFO strategy when a vehicle has no free storage space. Due to content redundancy and storage imbalance, a primary concern is how to maximize storage capacity of the network. In other words, the fundamental problem is to maximize the average lifetime of sensory data for possible investigations when we have no prior knowledge about when/whether a given piece of information will be useful in the future. In addition, it is true that non-volatile storage prices will no doubt reduce, but we still need to optimize storage capacity of this mobile surveillance system [11].

In this paper, we present VStore, a cooperative storage system in vehicular sensor networks for urban mobile surveillance. VStore is designed to maximize the average lifetime of image data facing data redundancy and storage imbalance. The main challenges of VStore are enumerated as follows. First, repeated sensing by multiple vehicles incurs data redundancy. At the same time, high mobility and dynamic topology make sensing task assignment/negotiation difficult. Second, forward facing cameras are assumed to have fixed sampling rate in most of previous works, which inevitably implies ineffective storage utilization. For example, a vehicle may reduce sampling rate to avoid unnecessary sensing when it stops before red light or has a low speed in traffic jam. We need

---

* Corresponding author.
E-mail address: colorfuldays_lee@hotmail.com (X. Li).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.11.008

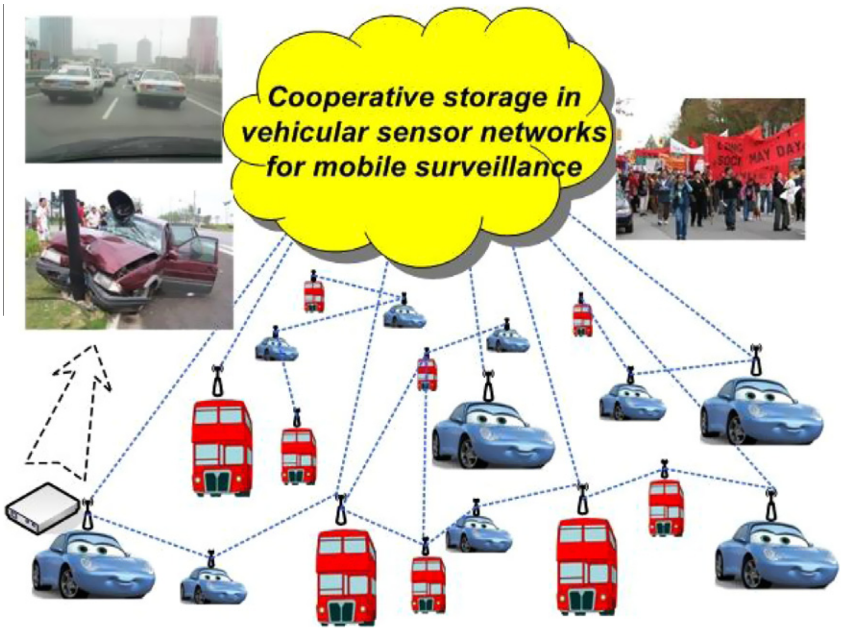


Fig. 1. Vehicular sensor networks for mobile surveillance in urban areas.

to design an adaptive sampling strategy in VStore based on vehicle status, such as speed, direction, etc. Finally, discrepancy of driving behaviors accounts for the fact that some vehicles have already overflowed storages while others still have free storage. Hence, VStore should balance data storage to make most use of network storage capacity from a global standpoint. Overall, compared with previous works, we propose a novel storage architecture for urban mobile surveillance and deal with technical issues in a mobile scenario.

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we present VStore scheme in details. The overview of data processing is presented in Section 4, followed by the performance evaluation part in Section 5. Section 6 concludes the paper.

2. Related work

Recent years, many techniques have been proposed for sensory data dissemination [1,19], such as data-centric storage (DCS), with which related detections are stored at predefined locations. Geared for application of mobile surveillance in our work, VStore has a completely different focus. That is, the sensory image data is always stored in local storage because whether it will be useful is unknown in advance. VStore mainly focuses on how to maximize storage capacity by cooperative storage. Thus, traditional DCS incurs unnecessary communication cost by migrating data from source to storing node.

Several other storage services have also been paid much attention in wireless sensor networks domain. Work in [2,21] introduced a two-tier data-centric storage and retrieval service using distributed hash table and double-ruling. PRESTO [3] and TSAR [4] proposed a two-tier data storage architecture comprising sensor nodes and proxies for data acquisition and query processing. DIMENSION [5] is designed to store long-term information by constructing summaries at different spatial resolutions using various compression techniques. TinyDB [6] organized sensor networks and their collected data as a distributed database and focused on query processing techniques to acquire data from such databases. EnviroStore [7] is a new cooperative storage system for sensor networks geared for disconnected operation. The goal of this system is to maximize data storage capacity by distributing storage utilization and opportunistically offloading data to external devices. EnviroMic [8] is a novel distributed acoustic monitoring, storage, and trace retrieval system. It is also designed for disconnected operation, where the existence of base station cannot be assumed. Different from these previous works, our work plans to study how to achieve cooperative storage in a mobile scenario, which has been untapped in most of previous works.

In addition, a number of works focused on vehicular sensor networks [9,14,15,18,20,23,24]. A survey of wireless multimedia sensor networks can be found in [9]. Lee et al. proposed MobEyes, which is an effective middleware specifically designed for proactive urban monitoring that exploits node mobility to opportunistically diffuse sensory data summaries among neighbor vehicles and to create a low-cost index to query monitoring data [10,16]. Compared with MobEyes, which is dedicated to query processing issue, VStore aims to maximize the averagelifetime of sensory data. Greenhill and Venkatesh proposed a mobile surveillance system with cameras on buses [11,12]. They first introduced a distributed query processing method by utilizing GPS track as index to upload on-demand data to sink [11]. In another work, they paid more attention on multimedia information processing of video stream data [12].

3. Cooperative storage in VStore

3.1. Basic model

In this paper, a vehicular sensor network for mobile surveillance is modeled as a set of n mobile vehicular nodes, denoted as a set $V = \{v_1, v_2, \ldots, v_n\}$ and every node has an individual id v_id (vehicle/node will be used interchangeably). Meanwhile, it is assumed that every node was equipped with GPS, digital map, 54 Mbps 802.11 g wireless transmitter, one forward facing camera and storage device. Similar assumptions can be found in related work [10–12]. The wireless transmitter has a communication range cr such that two nodes u and v can communicate directly if $|u - v| \leqslant cr$. Here $|u - v|$ is Euclidean distance between u and v. Moreover, the available bandwidth and storage on each node are limited but energy will be infinite resource provided by vehicle. Node-meeting is assumed to be short-lived because of high mobility of node.

