Distributed or Centralized Traffic Advisory Systems—The Application's Take

John S. Otto and Fabián E. Bustamante
Department of Electrical Engineering and Computer Science
Northwestern University
{jotto,fabianb}@eecs.northwestern.edu

***Abstract*—We consider the problem of data dissemination in vehicular networks. Our main goal is to compare the application-level performance of fully distributed and centralized data dissemination approaches in the context of traffic advisory systems.**

Vehicular networks are emerging as a new distributed system environment with myriad promising applications. Wirelessly-connected, GPS-equipped vehicles can be used, for instance, as probes for traffic advisory or pavement condition information services with significant improvements in cost, coverage and accuracy. There is an ongoing discussion on the pros and cons of alternative approaches to data distribution for these applications. Proposed centralized, or infrastructure-based, models rely on road-side equipment to upload information to a central location for later use. Distributed approaches take advantage of the direct exchanges between participating vehicles to achieve higher scalability at the potential cost of data consistency. While distributed solutions can significantly reduce infrastructures' deployment and maintenance costs, it is unclear what the impact of "imprecise" information is to an application or what level of adoption is needed for this model to be effective.

This paper investigates the inherent trade-offs in the adoption of distributed or centralized approaches to a traffic advisory service, a commonly proposed application. We based our analysis on a measurements study of signal propagation in urban settings and an extensive simulation-based experimentation in the Chicago road network.

I. Introduction

Vehicular networks are emerging as a new distributed system platforms with myriad promising applications. Wirelessly-connected, GPS-equipped vehicles can be used, for instance, as probes for traffic advisory or pavement condition information services, improving their coverage and accuracy while reducing their costs [1]–[7].

While diverse in goals, the architecture of most proposed vehicular network systems adopts either a centralized or a fully distributed model for data collection and dissemination. Centralized, or infrastructure-based, solutions rely on road-side equipment to upload information to a central location for later use (e.g. [3], [8]). Distributed, cooperative solutions, on the other hand, take advantage of the direct exchanges between participating vehicles to achieve higher scalability at the potential cost of data consistency (e.g. [9], [10]). While distributed solutions can significantly reduce infrastructures' deployment and maintenance costs, it is unclear what level of adoption is needed for this model to be effective and what is the impact of "imprecise" information to an application. There is an ongoing discussion on the pros and cons of these alternative models. This paper contributes to the conversation a comparative analysis of the application-level trade-offs of fully distributed and centralized data dissemination approaches for vehicular networks.

We carried out our study in the context of traffic advisory and vehicular navigation systems. The outstanding and mostly unplanned growth in the world's urban population is severely impacting our quality of life [11]. A Chicago driver will spend an average of 56 hours in traffic delays each year, with a total annual cost of $4 billion. This increase in traffic has had a devastating impact on the environment, responsible for 50% of the air pollution in Chicago [12]. Traffic advisory services have the potential to reduce the impact of vehicular traffic through more efficient navigation.

For this study, we designed and implemented basic centralized and distributed data dissemination algorithms. To compare the application-level trade-offs of each approach, we employ several dynamic traffic routing algorithms and compare their performance in terms of total travel time and distance traveled relative to a node using a static traffic routing algorithm.

The centralized and distributed systems trade-off between time savings and distance overhead. While the centralized system's "global" view yields greater time savings than the distributed system's "local" perspective, it also comes at the cost of longer routes. For instance, 20% of the centralized system's routes are at least 10% longer than the static node's route. In comparison, the distributed system only incurs this much overhead in 5% of its, races.

Centralized system performance depends on the distribution of access points (APs) that allow communication with a centralized server. In a planned deployment, where the placement and density of APs can be controlled, an evenly-spaced grid pattern will give the best performance. For example, with only 12 APs/km^2 the system achieves average time savings of 18%. Centralized systems relying on opportunistic connections require higher density of APs to reach similar performance level as a planned deployment of APs. In a dense urban environment with high AP density (e.g. 126 APs/km^2), average time savings can match that of a planned deployment (23%). However, in the case of a region with lower AP density (12 APs/km^2), time savings is only 12%–two-thirds the performance of a similar system with a planned deployment of APs.

Distributed data dissemination offers the reliability, scalability and cost benefits of distributed solutions, without negatively impacting application performance in terms of travel time and distance, relative to the centralized system. Given information about alternative road segments along their current route, routing based on a distributed data dissemination model can achieve time savings up to 14%. This performance is comparable to that of the centralized system under a low-density planned AP deployment (18%) or a realistic, non-uniform distribution of APs (12%).

The following section briefly reviews the background and context of our work. Section III presents the design of two basic centralized and distributed data dissemination protocols and Section IV describes the set of traffic routing algorithms we used to compare them. We describe our experimental setup in Section V and present the result of our evaluation in Section VI. We conclude in Section VII.

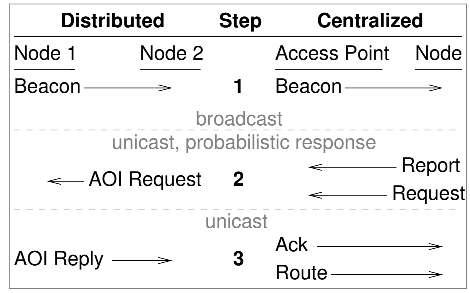


Fig. 1. Communication diagram for the distributed and centralized data dissemination models. In the distributed system, nodes send Area Of Interest (AOI) Requests for recent traffic measurements, and receive an AOI Reply. For the centralized implementation, nodes opportunistically communicate with the centralized server via an access point (AP). Nodes send Report messages of their measurements, and Request messages that ask for an updated route. The centralized server responds, via the AP, with Acknowledgment messages of measurements and Route messages.

