RegReS: Adaptively Maintaining a Target Density of Regional Services in Opportunistic Vehicular Networks.

Emmanouil Koukoumidis[†], Li-Shiuan Peh[*], Margaret Martonosi[†]

[†]Princeton University
{ekoukoum, mrm}@princeton.edu

[*]MIT
peh@csail.mit.edu

***Abstract*—Pervasive vehicle-mounted mobile devices are increasingly common, and can be viewed as a large-scale ad hoc network on which collaborative, location-based services can be directly supported. In order to support such services within a geographic region, a certain number of computational, storage and sensing mobile devices need to be carriers of the services. This paper introduces and evaluates Region-Resident Services (RegReS), a middleware that supports such regional services by maintaining, in a fully distributed fashion, a targeted density of service carriers. Carriers collaborate opportunistically to estimate the current service density in the region and coordinate the spawning of new service carriers when necessary. Unlike previous approaches that are static, RegReS adapts to dynamic conditions such as node speed, effectively maintaining the targeted density of service carriers in highly volatile vehicular networks. Results from the ORBIT testbed, using synthetic and real bus mobility traces, show that RegReS adapts to different system configurations, preserving the desired service density with less than 16% mean absolute error. We deployed an outdoor collaborative parking availability service atop RegReS and demonstrated RegReS's ability to maintain the target service density with only 10% error.**

I. Introduction

Networked mobile computing devices are becoming increasingly pervasive. Powerful smartphones are nearly ubiquitous. Furthermore, soon a large percentage of vehicles will be equipped with advanced Personal Navigation Devices (PND) that will have not only motion sensors (GPS, accelerometer, gyroscope, compass) but also fast, short-range ad hoc communications (e.g., DSRC [1]).

These trends are prompting exciting location-based services in Intelligent Transportation Systems (ITS) applications that leverage local sensing. For instance, on-board mobile devices using their wireless interfaces, GPS, gyro and accelerometer information can help estimate traffic conditions [2], detect road abnormalities [3], collect information for available parking spots [4], [5], measure air or noise pollution [6].

In this paper, we explore Region-Resident Services (RegReS), the hosting of these location-based services directly on the mobile devices in the region of interest, with the devices collaborating to form a distributed computing platform, obviating the need for any server computing infrastructure in the cloud. Given the high node density, mobility and availability of free vehicle-to-vehicle communications, such a grassroots distributed computing platform is particularly suited for ITS applications, as highlighted previously in [4], [7]. However, in these previous works, every service is epidemically *pushed* and maintained on *all* nodes in the region of interest. This oblivious use of all nodes is wasteful given the limited computing, storage and above all communication resources of mobile devices. On the other hand, only letting nodes *pull* and run the service on-demand may lead to too few or too many service carriers; normally a disparity will exist between the number of nodes that want to consume a service and the number of nodes that are necessary to support it.

While there have been approaches that selectively choose nodes for sensing or disseminating regional information [8], [9], [10], [11], they do not sufficiently tackle the requirements and traits of ITS applications; vehicular networks introduce a challenging environment of variable node mobility and density. As our results show, the static mechanisms that these schemes use fail to perform well across different system configurations.

This prompts us to propose RegReS, where each service specifies its desired service carrier density (the number of mobile devices that should host this service within a region) along with its region of interest and lifetime. The RegReS middleware, which runs on the carriers, ensures that this target carrier density is maintained in a distributed fashion. RegReS uses a collaborative and *adaptive* estimation scheme to track and estimate the current carrier density for a service. RegReS then employs spawn policies and carrier selection criteria to decide when and which nodes to spawn as new carriers.

The contributions of this work are the following:

1) We identify the four traits that middleware supporting ITS services should possess, argue for service carrier density as the suitable metric for ITS services, and propose the first solution that accounts for all four traits using this metric.
2) We propose the first collaborative approach in Mobile Adhoc Networks (MANET) that *adapts* to environment parameters and maintains a targeted density of services, and we show its effectiveness on a testbed.
3) We demonstrate RegReS's potential by using it as the middleware for the deployment of a collaborative parking availability service.