The road network of Shanghai is of large-scale with thousands of links and intersections. A *link* with a link id (denoted as *l_id*) is a road section between two intersections (called *point*). For a particular link, it always associates with two intersection points, named *fpoint* (from-point) and *tpoint* (to-point). Note that, for a particular intersection point, it could be the fpoint of a link and the tpoint of another link. A *road* consists of several ordered links and all of them share the same road name [17].

3.2. Problem definition

In this section we introduce some definitions in VStore.

Definition 1 (*Maximum photographic distance*). With forward facing camera, a node could proactively capture images from links. The camera, however, has a limited field of vision, i.e., it only has capability to generate a high quality image for limited distance in front of the vehicle. This parameter is defined as *maximum photographic distance*, denoted as *mpd*, as shown in Fig. 2. How to regulate camera orientation for image capturing and how to deal with multimedia information processing are beyond the scope this paper, related works can be found in [11,12]. In addition, here we mainly focus on mobile surveillance for streets and we assume that traditional fixed cameras could be used to monitor intersection areas.

For a particular *mpd*, a node will capture an image and generate an image file at each recording point. As shown in Fig. 2, if a node enters into link l_i from fpoint, it will capture images at points *a*, *b*, *c*, *d* and *e*. Conversely, if it enters from tpoint, the images will be captured at points *f*, *e*, *d*, *c* and *b* in sequence. For every segment between two recording points, we assign a segment id in increasing order from fpoint to tpoint, denoted as *s_id* (e.g., *s_id* from 1–5 in Fig. 2). Normally, the length of every segment is *mpd* meters, except for the last segment of link because link length is not always integral times of *mpd* (e.g., segment 5 in Fig. 2). With GPS and digital map, it is trivial for vehicles to calculate the locations of these recording points.

Definition 2 (*Attributes of image file*). In general, we define some attributes for an image:

$$Attrib(l\_id, s\_id, timestamp, ti, v\_id),$$

where (*l_id*, *s_id*) indicates location information where an image was generated, *timestamp* and *v_id* show when and which vehicle generated this image. And *ti* is a counter and defined as:

$$ti = \lceil t/\tau \rceil, \tag{1}$$

where *t* is the time duration from time 0 (i.e., the start time of simulation) to current time. τ is a predefined configurable parameter, named *valid time*. For a given segment, two images are regarded as same if the difference of their *timestamps* is smaller than τ, i.e., two images im_i and im_j are regarded as same images if and only if:

$$Attrib_i.l\_id = Attrib_j.l\_id \wedge Attrib_i.s\_id = Attrib_j.s\_id \wedge Attrib_i.ti = Attrib_j.ti \tag{2}$$

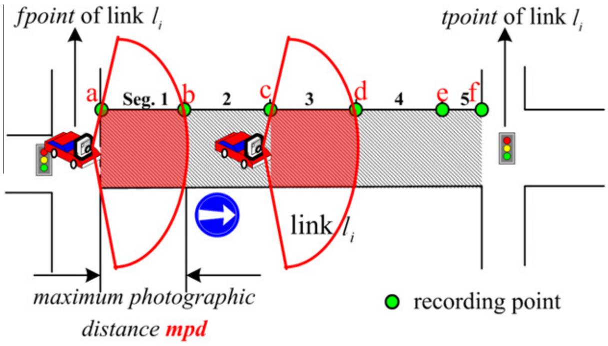


Fig. 2. The illustration of maximum photographic distance (*mpd*).

In addition, we can regulate τ with different values to support various surveillance purposes. As an example, it is reasonable to set τ with 10 min for traffic monitoring because traffic flow does not have a considerable change in a short time period.

Definition 3 (*Metadata of node*). For every node, it holds a metadata table for all nodes in the network, one item in metadata table is represented as follows:

$$Metadata(v\_id, rs, updatedTime),$$

It tells us that the free storage of node *v_id* is *rs* and this item was updated at time *updatedTime*. When two nodes meet, they first exchange their metadata tables and update items with latest *updatedTime*.

Definition 4 (*Redundancy ratio*). For a given VSN with *n* nodes, we define redundancy ratio *rr* of network at time *t* as:

$$rr(t) = \frac{\sum_{i=1}^{n}|S_i(t)| - |\cup_{i=1}^{n} O_i(t)|}{\sum_{i=1}^{n}|S_i(t)|}, \tag{3}$$

where $S_i(t)$ is the multi-set of image files generated by node v_i during $[0, t]$ and $|S_i(t)|$ is the size of this multi-set. $O_i(t)$ is the set of different image files generated by node v_i during $[0, t]$ and $|O_i(t)|$ is the size of this set. Note that, a multi-set can contain same elements whereas a set only includes individual elements (refer to Eq. (2)). This metric indicates redundancy degree of sensory data in the network, the higher *rr (t)* is, the more storage space has been wasted.

Definition 5 (*Average lifetime*). For images, the average lifetime in the network at time *t*, denoted as $\Omega(t)$, is defined as:

$$\Omega(t) = \left(\sum_{i=1}^{n} f(i)\right) \Big/ n \tag{4}$$

where *f (i)* is a function which calculates the earliest *ti* of image files carried by node v_i. Average lifetime $\Omega(t)$ indicates the average time duration that an image can stay in the network. VStore aims to maximize $\Omega(t)$.

As shown above, several definitions have been introduced in order to describe mobile surveillance application into an optimization problem. Specifically, maximum photographic distance models the capacity of vehicular camera. Attribute of image file and metadata of node facilitate us to eliminate data redundancy. Redundancy ratio and average life time are two metrics for evaluating performances of cooperative storage schemes.

3.3. Cooperative storage in VStore

In this section, we present VStore, a cooperative storage system in vehicular sensor networks for mobile surveillance. Two submechanisms are introduced in this section.