II. Background

Traffic congestion problems are, regrettably, an unavoidable part of our daily lives. According to a recent report, in 2005 alone US drivers wasted over four billion hours and nearly three billion gallons of fuel due to traffic delays [13]. Most early work on Intelligent Transportation Systems (ITS) was motivated by such problems associated with traffic congestion. Research focus on ITS has slowly moved toward remote traffic monitoring and incident detection using in-road sensors such as traffic cameras and loop detectors. Induction loop detectors are one of the most common and better understood monitoring tools. Despite their popularity, they remain expensive to deploy and maintain (price ranges from $900 to $2000 per sensor) and are known to yield very noisy and generally inaccurate velocity measurements [14]. While alternative infrastructure-based techniques have been proposed, such as the use of in-vehicle transponders (e.g. EZPass), license plate readers, and radar, they all incur high installation and maintenance costs that have limited their widespread implementation. For instance, the New York City Department of Transportation relies only on 22 cameras to monitor the traffic for the whole city [15].

The known limitations of such traditional sensors [14] and the increasing availability of wirelessly-connected, GPS-equipped vehicles has recently spawned a number of projects evaluating the use of vehicles as probes for measuring traffic state [2], [3], [5], [6], [10], [16], [17]. While some current commercial vehicular communication systems [8], [18], [19] rely on the cellular network for basic communication, it is unclear if this would be sufficient in terms of coverage, scalability and capacity, to support more demanding applications.

To the best of our knowledge, this is the first work to examine the trade-offs between distributed and centralized data dissemination models from an application perspective.

III. Data Dissemination Design

The data distribution services behind most existing vehicular-based distributed systems adopt either a centralized or a fully distributed model for data collection and dissemination. With a centralized approach, vehicles rely on road-side infrastructure, either in a planned [20], [21] or in an opportunistic manner [3], to communicate with a central location. Most existing ITS deployments follow a planned centralized approach, relying on roadside sensors, cameras and network access points (APs). While promising, their high deployment and operational costs prevent them from reaching their fullest potential.

Some recent proposals for vehicular-based systems using centralized models avoid such costs by opportunistically taking advantage of already deployed open APs (e.g. [3], [7], [22]). In general, however, these architectures bring with them the classic problems of scalability and resilience associated with any centralized solution, in addition to less obvious privacy issues [2].

When adopting a distributed, cooperative model (e.g. [9], [10]) for data dissemination, vehicles depend solely on inter-vehicle communication. Although distributed solutions could avoid the problems associated with infrastructure deployment and maintenance, it is unclear how quickly such a system can be bootstrapped or what level of adoption is necessary for the model to be effective.

Our goal is to compare the application-level trade-offs of fully distributed and centralized data dissemination approaches in the context of traffic advisory systems. To facilitate our study, we first introduce two basic, pull-based distributed and centralized protocol designs for data dissemination. Figure 1 illustrates both protocols, showing the communication steps for every interaction with a common framework that highlights the parallelism between them.

A. Distributed Data Dissemination

Our basic distributed data dissemination protocol has three steps. Every vehicle beacons to advertise its presence. Any vehicle can request (pull) traffic information from a nearby peer. Vehicles respond to traffic update requests the most recent measurements for the requested sections of the map. Hereafter, we use *vehicles* or *nodes* when referring to the instrumented vehicles participating in our system.

Step 1: Beacon. Every node in the system periodically sends a *beacon* message. Node beaconing allows surrounding nodes to learn of the presence of others and to estimate how many of them are within communication range. A node receiving a beacon probabilistically determines whether or not to respond according to the inverse of the number of beacons received in the last beacon interval. This improves system scalability under high node density by limiting the number of nodes responding to a beacon.

Step 2: Area of Interest Request. If a node decides to respond to a beacon, it sends an *Area of Interest (AOI) Request* message to the beaconing node which includes a list of road segments for which it requests updated state information.

Step 3: Area of Interest Reply. When a node receives an *AOI Request*, it compiles a reply (an *AOI Reply*) with the most recent data for each road segment requested. The actual data included (e.g. average temporal speed), will depend on the routing algorithm in use, but it is extracted from the responding node's local estimation of global traffic conditions. A vehicle's local view of global conditions is derived from the data reported by all vehicles with which it has previously been in contact. The node receiving the *AOI Reply* incorporates the new data into its estimation of global traffic conditions and, potentially, recomputes its route.

B. Centralized Data Dissemination

As in the distributed model, vehicles accumulate measurements for each road segment they traverse. Under a centralized architecture, vehicles report all recently collected measurements to the next access point (AP), which in turns reports it to the central location. Vehicles can also use the exchange to request route updates. The centralized location makes routing decisions, upon request, based on the reports from *all* instrumented vehicles. Both the frequency at which vehicles report collected measurements and obtain updated routes is limited by the availability of APs.

Step 1: Beacon Every AP in the system periodically sends a *beacon* message. AP beaconing allows surrounding nodes to learn of the presence of APs and to estimate how many of them are in communication range. To improve scalability under high node density, a node receiving a beacon probabilistically decides whether to respond or not, according to the inverse of the number of beacons received in the last beacon interval.

Step 2: Segment Status Report and Route Request If a vehicle decides to respond to a beacon, it sends to the beaconing AP all recently collected measurements that it has not yet reported as a *Segment Status Report*. A vehicle can also send a *Route Request* that includes its current location and destination.

Step 3: Central Location Acks and Route Updates The central location responds to a node via the AP from which it sent the messages. The centralized server sends *Acknowledgment* messages for *Segment Status Reports*, which signal that the node can remove those reports from its local buffer. In the case of a *Route Request*, the centralized location uses its global view to compute the new route and return it to the requesting node as a *Route Update* message.

IV. Traffic Routing Algorithm Design

It is clearly possible to directly compare centralized and distributed approaches to data dissemination in terms, for instance, of delivery latencies or packet drop rates. Such a comparison, however, would provide little insight into the impact of either approach on the performance of a client application.

This work focuses instead on understanding the application-level implications of each data distribution model in the context of traffic advisory systems. Traffic advisory or vehicular navigation systems, an increasingly standard feature of most vehicles, rely on positioning information and map databases to provide step-by-step navigation information to the driver.