Next, Section II surveys related work, and then in Section III RegReS's design is described in detail. Section IV describes our evaluation methodology, and Section V presents performance results. Section VI offers our conclusions.

II. Background and Related Work

In ITS applications a certain number of service carriers must be maintained in the region of interest for sensing, storage and computation. We term this service carrier density (d). We argue that each application service should choose and specify its own desired d in the targeted geographical region of deployment so as to be able to reflect its own cost-performance tradeoffs.

For instance, in an application that estimates average vehicle speed on roads in a region, sensing may be noisy, as a given vehicle that reports its speed may be moving faster or slower than most other vehicles. With a sufficient carrier density across the region of interest, a good enough sampling density can be attained, with outliers removed. On the other hand, having too many vehicles as service carriers leads to high cost in terms of computing, storage and communications overhead.

Similarly, for a parking availability service, higher density of service carriers leads to more frequent road scanning and more robust detection of free parking spots, mitigating noisy sensors such as ultrasonic sensors [5]. Yet, too many carriers lead to unnecessary consumption of regional computational resources.

While most ITS applications need to maintain such a desired regional sensing capacity based on their own cost-performance tradeoffs, there is often no intuitive definition of sensing range. For example, in the case of traffic estimation, what is the sensing range for a vehicle that is reporting its own vehicle speed measurement?

On the other hand, proposed ITS applications do not typically need hard guarantees on sensing coverage (k-coverage) as they are often not focusing on life critical applications (*e.g.,* intrusion detection in sensor networks). Opportunistic sensing is often both sufficient and the only approach possible given the uncontrolled vehicle density and mobility.

Therefore, we target service carrier density as the metric for ITS services. RegReS allows services to specify their desired carrier density (d) along with their region and lifetime. This carrier density determines the population of carriers that RegReS should seek to maintain within the service's region and for the specified service's lifetime.

A. ITS Traits

ITS applications possess four key characteristics, which make the maintenance of the target number of service carriers a challenging task that no prior works (Table I) sufficiently tackle:

1) *Uncontrolled and variable node mobility:* Proposed schemes should not assume that vehicles' movements can be orchestrated to better support services. They can only be used opportunistically. Proposed schemes should also adapt to variable node mobility.
2) *Uncontrolled and variable node density:* Vehicle density varies and can be especially high in cities, up to several hundreds or thousands in a $5km^2$ region [12]. More often than not, only a small fraction of the vehicular nodes is necessary to provide a given service. Schemes need to selectively use only as many nodes as needed.
3) *Significant service activation/replication (spawn) cost:* ITS services can sense and gather data rapidly, leading to a large amount of state that needs to be transferred whenever a new carrier is spawned. Furthermore, there is no control over the software each mobile device has, so signed code modules may need to be moved to new carriers as well. Spawns can hence be several tens of KB. Proposed schemes should thus retain a service carrier as long as possible (while it remains in the region), and minimize the spawning of new carriers.
4) *Challenging operating environments:* Proposed schemes need to be robust to node failures; nodes may crash, get powered off by their owners, or just exit the region of interest.

B. Service Replication Literature

Proposed grassroots approaches targeting ITS or other regional applications [4], [7], [13] have been largely based on schemes that epidemically **push** the service on *all* available nodes. However, as noted earlier, in dense urban environments using all nodes for a given service is both unnecessary and wasteful.

Other approaches, in contrast, do not proactively push the service to all regional nodes but have the nodes interested in the service epidemically **pull** it [13] or **subscribe** to receive it [22]. However, such approaches cannot guarantee that the critical number of service carriers will be available; often a disparity will exist between the number of service consumers/subscribers and the number of carriers that are necessary to support the service.