(1) *Cooperative recording and redundancy elimination*
The objective of cooperative recording is to capture images from links and keep unique image file in the network

according to Eq. (2). Here, an inherent assumption is that a link segment can be recorded by more than one vehicle during time interval *ti*(repeated sensing due to large vehicle density in downtown area). First, we review existing works in static wireless sensor networks, in which energy-efficiency is a critical issue. In WSN, when multiple nodes sensed an event simultaneously, they always form a group. A leader will be elected between these members, who will finally assign the task to one of members [8]. In our work, however, it is infeasible to employ this negotiation process because of high mobility of nodes and dynamic topology. Meanwhile, it is impossible to allocate tasks in advance because we cannot predict vehicle's paths. In this mobile scenario, we propose a new approach for cooperative recording. That is, every node first captures images from links, then whether it should delete some redundant images is based on information from others. To be more precise, when a node generates one image, it also creates a tag for this file including attributes introduced in Definition 2. When two nodes meet, they will exchange image tags and metadata of nodes (see Definition 3) in terms of metadata packet and retain a copy for new tags. After that, nodes will execute redundancy elimination using the following principle: for an image file im_i and a tag Tg_j carried by node v, v will delete im_i if and only if:

$$\begin{aligned} Attrib_i.l\_id = Tg_j.l\_id \wedge Attrib_i.s\_id = Tg_j.s\_id \wedge Attrib_i.ti \\ = Tg_j ti \wedge Attrib_i.timestamp < Tg_j.timestamp \end{aligned} \quad (5)$$

To avoid tag storm problem, every tag has its own TTL (Time-To-Live) so that tags will be eliminated from the network when they runs out of its TTL, it, as well as all their copies.

(2) *Distributed storage balancing*

In this mobile surveillance application, several reasons account for storage imbalance. For example, the sampling scheme in VStore can largely reduce excessive and unnecessary sensing, compared with previous works [10–12], in which sensors have fixed sampling rate. In fact, the forward facing camera can adapt its sampling rate based on vehicle status. The adaptive sampling rate ¢ can be defined as:

$$\text{¢} = 1/(mpd/spd) = spd/mpd, \quad (6)$$

where *spd* is the current speed of vehicle. Thus, *spd* will affect sampling rate and it also leads to storage imbalance because of various vehicle status and behaviors.

VStore needs to migrate data from heavy-load nodes to the light-load nodes in order to improve overall storage capacity. In traditional wireless sensor networks [7], researchers adopted lazy-offload scheme to save energy by postponing data balancing until the latest possible time. This is not applicable in proactive sensing scheme because images are generated continuously. Infrequent storage balancing implies potential network congestion because of massive data volume. At the same time, in traditional wireless sensor networks, node first decides whether it needs to offload and then starts to choose an appropriate neighbor. This approach is also not feasible in mobile scenario because high mobility of vehicles makes neighbor selection useless. Even if an appropriate neighbor can be identified, the limited connection/contact time and bandwidth still cannot guarantee a satisfying data transferring. Based on this discussion, we propose that nodes should offload data as soon as they need. In VStore scheme, v decides to offload image data to u in terms of *image packet* when its free storage satisfies following condition.

$$rs_v = rs_{min} \wedge (rs_v < rs_u) \wedge (rs_u - rs_v > \alpha \times rs_v) \quad (7)$$

where rs_v and rs_u are the free storage of nodes v and u, respectively. rs_{min} is minimum free storage space in the items of node v's metadata table, in which their *updatedTimes* are close to current time (We set 5 min in our work). α is a configurable parameter which determines how sensitive the node are to storage imbalance. The amount of data to be transferred from v to u, denoted by D_{vu}, is simply defined as (Note: a node will offload data from oldest to fresh in time series):

$$D_{vu} = (rs_u - rs_v)/2 \quad (8)$$

In summary, nodes in VStore first exchange metadata of nodes and image tags in terms of metadata packet during a connection. Then, whether they need to offload data depends on their current free storage. In addition, fresh image files will replace the oldest files by FIFO when a node has no storage. Overall, VStore utilizes some basic information for redundancy elimination and storage balancing, such as image tags and node metadata information, etc. We try to make the two sub-mechanisms simple, effective and avoid arbitrary because we mainly want to validate the feasibility of this new paradigm/architecture for urban mobile surveillance in the first stage. It is worth noting that, more information can be utilized, with which some sophisticated schemes can be designed based on VStore presented in this paper, we state it as a part of our future work.

4. Overview of data processing

4.1. Data background

Mobility of nodes has an important impact on the performance of a MANET. Different approaches have been used to obtain node trace in order to generate a mobility model, such as Random Waypoint, etc. In recent years, a number of researchers have studied how to obtain realistic mobility model based on traces of real subjects, such as human and vehicles. In our work, we also plan to utilize real traces of vehicles so that it could increase the solidity of the performance results of VStore. Currently, thousands of taxis in Shanghai have been equipped with GPS sensors for various purposes. For example, taxi companies could locate their taxis for effective vehicle dispatching, which essentially improve the efficiency of taxi service system. Normally, the real-time GPS data is uploaded from taxis to a data center by wireless communication, such as GSM, in which the typical sampling interval is around 1–2 min because it is sufficient to support vehicle dispatching. At the same time, the drivers of taxis also prefer to a long sampling interval as they need to pay GSM communication by data volume. Thus, the raw data is represented as discrete location points with large time interval. In our project, we could utilize GPS data of thousands of taxis in Shanghai, China. Fig. 3 shows a snapshot of taxi distribution in inner loop area of Shanghai, from which we can see a large-scale vehicular network can be constructed if we assume that vehicles could communicate with each other.

4.2. Data processing procedure

Basically, three sub-procedures will be involved with data processing of raw GPS data: map-matching, routing between two locations, and data interpolation. Map-matching is to map GPS data onto correct road section facing well-known GPS error. Since there is a large sampling interval, the coordinates of two consecutive GPS data (from a same vehicle) may be far from each other (e.g., several streets/blocks), how to decide the traveling path between these two coordinates is done by routing sub-procedure. Last, data interpolation operation is needed to make the position of vehicle at

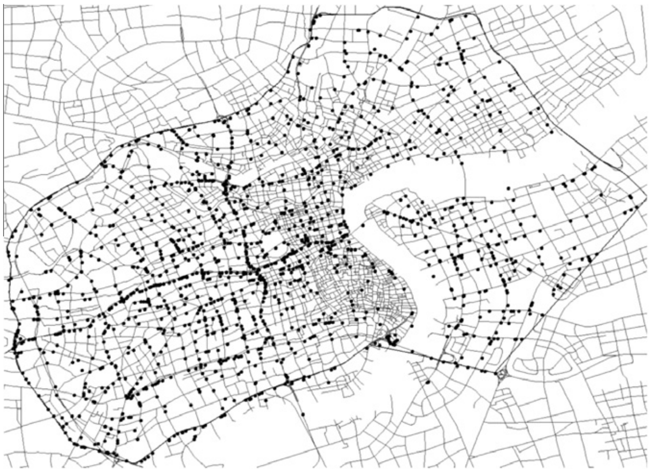


Fig. 3. A snapshot of taxi distribution. The surrounded area depicts the inner loop area of Shanghai. Every dot represents a taxi.

every second becomes available. In the following section, we will briefly introduce three sub-procedures [13,23,24].