For our study, we implemented a set of dynamic routing algorithms (*Temporal Speed*, *Spatial Speed* and *Surface Street Traffic Estimation*) and a basic static routing algorithm (*Travel Time*). All algorithms employ the A* search algorithm for finding the lowest cost path to a destination given the available segment cost data. The three dynamic routing algorithms use location and bearing information, provided by ideal GPS device, to estimate segment conditions. With position information provided once per second, each instrumented vehicle can measure or infer their required parameters, such as average speed over a segment, instantaneous speed and stop duration. When exchanging segment information, the dynamic routing algorithms prioritize measurements using an exponential weighted average that gives higher priority to more recent observations. When no recent measurements are available for a given segment, the dynamic algorithms fall back on static information (e.g. road length, speed limit) to estimate travel time.

Note that available bandwidth is not a limiting factor in the performance of these traffic routing algorithms. In our experiments on city streets (Section V), using a 25 miles-per-hour speed limit an interaction lasts approximately 45 seconds over which nearly 2 MB of data can be transferred. This is at least an order of magnitude greater than what is required for any configuration of our traffic advisory system. In an *AOI Request* message, each road segment is represented by a unique 8-byte identifier; we could request information on every road segment in over 1 km^2 of a dense urban environment (300 segments) in 2.4 kB of data. The *AOI Reply* message in response to such an *AOI Request* would require about 32 kB of data, since each measurement requires about 105 bytes. In total, one interaction with a large area of interest would consume only about 35 kB of data.

The following paragraphs describe each of our routing algorithms, starting with our baseline algorithm – *Travel Time*.

A. Travel Time Routing Algorithm

Travel Time is a naïve routing algorithm that relies exclusively on static data. A road segment's length and speed limit, as well as a global estimated red light duration, are used to estimate the idealized travel time. While under light traffic conditions *Travel Time* could be sufficient, one would expect suboptimal routing performance under less ideal conditions. We use *Travel Time* as a baseline to *normalize* the performance of the dynamic routing algorithms.

B. Temporal Speed Routing Algorithm

Both *Temporal Speed* and *Spatial Speed* rely directly on information reported by GPS devices to estimate travel time. *Temporal Speed* routing is a straightforward extension of the *Travel Time* algorithm. Temporal mean speed – the length of the road divided by the total segment traversal time–gives a travel time estimate grounded in the actual experience of a vehicle.

A potential problem with temporal speed as a cost metric is that it can be significantly affected by traffic lights in urban settings. While a sufficiently large number of measurements can reduce the impact of the random delay introduced by traffic lights, this makes the approach less sensitive to subtle changes in traffic congestion.

C. Spatial Speed Routing Algorithm

Spatial Speed routing uses the average instantaneous speed (collected at 50-foot intervals along the road segment) to estimate the travel time for a segment. Since the model is insensitive to the amount of time a vehicle stopped while traversing the segment, its travel time estimate is not affected by the random red light delay. Logically then, spatial speed does not account for red light delays although these can be a significant proportion of the total trip time on urban roads.

D. Surface Street Traffic Estimation Routing Algorithm

Our last routing algorithm adapts Yoon et al.'s [6] *Surface Street Traffic Estimation* (SSTE) approach, which incorporates both temporal and spatial speed metrics and relies on a dynamically maintained model of segment condition for its estimations of travel times. Since the original proposal of the SSTE model made no prescription for how to use it for routing, we first present an interpretation of the model that enables us to perform route cost calculations based on these data. Given that SSTE incorporates both temporal and spatial speed metrics, we expect that this algorithm will offer the best performance among the alternatives.

For each road segment, SSTE aggregates a distribution of segment traces that characterizes the road segment across the range of traffic conditions. A SSTE distribution (e.g. Figure 2) is represented by a "spatio-temporal" plot, named because each segment trace is a data point plotted at its temporal mean speed on the X axis and the spatial mean speed on the Y axis.

The temporal threshold (vertical line) defines the boundary between temporal speed values corresponding to uncongested or congested traffic conditions. Variation in temporal mean speed above this threshold (A) can be attributed to the red light delay, which is estimated from the 95th percentile of the distribution of stopping durations (the duration of a stop is inferred from the GPS data). Temporal speed below this threshold (B) indicates delays greater than the maximum expected delay from waiting at the traffic light. Therefore, when a measured temporal speed is lower than the threshold, we infer that there is traffic congestion causing delays.

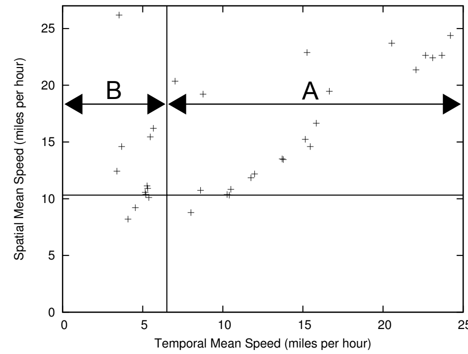


Fig. 2. A sample SSTE spatio-temporal plot taken from our simulator for a road segment with a speed limit of 25 miles per hour. Variation in temporal speed along A can be attributed to waiting at the traffic light; measurements of temporal speed along B are slower than would be expected due to only waiting at the traffic light, and are an indication of poor traffic conditions.

The spatial threshold (horizontal line) separates stop-and-go traffic from smoothly flowing traffic traces. This threshold is set at the 5th percentile of the spatial mean speed measurements on the right side of the temporal threshold. Spatial mean speeds above the spatial threshold indicate that traffic is moving smoothly, while lower spatial mean speeds imply stop-and-go traffic.

For routing we use the temporal threshold to switch between two ways of estimating travel time, based on whether we infer traffic congestion on the road segment. In the first case, when the temporal mean speed of a segment trace is below the temporal threshold, we assume that delays caused by traffic congestion dominate the red light delay. Therefore, we simply use temporal mean speed to calculate the estimated traversal time for the segment, ignoring the red light delay. We identify the second case–no significant traffic congestion–when the temporal mean speed is greater than the temporal threshold. In this scenario, the temporal speed is subject to significant variations due to a random red light delay. For this case, the trace's spatial mean speed can be used to estimate travel time if there were not a traffic light. We add half of the estimated red light delay to the travel time cost when there is a traffic light present.