A third class of approaches tries to control the number of service carriers by instructing current carriers to epidemically spawn new carriers with a defined probability [16], [17], [18]. While such approaches are not as wasteful as approaches that use all nodes, our results show (Section V-A) that a fixed spawn probability may work well for a specific case, but fails to adapt across different configurations (node mobility, region size, *etc.*).

A fourth class of theoretical works [11], [19], [20], [21], [22] begin by creating the desired number of service replicas/tokens and assume that the services can be moved between nodes *without accounting for node failures*. Lastly, **k-coverage** literature in sensor networks assumes either that nodes are *static* [8] or that their mobility can be *controlled* [9], [10]. None of these assumptions hold in real-world vehicular environments.

C. Density Estimation Literature

In RegReS, service carriers track their density across the region so that they can make informed decisions about whether to spawn a new carrier or not. No prior art has used service density as the metric to guide service activation/replication. Furthermore, despite the importance of determining density in MANETs, little work has been

	service replication								estimation schemes		
	epidemic push	epidemic pull	probabilistic spawning	N initial replicas			k-coverage		centralized	distributed	
	[4], [7], [13], [14], [15]	[13]	[16], [17], [18]	[11], [19]	[20], [21]	[22]	[8]	[9], [10]	[23]	[7], [24], [25], [26]	RegReS
high mobility	√	√	√	X	√	√	X	√	√	√	√
uncontrolled mobility	√	√	√	X	√	√	X	X	√	√	√
adaptivity	X	X	X	X	X	X	X	X	X	X	√
spawn cost	X	√	√	X	X	√	X	X	N/A	N/A	√
fault tolerance	√	√	√	X	X	X	√	√	X	√	√

Table I: **Comparison table.**

done so far in estimating it. In [23], the authors propose a *centralized* node census approach, which is not suited for highly-distributed ITS environments.

Our collaborative distributed density estimation scheme draws from [24]. In contrast to [24], RegReS carriers do not exchange complete logs of raw measurements with their timestamps but only the estimates they themselves have built or received from other carriers together with the confidence value for these estimates. Such estimates may be based on multiple such measurements. Thus the amount of information exchanged for density estimation is significantly reduced in RegReS.

Collaborative schemes for estimating local density are also used in [7], [25], [26] to guide (request or vehicle) routing decisions. However, these schemes focus only on local density and limit the information exchange to only between direct neighbors within communications range and do not leverage opportunistic forwarding.

Above all, RegReS, unlike prior art, adapts its estimation scheme to node dynamics. As shown in Section V-A this adaptivity is of critical importance.

D. *RegReS for ITS*

Here, we outline how RegReS handles the four key traits of ITS applications, tackling a key gap that has not been addressed by prior research:

1) *Uncontrolled and variable node mobility:* RegReS uses nodes opportunistically and adapts its density estimation scheme to their mobility patterns (Section III-D).
2) *Uncontrolled and variable node density:* RegReS targets an application-specified density of regional nodes (the designated carriers) for maintaining the service.
3) *Significant service activation/replication (spawn) cost:* RegReS uses nodes as service carriers for as long as they remain in the region, as opposed to k-coverage schemes that use sleep-schedule-based activation [9], [10]. Through simulations we found that this can result in up to 4.9x reduction in the number of spawns for the experimental scenarios of this paper.
4) *Challenging operating environments:* RegReS is fully distributed and adaptively tracks and reacts to the current density of carriers. This makes it robust to carrier failures and departures from the region.

III. CARRIER DENSITY MAINTENANCE

To maintain the desired carrier density, RegReS uses a collaborative and adaptive distributed approach to *estimate* it across the region and react accordingly. It determines *when to spawn* new carriers and *how to select the carriers*[1]. RegReS is thus broken down into three subproblems that this paper tackles:

1. How to estimate carrier density: Carriers measure the density within their communication range periodically. What information should carriers exchange based on their measurements and how should it be used to calculate an estimate? Above all, how can this estimation scheme automatically adapt to system parameters (node mobility, region size, *etc.*)?

2. When to spawn: When can a carrier be sufficiently confident, given its estimate, that it needs to spawn a new carrier?