4.2.1. Map-matching

In Section 2, we showed road network model. Actually, in a digital map file, each road section (i.e., link in road network model) is implemented as a B-graph object (white bounding box in Fig. 4), in which polyline is used to describe the geometry of roads), as illustrated in Fig. 4. For a given GPS data, it may fall into a wrong B-graph or resides multiple B-graphs because of GPS error, so we need to design an effective map-matching algorithm to rectify data onto correct road sections. In city urban area, several factors will degrade the accuracy of map-matching algorithm, such as dense road network, line-of-sight effect, etc.

4.2.2. Routing between two locations

After map-matching, the next step is to determine the path between two locations. We observed that taxis do not always follow the shortest path and we developed an angle-based heuristic algorithm following the principle of minimizing turns, which is based on a survey from taxi drivers. In this algorithm, we assume that taxis always keep on running on their current roads until they have to make a turn towards their destinations.

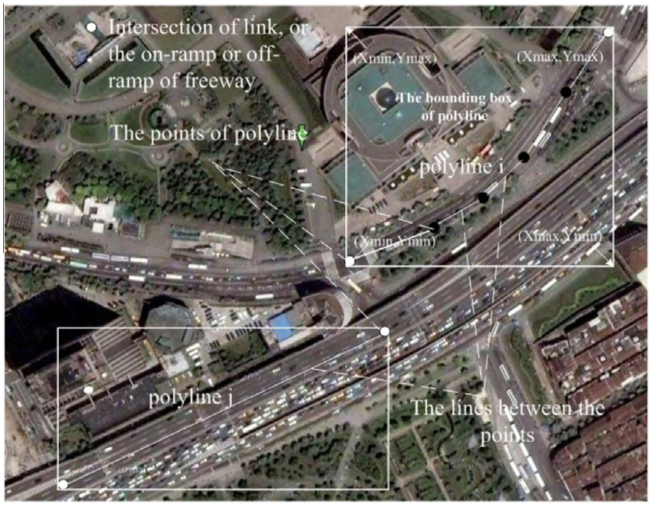


Fig. 4. Illustration of a B-graph.

4.2.3. Data interpolation

Last, by calculating the time interval Δt (second) and the length Δs (meter) of the path between two consecutive data, we could generate intermittent data for every $\Delta s/\Delta t$ m. Now, for a given vehicle, we obtain its locations at each second. To validate the accuracy of data interpolation procedure, we used video camera to record some taxi traces and compared the interpolated data with the real traces from the video, we found that there is no considerable difference between two traces and it is sufficient for us to support simulation of mobile surveillance application presented in this paper.

5. Performance evaluation

We developed a trace-driven simulator (we feed the massive GPS data into the simulator as input), named Shanghai Urban Vehicular Networks (SUVNet), in which various components have been built. With SUVNet, we have already studied lots of interesting research topics, ranging from vehicular ad hoc networks (DTN routing protocol design, connectivity in VANET, mobility modeling) to intelligent transportation systems (traffic monitoring, real-time navigation, etc.). In SUVNet, the data pre-process module has implemented the procedures as introduced in Section 4. For example, it is used to handle GPS error problem and more than 90% GPS can be located on the right road sections by effective map-matching algorithm. This module provides reliable node traces for the wireless communication module and provides accurate location data for the traffic monitoring module. The communication module is to model the wireless communication environment, which enables us to design and evaluate routing and data dissemination protocols on SUVNet. In this section, we will present the testing results of VStore on the SUVNet platform. The primary goal of VStore is to maximize the storage capacity of network. Two sub-mechanisms in VStore are proposed in Section 3: Cooperative recording and redundancy elimination, and distributed storage balancing. In this section, we present performance evaluation results of VStore.

5.1. Experimental setup

The testing is based on SUVNet simulator utilizing realistic taxi traces, as described in last section. We compare VStore with BASELINE, which is without cooperative storage scheme as in VStore. In BASELINE, it is assumed that vehicles only keep proactive sensing and store image files in their local storages. Table 1 lists some properties of tested area and the default parameters used for all the experiments in our simulation. We set the values of the experimental parameters based on field test in Shanghai urban area, such as maximum photographic distance, communication range, etc. For example, we set maximum photographic distance with 30 m. Also, we consider that *valid time* = 60 s is enough for mobile surveillance application. We set bandwidth = 20 Mbps as used in

Table 1
Default experiment parameters.

Focused area	50 km^2
Simulation time duration	3600 s
Number of taxi (n)	500
Communication range (cr)	100 m
Max. photographic dist. (mpd)	30 m
Valid time (τ)	60 s
TTL of tag (TTL)	60 s
Packet size	256 KB
Bandwidth	20 Mbps
Storage size on every node	250 packets

[11], in which the authors proved that for a 54 Mbps 802.11 g wireless link, in practice the effective payload throughput is only about 20 Mbps. Meanwhile, we predict some real technology will support higher bandwidth and communication range requirement soon for vehicular network scenario. The packet size is assumed constant 256 KB for simplicity, including metadata packet and the image packet (we assume every image captured by camera is 800 × 600, JPEG format less than 256 KB, thus one image can be transferred in one image packet). Moreover, compared with image, the tag information and the metadata of node are much smaller, which can be only tens of bytes. In our work, every tag is 40 bytes including related attributes as defined in Definition 2. Each item in metadata table is 20 bytes as defined in Definition 3. Each data point in all figures is averaged over 30 runs. In addition, in our work, we mainly focus on an area of 50 km^2 in downtown because of high taxi density, as shown in Fig. 5 and we selected vehicles traces that are in the focused region during the most of simulation time. With the communication module in SUVNet, we set the connection-establish time to be one second as used in [22] before data transmission between two nodes. At the same time, if there is already an ongoing communication and the neighbor is still connected, the sender immediately goes on with the transmission. If the connection is broken, the sender will stop the transmission, and the receiver drops the suspended packet.