An SSTE distribution must include sufficient segment traces before it can be used to reliably classify traffic state. The SSTE authors claim that a distribution having 10 data points is sufficient to achieve 90% accuracy in classifying traffic state; we adopt this value for our trace count threshold. Clearly, this threshold introduces a delay between when a node creates a distribution for a road segment and when it has aggregated enough data to be able to accurately infer traffic on the segment. To minimize this delay, a node's distribution is bootstrapped based on the SSTE distribution parameters from another node that already has a sufficient number of segment traces. Once a node has accumulated enough traces in its

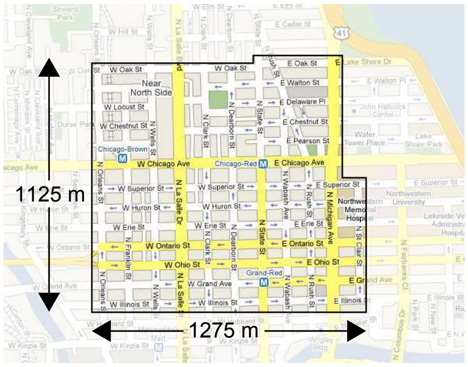


Fig. 3. The simulation region from downtown Chicago, 1275 m wide by 1125 m tall with area 1.43 km^2. Bounded by Oak to the north, Illinois to the south, St. Clair to the east and Orleans to the west.

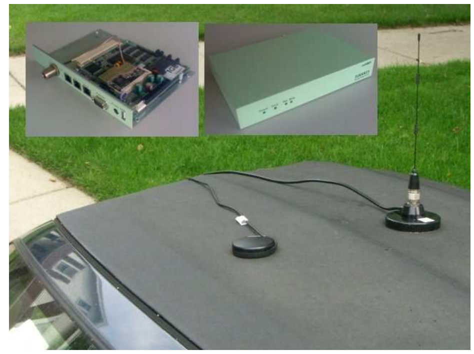


Fig. 4. GPS and radio antennas installed on the roof of a vehicle. The car's windshield is visible in the lower left-hand corner of the image. The left inset shows the internal hardware of the Soekris machine. The right inset shows the front of the node.

distribution, it includes its thresholds and model parameters when exchanging measurements with other nodes.

V. Experimental Setup

We use simulation to study the application-level performance of centralized and fully distributed data dissemination models. The following paragraphs describe our simulation framework and experimental setup. We introduce the use of *vehicular races* as a novel framework for evaluating data distribution models in the context of a traffic advisory service.

A. Simulation Framework

For simulation, we use the JiST/SWANS simulator [23]. JiST/SWANS provides an integrated, configurable, and flexible environment for evaluating ad hoc routing protocols, especially for large-scale network scenarios. It contains a detailed model of the IEEE 802.11 wireless LAN protocol and a stochastic radio channel model, both of which we use in this work. We model vehicular mobility using the STreet RAndom Waypoint (STRAW) [24] mobility model. STRAW captures realistic vehicular mobility, incorporating well-understood car-following and lane-changing models and traffic control (e.g. traffic lights, stop signs), over actual city road maps imported from the TIGER/LINE database [25]. Since its release, STRAW has been adopted by hundreds of researchers world-wide from both industry and academia.

We conduct all our simulations in a region of Chicago's downtown area, shown in Figure 3. The region has a total area of 1.43 km^2. The 277 road segments are in total 52 km long (counting each direction separately) and predominantly form a Manhattan grid pattern. To ensure traffic congestion in the map, we simulate 322 vehicles per square kilometer. Of these vehicles, we instrument 15%, yielding a density of 48 participating nodes per km^2. For the centralized system, APs are modeled as having a loss-less, low latency and infinite bandwidth connection to the centralized server. Each simulation is 3 hours long, with a 1 hour warmup period.

B. Setting Signal Propagation Parameters

The performance characteristics of the network stack's underlying physical layer defines the boundaries of a system's abilities [26]–[28]. We adopt the shadowing model with log-distance path loss [29] to model the physical layer. While simpler, deterministic models exist, e.g. free space and two-ray ground, they do not account for key aspects of real radios or capture typical VANET settings environments with complex obstacle patterns [30].

The model relies on two empirical parameters to characterize an environment: a path loss exponent and a shadowing value. The mean received signal strength decays with distance according to a power-law path loss model. The path loss exponent describes an environment-specific decay rate of signal strength. The shadowing factor models the unpredictable effects of a complex environment on signal propagation with a log-normal random variable added to the deterministic signal strength given by the path loss exponent. To appropriately configure the signal propagation model use in our evaluation, we carried on a series of measurement experiments in an urban setting, using a set of instrumented vehicles [31].

Figure 4 shows one of such vehicles. Each of them carries a Soekris [32] net4801-60 machine, with an Ubiquiti Networks [33] SuperRange2 2.4 GHz 802.11b/g mini-PCI module for wireless communications, a roof-mounted Pacific Wireless [34] 7 dBi 2.4 GHz omnidirectional antenna, and a Garmin [35] GPS 18 USB device to provide location data. The Soekris nodes run Linux and use standard utilities such as iperf [36] and tcpdump [37] to generate and record network traffic. The wireless cards were configured to operate on IEEE 802.11b channel 1 (2.412 GHz) in ad hoc mode with bitrate fixed at 2 Mb/s, request-to-send and fragmentation disabled, and with transmit power of 26 dBm.