3. How to select carriers: Once a carrier decides to spawn a new carrier, which node should it pick?

Solutions to these three subproblems constitute the novel contributions of RegReS. Other peripheral functions, such as service discovery and updates are done epidemically.

A. *Carrier Density Estimation*

RegReS estimates carrier density in a collaborative fashion, through small metadata packets that nodes broadcast every P seconds. These Service Advertisement (SA) packets (Figure 1) contain information about the services (if any) nodes carry. They allow for: 1) service discovery, 2) service version updates, 3) discovery of potential carrier nodes if spawning is needed. RegReS leverages these packets to measure local density as well as allow carriers of the same service to exchange density estimations.

The carrier density d is expressed in units of number of carriers per πR^2 area, where R is the communication range:

$$d = \frac{N_{carriers}}{\alpha \times Area} \quad , \quad \text{where} \quad \alpha = \frac{1}{\pi R^2} \tag{1}$$

If $N_{carriers} = 100$ is the target population of carriers within the service's region, $Area = 2216m \times 2216m$ is the region size and $R = 250m$, then in RegReS, the targeted service carrier density is d=4 carriers per communication range.

1) Estimation Algorithm: Every P seconds[2], carriers measure the number of other carriers (of the same service) in their range, by listening to SA packets. Measurements are exponentially-weighted-averaged to form a density estimate that biases towards newer measurements. The factor $0 \leq f < 1$ by which the measurements' weights decay over time is called decay factor. The smaller f is, the faster measurements decay. If m_{-k} is the measurement done k

[1]The maintenance of the service is the responsibility of only the existing carriers. Non-carrier nodes are not involved and hence do not incur any overhead.

[2]Nodes are time-synchronized with a GPS device. They use a random backoff scheme to broadcast their SA packet within each period.

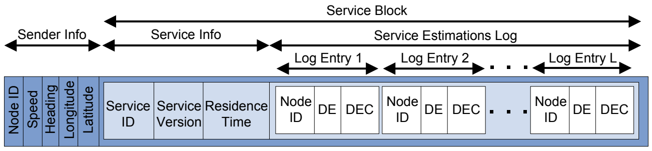


Figure 1: **Service Advertisement packet.** *Node IDs* **are unique identifiers of nodes.** <*Speed, Heading, Longitude, Latitude*> **determines the current position, average speed and direction that a node is moving towards, and are used in carrier selection to calculate ERRT.** *Service ID* **and** *Version* **uniquely identify each carried service.** *Residence Time* **tracks the number of periods that the carrier has resided within the region of this service and is used for determining the value of f. The** <*Node ID, DE, DEC*> **triplets are used for collaborative density estimation. One such** *Service Block* **is included in the SA packet for each service the node is carrying.**

periods ago, the current Density Estimate (DE) is calculated as follows:

$$DE = f \times DE_{old} + (1 - f) \times m_0 \quad (2)$$

The sum of the weights of these measurements converges to 1 and we term this Density Estimation Confidence (DEC). DEC is the **confidence** that a carrier has in its estimate. It grows over time as the carrier gathers more and more measurements:

$$DEC = \sum_{i=0}^{k} (1 - f) \times f^i \to 1 \quad (3)$$

RegReS adapts the value of the decay factor f to system parameters (node mobility, region size, *etc.*). The adaptation is based on a regression model described in Section III-D. This adaptation influences both the value of DE (Equation 2) and the rate with which carriers accumulate confidence (Equation 3) for their density estimates. As shown in Section V-A1, adaptation is critical in vehicular networks and offers RegReS improved performance over a wide range of system configurations.