5.2. Experiment results

To illustrate how VStore maximizes the storage capacity via redundancy elimination and storage balancing, Fig. 6 first plots free storage space changes over time with BASELINE and VStore schemes, respectively (initially, the network storage capacity is 250 × 500 = 125000 packets). As seen in Fig. 6(a) and (b), for all values of *mpd*, VStore essentially delays the time when the global network storage comes to saturation. For example, with *mpd* = 30 m, the saturation time is delayed from 1089 s in BASELINE to 1947 s in VStore. Meantime, it is shown that as *mpd* decreases, both BASELINE and VStore have higher storage consumption rate. The underlying reason is that the sampling rate becomes higher with *mpd* decreasing, as defined in Eq. (6), which incurs more image files to be generated in the network.

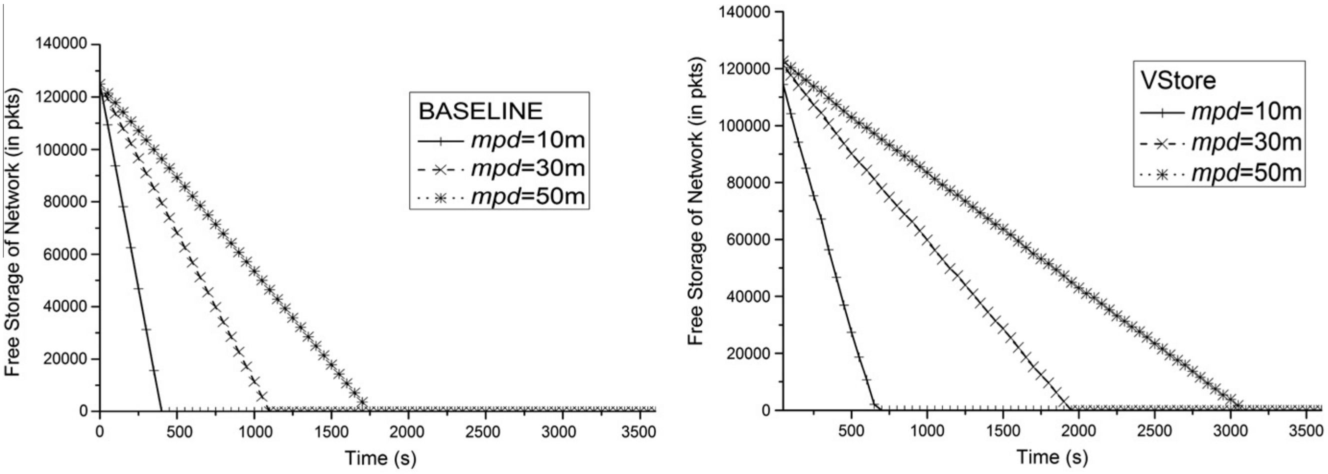
Fig. 5. Visualization of the focused area (every red dot represents a vehicle).

For the average lifetime of images, we present the comparison in Fig. 7(a). We used VStore with infinite storage on every node as an OPTIMAL mechanism. Thus, all the image files can be stored in the network without data loss because there is no storage overflow. In Fig. 7(a), we show the average lifetime $\Omega(3600)$ by Eq. (4) with three different schemes. Obviously, the average lifetime is 3600 s in OPTIMAL because of infinite storage capacity. Meantime, the average lifetime in VStore is 3272 s, which is about 90.8% of the average lifetime in OPTIMAL. Note that, such performance is based on limited storage space. BASELINE, however, performs worst in three mechanisms, in which the average lifetime of packet is only 1744 s. It means that the network can only store images which were generated during [1856 s, 3600 s] when $t = 3600$. The images generated during [0 s, 1855 s] were already deleted because of the FIFO replacement policy. Based on above analysis, we have shown the effectiveness of VStore in prolonging the average lifetime of images.

We also explore communication overhead of spreading tags and metadata between nodes in terms of metadata packet. The total number of metadata packet sent per second is about 125.7 on average, as plotted in Fig. 7(b). In other words, the bandwidth cost for each node is only 0.25 packet/s, which is acceptable for our application scenario because metadata packets only consume a fraction of bandwidth for redundancy elimination and information exchanging. In the following section, we present the testing results to analyze the effectiveness of two sub-mechanisms separately, i.e., redundancy elimination and storage balancing. Specially, we assume every node has an *unlimited* storage capacity, which is easy for us to evaluate their performances. BASELINE is still tested with default parameters in all simulations.

To investigate the performance of redundancy elimination, Fig. 8 (a) and (b) plot redundancy ratio with different communication ranges *cr* and vehicle numbers *n*. As shown in Fig. 8(a), we found with default communication range (i.e., 100 m), the redundancy ratio of VStore is about 10% whereas BASELINE is about 53%, more than 80% reduction of redundancy ratio can be achieved by VStore. At the same time, we see a drop in redundancy ratio of VStore with increase of communication range. We give the explanation as follows: for a particular valid time τ (For example, $\tau = 60$ s), as *cr* increases, the tags of image files can be propagated in a larger region than a small *cr*. Thus, more nodes can receive tags for redundancy elimination, which leads to a low redundancy ratio. We point out that, however, for a particular τ and an image *im*, essential improvement cannot be expected as *cr* is larger than a threshold. The main reason is that it is unnecessary to disseminate tags to nodes far away from the street, from which *im* was generated. In other words, it is almost impossible for remote vehicles to generate a same image in a short time interval.