The vehicles traveled back and forth along 3 blocks of the same street to collect line-of-sight (LOS) and non-LOS signal strength measurements across a wide range of distances. Transmitting nodes were configured to send a 1 Mb/s stream of UDP packets, while other nodes were configured to receive that stream. We categorize all packets received during our experiments into either LOS or non-LOS communication and perform aggregate analyses to estimate the model parameters using measured signal strength and distance information. Ta-

LOS		non-LOS		Combined	
β	σ_{dB}	β	σ_{dB}	β	σ_{dB}
3.17	9.15	4.05	10.74	3.43	11.95

TABLE I
MEDIAN PATH LOSS EXPONENT (β) AND SHADOWING STANDARD DEVIATION σ_{dB} FOR EACH CONFIGURATION: LINE-OF-SIGHT (LOS), NON-LOS AND COMBINED.

ble I contains the observed path loss exponent and shadowing parameters for the urban environment under LOS communication, non-LOS communication or both–Combined. Aggregating LOS and non-LOS measurements yields average path loss exponent and shadowing parameters for the environment as a whole. The shadowing parameter for the Combined data set is logically larger than that of the subsets.

We use our empirically determined parameters and the given radio configuration to simulate communication in an urban environment. For the distributed system, we opt for a conservative configuration using the Combined parameters. For the centralized model, we selected the non-LOS parameters for communication with APs. To reduce the impact of noise due to multiple communicating nodes, we adopt low penetration ratios of instrumented vehicles.

C. Race Framework

To evaluate the relative performance of routing algorithms, we set up races between two random points in the road network. The start and destination are required to be at least 1200 m apart so that races are non-trivial.

Each race involves two vehicles – one using a dynamic routing algorithm and one using the static *Travel Time* algorithm. Both vehicles are placed on the same road segment at the start location. Placement order alternates between runs. When the first race node reaches its destination, it waits for the second node to start a new race. For our analysis, we consider only races that started after 60 minutes into the simulation to give the traffic routing algorithms a chance to warm up.

D. Evaluation Metrics

We focus on understanding the application-level implications of centralized and fully distributed data dissemination models. We do this in the context of traffic advisory systems using two application-layer metrics – *Time Savings* and *Distance Overhead*.

Time savings is the percent difference between the dynamic node's time to complete the race relative to the static node's completion time. Greater time savings indicate better performance of a routing algorithm.

Distance overhead, computed in the same manner as the time savings metric, is the percent difference of the distance traveled by the dynamic traffic routing node relative to static one. Lower distance overheads are preferred.

VI. EVALUATION

This section presents results from our extensive evaluation of centralized and fully distributed models for data distribution. For this evaluation, we induce traffic delays in the road network to observe the relative performance of the different routing algorithms. Without traffic congestion, the nodes using a dynamic traffic routing algorithm would not have any congestion to avoid, and therefore would not be able to gain an advantage over the static traffic routing algorithm. We do this by simulating traffic events on a subset of road segments and artificially reducing the speed limit of the segment to about 2 miles per hour for 30 minutes. Such an event mimics the behavior of traffic in a construction zone or in the aftermath of an accident, causing significant congestion on and around that road segment. After a 30 minute warmup period without traffic events, we add traffic events so that approximately 10% of the road segments in the map are affected at any given time.

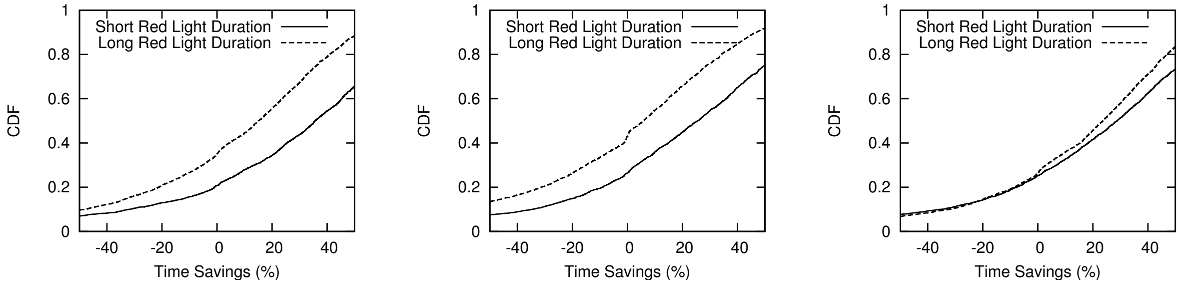
Given the impact of traffic-light duration in routing through an urban network, we first analyze the performance robustness of alternative traffic routing algorithms to it. Then, we discuss the performance of distributed and centralized data distribution services for traffic routing under varied node densities. Last, we analyze the sensitivity of both data distribution models to key configuration parameters: AP densities (centralized) and area of interest (distributed).

A. Red Light Duration and Traffic Routing

The various traffic routing algorithms respond differently to variations on red light duration. To understand the impact of varying red light durations, we configure each intersection in the simulation with an independently configured red light. To model low traffic light delays, we randomly assign intersections to have red light durations from 12 to 48 seconds. We selected the interval based on average travel time for a median length segment in our simulation - 24 seconds. The longest delay due to a red light (48 seconds) at most triples the travel time for that segment. To model higher traffic light delays, we use a base value of 96 seconds, giving red light durations from 48 to 192 seconds.

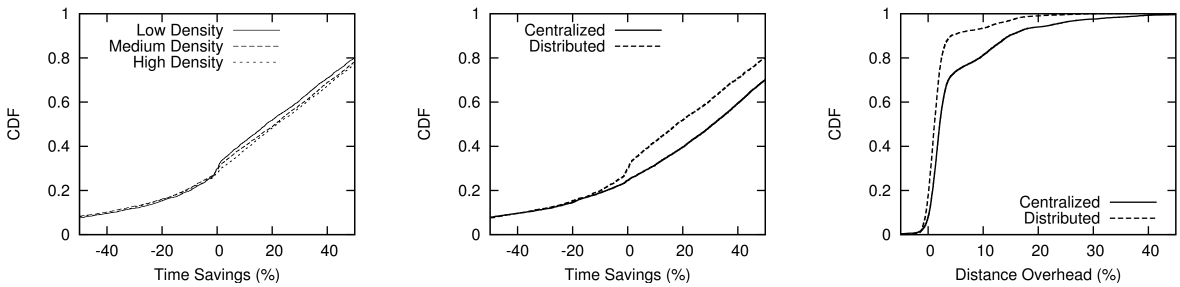
Figure 5 is a cumulative distribution function (CDF) of time savings for the three dynamic routing algorithms under short and long light durations. The time savings performance of the Temporal Speed and Spatial Speed routing algorithms are very sensitive to different red light durations, particularly when contrast with SSTE. The SSTE algorithm's time savings under long red light durations is no more than 10% lower compared to its time saving under short red light durations (Figure 5(c)). Temporal Speed (Figure 5(a)) and Spatial Speed (Figure 5(b)) routing show similar performance to SSTE with short red light durations – the rerouting nodes win the race about 80% of the time, and 25% and 35% of the races result in significantly large time savings of more than 50% over the naïve algorithm. However, under longer red light durations, both the Temporal and Spatial Speed routing algorithms only win about 60% of the races–20% fewer than the SSTE algorithm.