Carriers use SA packets to exchange Density Estimations (DE). SAs include a list of triplets <Node ID, DE, DEC> that record along with the estimate (DE), its confidence (DEC) and also the ID of the carrier node that had generated it (to detect and discard duplicates). Carriers for each service maintain and exchange a log of size L of such entries. This exchange helps carriers populate their logs with triplets from other carriers and forms the basis of the collaborative density estimation scheme. A carrier uses the information in this log (that includes its own estimate too) to build a more accurate estimate by weighting the estimations (DE) using their confidences (DEC):

$$DE_{merged} = \frac{\sum_{i=1}^{L} DE_i \times DEC_i}{\sum_{i=1}^{L} DEC_i} \quad (4)$$

As new triplets are received, only the L most confident ones are preserved. Log entries are decayed at the end of each period, by multiplying their DEC by the decay factor f. In this way their effect in the averaging operation (Equation 4) is also decayed to reflect their increasing staleness.

B. Spawn Policies: When to spawn?

The service carriers are mobile and stay within the service's region only for a limited amount of time. In order to maintain the desired carrier density, new carriers need to be spawned over time to replace carriers that exit. We propose and investigate three alternative spawn policies:

Policy 1 (P1): Spawn if $m_0 < d$. A carrier will spawn a new carrier whenever the measurement it made over the last period indicates that the existing carrier density is lower than the target value. Since carriers are highly mobile, their spatial distribution changes all the time and several transient carrier clusters and dispersals are created across the region. As a result this spontaneous policy ends up overspawning carriers (Section V-B1).

Policy 2 (P2): Spawn if $DE_{merged} < d$. A carrier bases its spawn decision on the density estimate it builds over time and not solely on the last measurement.

Policy 3 (P3): Spawn if $DE_{merged} < d$ *and* $DEC \geq C_{thres}$. The very first estimations that a carrier makes are not that accurate as they are based on a limited number of measurements and exchanges (if any) with other carriers. It takes time for a carrier to build a more accurate and confident density estimation. Therefore, a further optimization enforces a confidence threshold (C_{thres}). A carrier will spawn only if its density estimation confidence (as defined in Equation 3) exceeds the C_{thres} threshold.

The confidence of a carrier's merged density estimation (DE_{merged}) is defined as in Equation 3 and grows as the carrier spends more and more time in the region. Confidence grows sublinearly though as old measurements are not as indicative as new ones. The rate at which confidence increases depends also on the value of the decay factor f.

Spawn packets are unicast UDP packets that contain the service ID, version, region, lifetime and data. The service data consists of service state, as well as signed code modules should the newly spawned carrier not have the necessary modules to run the service.

C. Carrier Selection Criteria: On whom to spawn?

Given the high speed of vehicles, the number of distinct carriers needed to preserve the desired density can be in the order of several thousands in our experimental scenarios. Spawns can be reduced by accounting for node mobility and selecting as carriers only nodes that will be staying within the region for more than a threshold amount of time.

The Estimated Residual Residence Time (**ERRT**) of a node *i.e.,* the estimated amount of time left for which the node will be remaining within the region of the specific service, depends on its current location, heading and speed. Carriers calculate the ERRT of encountered nodes using the <*Longitude, Latitude, Speed, Heading*> information of the SA packets (Figure 1) they receive. At the same time carriers also calculate the mean of these ERRTs across nodes; this is termed Nodes' Mean Residual Residence Time (**NMRRT**).

Our carrier selection criteria select as carriers only nodes whose ERRT is greater than NMRTT by some factor. Different carrier selection criteria may have varying effects on the uniformness of the carrier distribution across the region and we investigate this in Section V-B2.

D. Decay Factor Adaptation Model

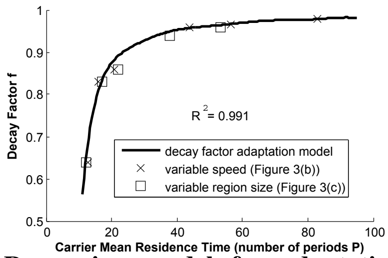


Figure 2: **Regression model for adaptation of decay factor f to different system configurations.**

From Equations 2-4, it is clear that the decay factor f greatly affects the performance of RegReS. More specifically, a small decay factor yields density estimates that are highly biased towards recent measurements while larger decay factors give more emphasis to older measurements. Intuitively, when the rate of change of carrier density (due to carrier exits and new carrier spawns) is high, newer measurements are a lot more indicative compared to older ones and thus the decay factor should be smaller. Conversely, when the rate of change of carrier density is low, older measurements reflect the current density of carriers almost as well as the most recent ones, and the decay factor should be larger.