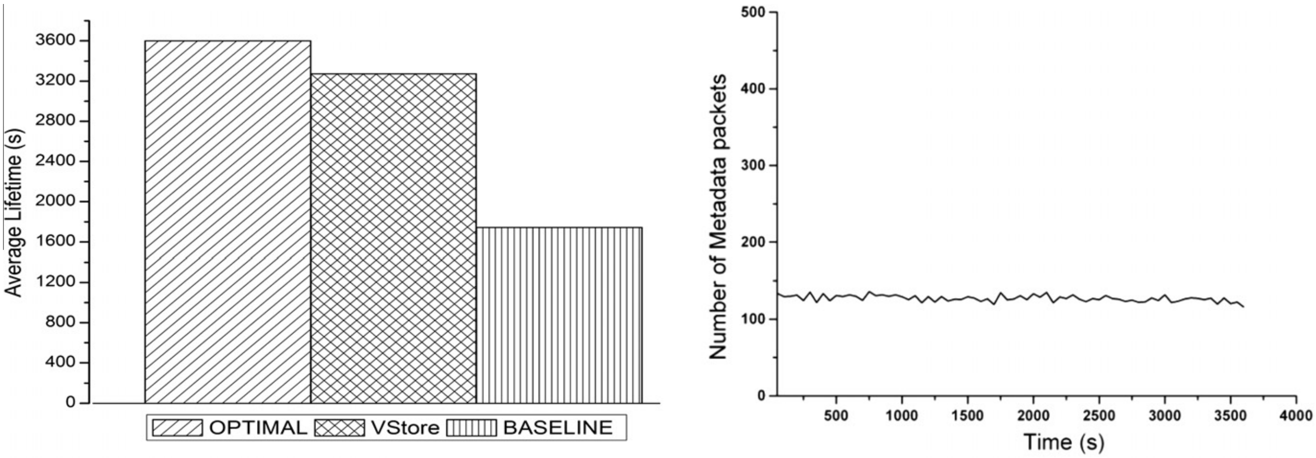
Fig. 8(b) shows redundancy ratio changes as the size of network increases from 125 to 500. With increase of *n*, we can see the redundancy ratio also keeps increasing. As mentioned in Section 3, an inherent assumption of VStore is that a link segment can be recorded by more than one vehicle. To be precise, the higher the node density is, the more effective the VStore will be. As shown in Fig. 8(b), the redundancy ratio is only around 5% when there are 125 nodes in the network. In this situation, it is true that VStore will contribute to redundancy elimination, but more importantly, most of segments can only be recorded once during time duration τ because of sparse node density. Actually, there are about 67000 taxis and 50000 buses in Shanghai and the real vehicle density is much higher than the vehicle density in our simulation scenario. At the same time, we predict that more and more public transport will be equipped with sensing devices for mobile surveillance, which support the applicability of VStore. Overall, VStore can achieve 80% reduction of redundancy ratio, which enables average lifetime of image to increase by more than 76%.



(a)

(b)

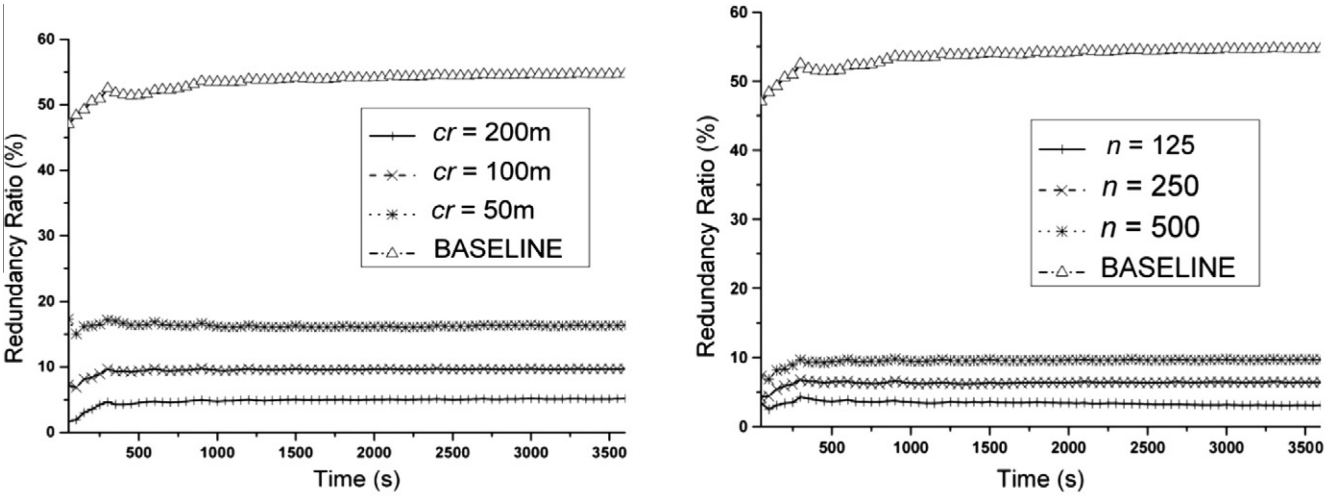
Fig. 6. (a) The free storage consumption in BASELINE. (b) The free storage consumption in VStore.



(a)

(b)

Fig. 7. (a) The average lifetime of three mechanisms. (b) Communication overhead of exchanging metadata packet, including tags and metadata of nodes.



(a)

(b)

Fig. 8. (a) The redundancy ratio with different *cr*. (b) The redundancy ratio with different *n*.

Next we investigate the relationship between redundancy elimination and storage balancing. As shown in Fig. 9(a), when α increases from 0.1 to 1, a lower redundancy ratio we can get. The reason is that with limited bandwidth, a node will offload data frequently if it is sensitive to storage imbalance (small α). Accordingly, it will also keep longer contact time to transfer image packets. As a result, image tags will be propagated to fewer nodes before their *TTLs* and some duplicated images cannot be eliminated. Conversely, if we set α with a large value, a fairly low redundancy ratio can be achieved but there will be a large load imbalance between nodes. Fig. 9(b) shows the data distributions with different α. The smaller the α is, the more flat data distribution we can observe, i.e., there is a tradeoff between redundancy elimination and storage balancing. In addition, we found that communication range and bandwidth also have an essential impact on storage balancing. For example, with a small *cr*, a connection between two nodes will be disrupted even if they still did not finish data transferring.

From Fig. 9 (a) and (b), we have seen that how to achieve flat data distribution with distributed storage balancing. At the same time, we have explained that why both low redundancy ratio and flat data distribution cannot be obtained simultaneously, i.e., distributed storage balancing will hinder image tag dissemination. In addition, due to node's mobility, we found that distributed storage balancing is a global process because storage balancing can be executed in any of two vehicles once they meet each other. In that case, storage balancing can happen throughout urban area because of vehicle's movement.

In addition, we should point out that in our simulation, we basically selected the vehicles, which did not leave the network in most of simulation time. Accordingly, it is not difficult to control data replications. In the reality, however, due to real-time availability of vehicles, e.g., enter/exit a region, data loss may become an issue. In the meanwhile, since different applications need various QoS requirements, how to guarantee that the requested data can be always ready to use by considering the possible unavailability of vehicles needs to be handled. Another issue is about data query, which is not studied in this paper. Actually, an interesting problem can be how to control data redundancy degree in order to provide a reasonable response time for a given in-network query. The issues addressed above are all interesting topics and will be well studied in our future work.