Given its robustness to the changes in traffic light duration, for the remainder of our discussion we use SSTE-based routing and limit red light durations to 12 to 48 seconds.



(a) Temporal Speed Routing (b) Spatial Speed Routing (c) SSTE Routing

Fig. 5. Centralized time savings of each of the three routing algorithms, under high AP density with a grid layout (see Section VI-C). SSTE routing gives lower variation in performance across variations in red light duration in comparison to Temporal and Spatial Speed routing.



(a) Time Savings: Distributed (b) Time Savings (c) Distance Overhead

Fig. 6. Impact on time savings as a result of varying the node density in the distributed system. Distributed and centralized time savings and distance overhead under low node density.

B. Distributed and Centralized: Varying Node Density

Node density, i.e. the percentage of instrumented vehicles in the system, is critical to the performance of a traffic advisory service – larger numbers of instrumented nodes result on more detailed knowledge of traffic conditions for both the centralized and distributed data dissemination models. In the distributed model, the density of participating nodes is also directly associated with the rate at which nodes meet and are able to exchange traffic information. We assess the application-layer performance of the centralized and distributed systems under varied node densities (24, 48, and 96 nodes/km^2).

We find that the time saving of the centralized data distribution model system is unaffected by changes in node density. A density of 24 nodes/km^2, the lowest of those evaluated, appears to be sufficient to generate a map of traffic congestion for efficient routing. The time savings performance for the centralized system under all node densities is identical to the centralized curve in Figure 6(b). However, as node density increases the routing algorithm becomes more "reactive" to changes on traffic conditions. This ultimately results on longer detours, with greater distance overhead (not shown) but no additional time savings. In the case of the distributed data dissemination model, on the other hand, varying node density does similarly increase distance overhead although this also comes with improved time savings (Figure 6(a)).

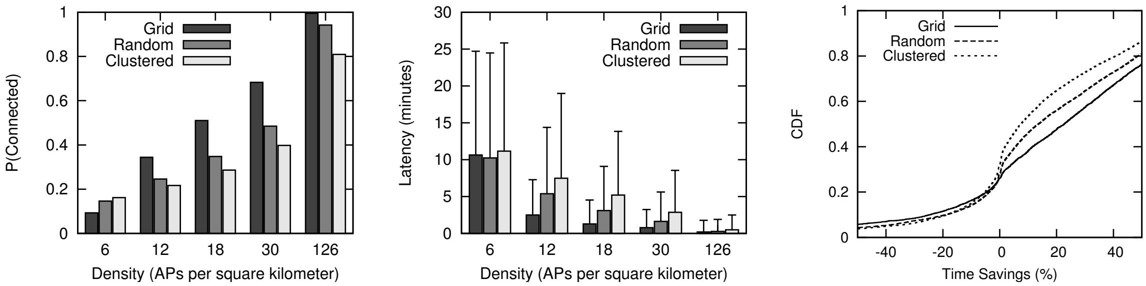
As Figures 6(b) and 6(c) show, both the centralized and distributed-model based routing algorithms results in faster routes in over 75% of the races under low node density. In addition, in 75% of the races both approaches achieve these time savings with relatively low distance overhead ($<10\%$) when compared with the static Travel Time algorithm.

At this density, the distributed and centralized approaches trade time savings and distance overhead. Since the centralized system's routing algorithm has "global" knowledge of traffic conditions, it finds slightly faster paths than the nodes in the distributed system (Figure 6(b)). As shown in Figure 6(c), the centralized system's "global" knowledge also results in larger detours around congested areas, which translates in 20% of its races having distance overhead greater than 10%. In comparison, distributed nodes only have distance overhead $>10\%$ in 5% of their races.

C. Centralized Performance: Varying Access Point Density

To assess the impact of less reliable connectivity on the performance of applications using centralized data dissemination, we study application-level performance under varying distribution and density of APs. It has been reported that APs are clustered and randomly distributed [38]. We therefore adopt three AP placement algorithms – uniform, random, and clustered – to evaluate the effect of increasing levels of unpredictable connectivity with the centralized server. Under uniform placement, APs are arranged in a rectangular grid. Random placement results in APs being placed randomly in the simulation map. In clustered placement, with some probability P (0.5) an access point is placed within a subregion of the simulation map (a 300 m square, 6% of the whole map), otherwise it is placed randomly.

We evaluate each AP placement algorithm with different AP



(a) P(Connected) (b) Measurement Latency (c) Placement Time Savings

Fig. 7. Effect of varying access point (AP) density on the centralized system. P(Connected) shows the probability that a node is connected to an AP. Measurement Latency displays the mean and standard deviation of the latency from when a measurement was recorded to the time that it was received at the centralized server. The Time Savings plots show the mean and standard deviation of percent time saved by rerouting in a race, relative to the naïve node. The Placement Time Savings plot (Figure 7(c), 12 APs/km^2) shows that AP placements resulting in lower measurement latency yield better time savings performance.

densities. We determined a typical urban access point density by searching the WiGLE wardriving database [39] for free or commercial access points (APs) in our simulation region. We found a total of 191 access points–174 commercial and 17 free–that had been detected during the time from January 2008 to October 2008, yielding about 130 access points per square kilometer. With the centralized signal propagation and radio model parameters described above, the base transmission radius is about 60 m. With 126 APs/km^2 placed in a grid, a centralized node is usually in range of at least one access point. To simulate areas with lower AP densities, we selected four additional AP densities as low as 6 APs/km^2.

Figures 7(a) and 7(b) shows the impact on physical-layer and application-independent metrics when varying the AP distribution and density. The P(Connected) metric represents the proportion of time that a node is in range of an access point; as AP density increases, so does P(Connected). Measurement Latency gives the time from when a measurement was collected until it was received at the centralized server. As density increases, latency decreases because nodes have to travel for a shorter time before encountering an AP.

Corresponding with the P(Connected) and Measurement Latency metrics, Figure 7(c) shows that higher P(Connected) and lower latency result in greater time savings performance. The grid placement algorithm, therefore, results in the highest time savings. Random placement has the second highest performance, while clustered placement yields the lowest time savings. For example, under the grid placement algorithm, about 25% of the rerouting race nodes have greater than 50% time savings compared to the naïve node. However, for the clustered algorithm, only 15% of the races are won with more than 50% time savings.

Figure 8 shows that there is an upper bound of time savings performance as AP density is increased. At low AP densities, increasing the density can significantly improve performance. For example, doubling the AP density from 6 APs/km^2 to 12 APs/km^2 results in nearly a 20% increase in time savings for more than a quarter of the races. However, at sufficiently high AP densities, adding more APs will not improve performance. A prime example of this is seen when increasing AP

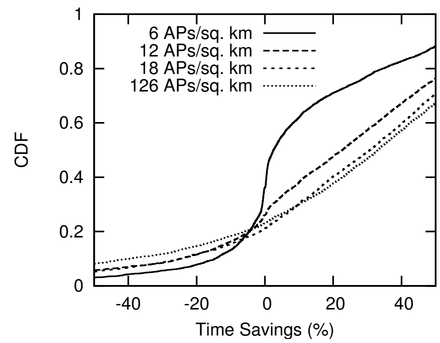
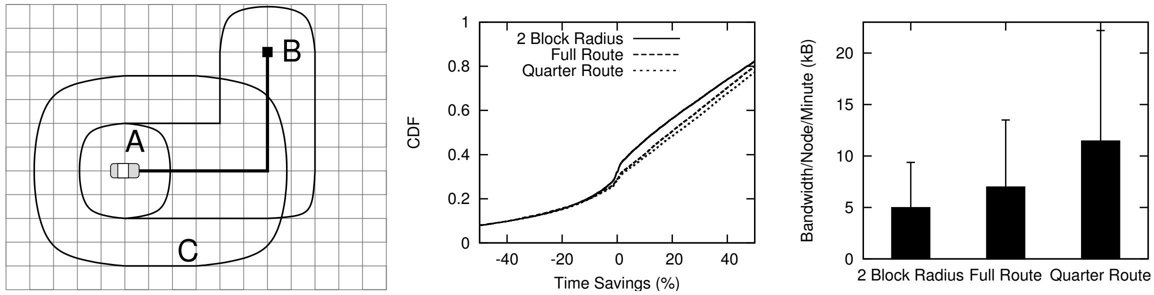


Fig. 8. Time Savings: Diminishing Returns on AP Density. The plot shows the upper bound of time savings performance as AP density increases with grid AP placement.

density by a factor of 7, from 18 APs/km^2 to 126 APs/km^2. In this case, the highest AP density resulted in less than 2% time savings for the majority of the races. The diminishing marginal returns on increasing AP density suggest that the connectivity provided by 18 APs/km^2–nodes connected 50% of the time, mean measurement latency of 5 minutes–is sufficient to provide near-upper-bound performance levels at one-seventh the AP density that exists in a dense urban area.

D. Distributed Performance: Varying Area of Interest

To evaluate the trade-off between data consistency and bandwidth consumption in our distributed traffic advisory system, we compare the application-level performance under three types of Area of Interest (AOI) messages: “2 Block Radius”, “Full Route” and “Quarter Route” (Figure 9(a)). The “2 Block Radius” (A) region restricts a node’s knowledge to traffic conditions just beyond the next intersection; this means that the routing algorithm is only able to make decisions based on local traffic conditions. The “Full Route” (B) region includes road segments along the node’s entire route. The intent of the Full Route AOI is to give the routing algorithm early warning of upcoming congested areas along the node’s route, so it can reroute ahead of time. The “Quarter Route” (C) region focuses on providing the routing algorithm with traffic information for a large number of alternative routes in



(a) Area of Interest Types (b) Time Savings (c) Bandwidth

Fig. 9. Area of Interest Types shows the sample regions covered by the three distributed system AOIs that we study. 2 Block Radius (A) is a circular region around the node's current location. Full Route (B) includes every road segment within 2 intersections of any point on the node's route (indicated with a heavy black line). Quarter Route (C) includes every road segment within 4 intersections of the first quarter of the node's route. The Quarter Route AOI requires the most bandwidth because its region encompasses the most road segments. The Time Savings plot shows performance increases with the size of the AOI while the Bandwidth plot shows the mean and standard deviation of KB/s received by each node. Requesting data about road segments along the node's route and increasing the breadth of the requested data both offer marginal increases in application layer performance at the cost of increased bandwidth consumption.

the short term; it includes road segments within 4 intersections away from the first quarter of the node's route.

Figure 9(b) shows that providing the routing algorithm with more short-term alternate paths yields the best performance. In general, larger AOI regions yielded better time savings performance. Specifically, 2 Block Radius had mean time savings of 9.8%. Full Route, with mean time savings of 12.8%, has better performance than 2 Block Radius. However, Quarter Route has the largest mean time savings: 14.4%. When comparing Quarter Route and Full Route AOI, it is clear that Full Route's routing algorithm cannot capitalize on traffic information along the furthest segments of the node's route – as their conditions may change to invalidate previous routing decision.