5.3. Addressing some realistic issues and considerations

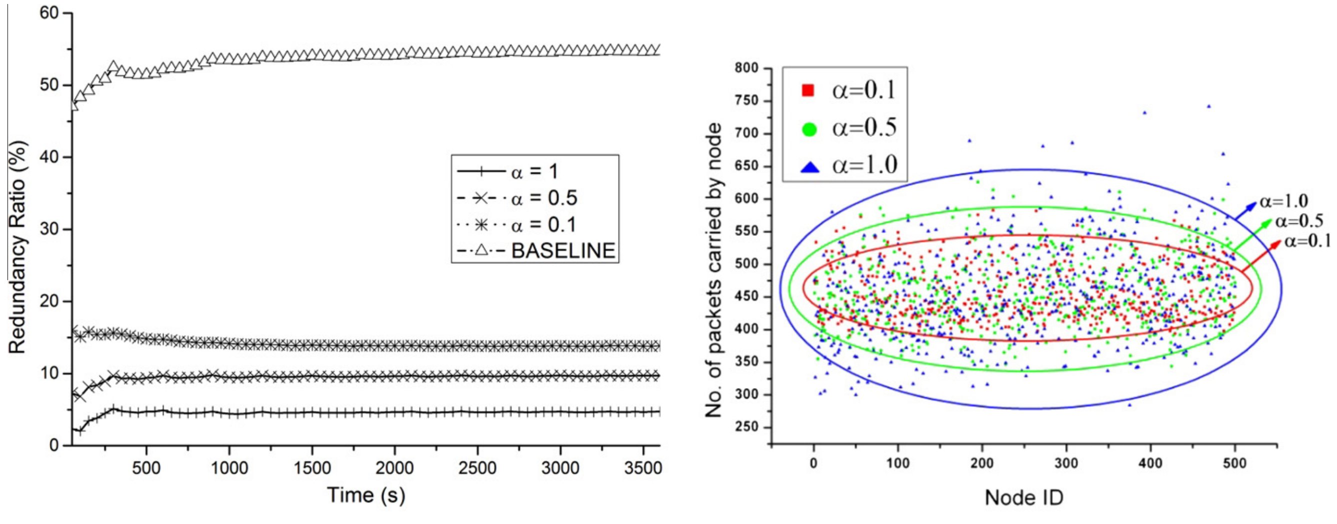
5.3.1. About Storage size

5.3.1.1. From an application-level standpoint. The storage size has impact on usability of the cooperative storage scheme. Actually, from an application-level standpoint, it is essential to evaluate that whether the storage optimization is necessary based on specific applications and scenarios. As an example, if an application never needs images older than one week and the local storage is sufficient for keeping images that relate to the last month, the topic of storage optimization may become less interesting in this application. Hence, it gives us an implication that for a specific application, we should first estimate whether the basic network storage capacity can satisfy application requirements before deploying a VStore-based cooperative storage system.

In this paper, we mainly proposed a general architecture for urban surveillance, in which specific applications are still not involved. In other words, an implied assumption in this paper is that the basic storage capacity cannot meet application requirements without cooperative storage scheme. In that case, we mainly want to validate how about the performances of two sub-mechanisms in redundancy elimination and load balancing without an application-level consideration that whether VStore can effectively support specific applications.

5.3.1.2. An implementation/test consideration. By comparison, in our simulation, we set the storage size on every node to 250 packets. This assumption, however, is not that true in real world. The reason of setting a small storage size is just to validate the performance of VStore during simulation. Actually, the redundancy elimination mechanism can be executed all the time. The storage balancing mechanism, however, can only take into effect when some nodes do not have sufficient storage space. A large storage size will delay the time when the storage balancing mechanism starts to work. In this situation, we cannot see the effectiveness of distributed storage balancing mechanism because most of nodes still have sufficient free storage even if we run the simulation for a long time.

In order to validate small value of storage size does not impair our testing results presented in last section, we re-run a long-time simulation (around 10 h), and the storage size on every node is set to be 2500 packets (compared with default value in our simulation, i.e., 250 packets) at this time (Initially, the network storage capac-



(a) (b)

Fig. 9. (a) The redundancy ratio with different α. (b) The data distribution between nodes with different α.

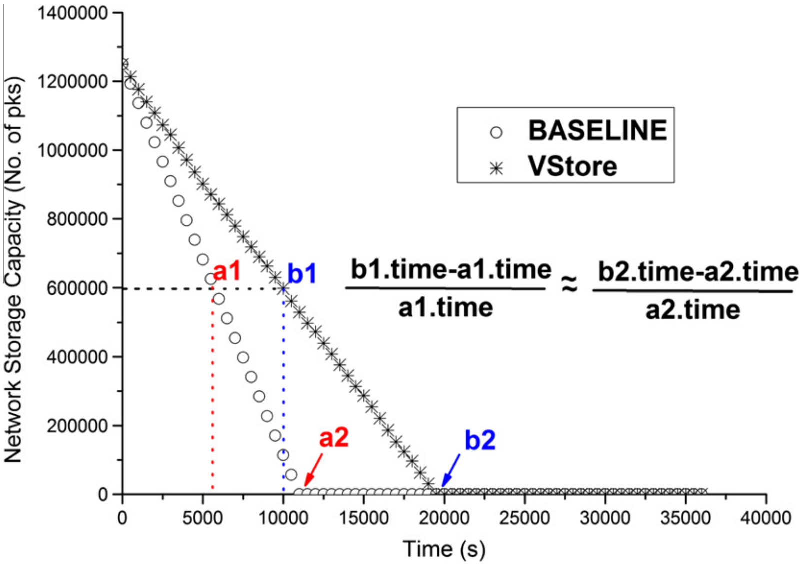


Fig. 10. The free storage consumption in BASELINE and VStore: a1 and b1 are the network saturation time points with BASELINE/VStore when network storage capacity is 60000 packets, respectively; a2 and b2 are the network saturation time points with BASELINE/VStore when network storage capacity is 1250000 packets, respectively.

ity is $2500 \times 500 = 1250000$ packets). Note that, in this experiment, the trace of each vehicle needs to be synthetically generated by overlapping several other traces because we need to keep each vehicle do not leave the network in most of simulation time. As shown in Fig. 10, we found that the two curves keep linear decrease with time, which is similar with Fig. 5. For example, if we assume network storage capacity is half of 1250000 packets, i.e., around 600000, the improvement of VStore over BASELINE is nearly the same as the improvement when the network storage capacity is 1250000 packets. Due to the geometry property of similar triangles, it is easy to see that different storage capacities do not affect performance improvement of VStore, which explains that why we choose a relative small value in our simulation.