Not surprisingly, larger AOIs imply higher bandwidth demands. Figure 9(c) shows the required bandwidth for the these different AOIs. The Quarter Route AOI requires the most bandwidth, 10 kB/minute per node–twice as much as 2 Block Radius. The standard deviation of both Full Route's and Quarter Route's bandwidth are greater than 2 Block Radius because the size of the nodes' AOI requests vary with the length of the node's current route to its destination.

VII. Conclusion

We consider the problem of data dissemination in vehicular networks. Our goal was to explore the inherent trade-offs in the adoption of distributed or centralized approaches to a traffic advisory service, a commonly proposed application. We based our analysis on a measurements study of signal propagation in urban settings and an extensive simulation-based experimentation in the Chicago road network. We found that both approaches can provide comparable performance, even under challenging low adoption conditions. For instance, with a 7% penetration ratio (24 nodes/km^2) both systems are able to provide a faster route at least 70% of the time, while incurring less than a 10% increase in distance traveled 75% of the time.

References

[1] 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 *Proc. of ACM/IEEE MobiCom*, Montreal, Canada, September 2007.

[2] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson, "Virtual trip lines for distributed privacy-preserving traffic monitoring," in *Proc. of ACM/USENIX MobiSys*, Breckenridge, CO, June 2008.

[3] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: a distributed mobile sensor computing system," in *Proc. of ACM SenSys*, 2006.

[4] H. Zhu, Y. Zhu, M. Li, and L. M. Ni, "HERO: online real-time vehicle tracking in Shangai," in *Proc. of IEEE INFOCOM*, 2008.

[5] M. D. Dikaiakos, S. Iqbal, T. Nadeem, and L. Iftode, "VITP: an information transfer protocol for vehicular computing," in *Proc. of ACM VANET*. New York, NY, USA: ACM, 2005, pp. 30–39.

[6] J. Yoon, B. Noble, and M. Liu, "Surface street traffic estimation," in *Proc. of ACM/USENIX MobiSys*, 2007.

[7] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," in *Proc. of ACM/USENIX MobiSys*, 2008.

[8] "Dash Navigation, Inc." http://dash.net/.

[9] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "Trafficview: traffic data dissemination using car-to-car communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, no. 3, 2004.

[10] L. Wischhof, A. Ebner, H. Rohling, M. Rohling, M. Lott, and R. Halfmann, "SOTIS – a self-organizing traffic information systems," in *Proc. of IEEE VTC*, April 2003.

[11] United Nation Population Fund, "State of the world population 2007: Unleashing the potential of urban growth," *Report*, 2007, www.unpfa.org.

[12] M. DeBacker, "Chicago area transportation crisis," *Metropolitan Planning Council*, February 1999.

[13] D. Schrank and T. Lomax, "The 2007 urban mobility report," Texas Transportation Institute, The Texas A&M University System, College Station, TX, Tech. Rep., September 2007.

[14] B. Coifman, "Using dual loop speed traps to identify detector errors," *Transportation Research Record*, vol. 1683, no. 1, 1999.

[15] New York City DOT, "Real-time traffic cameras," 2008, http://www.nyc.gov/html/dot/html/motorist/atis.shtml.

[16] J. Herrera and A. Bayen, "Traffic flow reconstruction using mobile sensors and loop detector data," in *87th TRB Annual Meeting*, 2007.

[17] X. Dai, M. Ferman, and R. Roesser, "A simulation evaluation of a real-time traffic information system using probe vehicles," in *IEEE Intelligent Transportation Systems*, 2003.

[18] K. Bensinger, "Chrysler will offer wireless internet access in 2009 models," 25 June 2008, http://www.latimes.com/business/la-fi-wificar25-2008jun25,0,1676276.story.

[19] "NextBus Inc." http://www.nextbus.com.

[20] V. Taliwal, D. Jiang, H. Mangold, C. Chen, and R. Sengupta, "Empirical determination of channel characteristics for DSRC vehicle-to-vehicle communication," in *Proc. of ACM VANET*, 2004.

[21] Z. Wang and M. Hassan, "How much of dsrc is available for non-safety use?" in *Proc. of ACM VANET*, September 2008.

[22] A. Balasubramanian, Y. Zhou, W. B. Croft, B. N. Levine, and A. Venkataramani, "Web Search From a Bus," in *Proc. of ACM CHANTS*, 2007.

[23] R. Barr, "An efficient, unifying approach to simulation using virtual machines," Ph.D. dissertation, Cornell University, 2004.

[24] D. R. Choffnes and F. E. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks," in *Proc. of ACM VANET*, 2005.

[25] "U.S. Census Bureau - TIGER/LINE," http://www.census.gov/geo/www/tiger/.

[26] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *Proc. of ACM MSWIM*, 2004.

[27] M. Takai, J. Martin, and R. Bagrodia, "Effects of wireless physical layer modeling in mobile ad hoc networks," in *Proc. of ACM MobiHoc*, Long Beach, CA, October 2001.

[28] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *Proc. of ACM/USENIX MobiSys*, Boston, MA, June 2004.

[29] T. Rappaport, *Wireless communications: Principles and Practice*, 2nd ed. Prentice Hall, 2002.

[30] V. Naumov, R. Baumann, and T. Gross, "An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces," in *Proc. of ACM MobiHoc*, 2006.

[31] J. S. Otto, F. E. Bustamante, and R. A. Berry, "Down the block and around the corner – the impact of radio propagation on inter-vehicle wireless communication," in *Proc. of ICDCS*, 2009.

[32] "Soekris Engineering, Inc." http://www.soekris.com/.

[33] "Ubiquiti Networks, Inc." http://ubnt.com/.

[34] "Pacific Wireless," http://www.pacwireless.com/.

[35] "Garmin, Ltd." http://www.garmin.com/.

[36] "Iperf," http://dast.nlanr.net/Projects/Iperf/.

[37] "Tcpdump/libpcap," http://www.tcpdump.org/.

[38] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ Wi-Fi networks," in *Proc. of ACM/IEEE MobiCom*, 2006.

[39] "WiGLE - Wireless Geographic Logging Engine," http://www.wigle.net/.