5.3.2. About bandwidth setting

In our work, the bandwidth is set to be 20 Mbps as used in [11] and we predict some real technology will support higher bandwidth and communication range requirement in the future. We should admit that 20 Mbps is still an optimistic assumption in vehicular ad hoc networks scenario. The main reason of using high bandwidth assumption is that the distributed storage balancing mechanism of VStore needs high bandwidth to transfer image data between nodes. In other words, distribute storage balancing will not work well with low bandwidth because large volume image cannot be redistributed between nodes. At the same time, we notice that it will be a long way to go before large-scale real deployment of mobile surveillance and we hope some promising wireless communication products can support high-bandwidth in the near future. In addition, we re-examined the simulation and found that in each contact between two nodes, the contact duration/time is different. Specifically, in VStore, the two nodes cannot exchange at will even if high-bandwidth link and long contact time are available during a contact, e.g., Eq. (7) explains that whether two nodes will hold a connection depends on the current storage status of them.

6. Conclusion

In this paper, we proposed VStore, a new paradigm for urban mobile surveillance with vehicular sensor networks (VSN). The critical issue is how to maximize storage capacity of network, i.e., maximize the lifetime of sensory data by cooperative storage. Compared with previous works, we propose a novel storage architecture for urban surveillance and deal with challenges in a mobile scenario. With proactive sensing by forward facing cameras, nodes in VStore first capture images from links/streets and then eliminate redundant data by exchanging image tags between vehicles. At the same time, VStore also includes a distributed storage balancing mechanism to offload data from heavy-load nodes to light-load nodes. We carried out our testing on a trace-driven simulator. The results show that VStore can largely prolong the average lifetime of image data by redundancy elimination and storage balancing. Nearly 80% reduction of redundancy ratio can be achieved, which enables the average lifetime of image to increase by more than 76%. We point out that the technical solution presented in this paper is simple and the main purpose of this work is to introduce a new application of vehicular sensor networks and provide basic experience and reference/guide for real deployment.

Acknowledgements

The authors are grateful for a variety of valuable comments from the anonymous reviewers. This research was partially supported by NSF of China Grant No. 61073158, 60773091 and 61003247, the National 973 Program of China under Grant No. 2006CB303000, the National 863 Program Grant No. 2006AA01Z247, Jilin University Grant No. 200903194, and the NSF of USA Grant No. CNR-0626380. Dr. Xu Li was supported by NSF of USA Grant No. NSF-CPS-1035733.

References

[1] W. Zhang, G. Cao, T.L. Porta, Data dissemination with ring-based index for wireless sensor networks, IEEE Trans. Mobile Comput. 6 (7) (2007) 832–847.
[2] Q. Fang, J. Gao, and L. Guibas, Landmark-based information storage and retrieval in sensor networks, in: Proceedings of INFOCOM, (2006).
[3] M. Li, D. Ganesan, and P. Shenoy, Presto: feedback-driven data management in sensor network, in: Proceedings of NSDI, (2006).
[4] P. Desnoyers, D. Ganesan, and P. Shenoy, Tsar: a two tier sensor storage architecture using interval skip graphs, in: Proceedings of SENSYS, (2005).
[5] D. Ganesan, B. Greenstein, An Evaluation of Multi-resolution Storage for Sensor Networks, in: Proceedings of SENSYS, (2003).
[6] S.R. Madden, M.J. Franklin, Tinydb: an acquisitional query processing system for sensor networks, ACM Trans. Database Syst. (2005)

[7] L.Q. Luo, C.D. Huang, EnviroStore: a cooperative storage system for disconnected operation in sensor networks, in: Proceedings of INFOCOM, (2007).

[8] L.Q. Luo, Q.Cao, EnviroMic: towards cooperative storage and retrieval in audio sensor networks, in: Proceedings of ICDCS, (2007).

[9] I.F. Akyildiz, A survey on wireless multimedia sensor networks, Elsevier Comput. Netw. 51 (4) (2007) 921–960.

[10] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, A. Corradi, Dissemination and harvesting of urban data using vehicular sensor platforms, IEEE Trans. Veh. Technol. 58 (2) (2009) 882–901.

[11] S. Greenhill, S. Venkatesh, Distributed query processing for mobile surveillance, in: Proceedings of ACM Multimedia, (2007).

[12] S. Greenhill, S. Venkatesh, Virtual observers in a mobile surveillance system, in: Proceedings of ACM Multimedia, (2006).

[13] H.-Y. Huang, P.-E. Luo, X. Li, Performance evaluation of SUVnet with real-time traffic data, IEEE Trans. Veh. Technol. 56 (6) (2007).

[14] U. Lee, M. Gerla, A survey of urban vehicular sensing platforms, Elsevier Comput. Netw. 54 (4) (2010) 527–544.

[15] U. Lee, R. Cheung, M. Gerla, Emerging Vehicular Applications Vehicular Networks: From Theory to Practice, CRC Press, Taylor & Francis Group, 2009.

[16] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, A. Corradi, MobEyes: smart M obs for urban monitoring with a vehicular sensor network, IEEE Wireless Commun. 13 (5) (2006) 52–57.

[17] <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.

[18] B. Hull, V. Bychkovsky, CarTel: a distributed mobile sensor computing system, in: Proceedings of ACM SENSYS, (2006).

[19] J. Eriksson, H. Balakrishnan, S. Madden. Cabernet: Vehicular content delivery using WiFi, in Proceedings of 14th ACM MOBICOM, (2008).

[20] A. Thiagarajan, L. Ravindranath, VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones, in: Proceedings of ACM SENSYS, (2009).

[21] M. Seligman, K. Fall, P. Mundur, Storage routing for DTN congestion control, in: Proceedings of WCMC, (2007).

[22] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, A measurement study of vehicular internet access using in situ Wi-Fi networks, in: Proceedings of ACM MOBICOM, (2006).

[23] X. Li, W. Shu, M. Li, et al., Performance evaluation of vehicle-based mobile sensor networks for traffic monitoring, IEEE Trans. Veh. Technol. 58 (4) (2009) 1647–1653.

[24] H. Huang, Y. Zhu, X. Li, META: a mobility model of metropolitan taxis extracted from GPS traces, IEEE Wireless Commun. Netw. Conference (WCNC'10), (2010).