Vehicular networks are highly dynamic and hence a model is needed for automatically adapting the decay factor to the rate of change of carrier density. As our results show (Section V-A), an accurate carrier density can be maintained only if the decay factor adapts to node mobility or region size changes[3]. Intuitively, these two factors determine the rate of density changes as the faster nodes move or the smaller the region size the faster existing carriers exit the region and new carriers need to get spawned. Conversely, for small speeds (or big regions) the rate of carrier density changes is lower.

As the basis of our adaptation model, and in order to track the the rate of change of carrier density, we choose the Carrier Mean Residence Time (**CMRT**) metric. The CMRT is the average amount of time that carriers reside within the region and we use it derive a regression-based model for f. Carriers estimate and update the value of CMRT using the information in the SA packets received from other carriers; The carrier (past) Residence Time (Figure 1) is added to the calculated (future) ERRT time to estimate the total residence time for each encountered carrier. The total residence times of the carriers are then averaged using a simple arithmetic mean to form CMRT. Armed with CMRT, nodes then use the regression model to select the decay factor f they should use for DE and DEC calculations.

[3]We found that other system parameters like regional node population (Figure 3(a)) or target density of carriers (Figure 3(d)) do not affect the performance of the estimation scheme.

In Figure 2, we plot the values of the best[4] decay factor against CMRT for the configurations of variable speed of Figure 3(b) and the configurations of variable region size of Figure 3(c). Figure 2 also shows the regression-based approximation model that RegReS uses to adapt its density estimation scheme as a function of CMRT. The adaptation model is based on 9-nth order polynomial regression. We found that higher orders or non-polynomial kernels do not improve the approximation accuracy significantly. Therefore, using the approximation of Figure 2, the best value for the decay factor can be determined for arbitrary configurations by only knowing the CMRT.

While our regression model is developed based on Random Waypoint (RWP) mobility model, our results in Section V-A2 show that the model is general and works effectively with real bus traffic traces as well.

IV. Methodology

A. Testbed

Region size	$2216m \times 2216m$
Regional node population	N=200 nodes
Decay factor	f=0.86
Mobility model	RWP: speeds in [5, 15] m/s
Spawn policy	P3, C_{thres}=0.6
Carrier selection criterion	random
Estimation log size	L=4
Radio range	R=250m
Target carrier density	d=4
SA packets period	P=10sec
Experiment duration	1 hour (360 periods)
PAS service (Section V-C) spawn size	20KB

Table II: **Default experiment parameters. For region sizes R$\geq$3133m we use all available ORBIT nodes (N=350) and set d=1. This decay factor value (f=0.86) yields the most accurate density estimations *i.e.,* minimizes the density mean estimation error for the default configuration. The value of C_{thres} was determined empirically to minimize density mean absolute error for P3. Further increasing the size of the log does not yield significant benefits. Density estimation results for different log sizes or decay factors are not shown in the interest of space.**

For the evaluation of RegReS we prototyped a real system on the ORBIT[5] radio grid testbed [27] that provides a facility of 400 wireless Debian 4.1 nodes. For our experiments we used up to 350 of these nodes configuring their Atheros AR5002X Mini PCI cards in 802.11a ad-hoc demo mode. The radio range was 250m and mobility was emulated by filtering out packets from nodes whose virtual distance was greater than this. Table II shows the default parameter values. Packet loss rate was in the range of 1-4%.

[4]Ideally the carrier density should always equal the target value. Thus, the objective function that RegRes seeks to minimize is the density mean absolute error calculated as the absolute differences between the actual density of carriers and the target density at each period, averaged over the duration of the experiments (360 periods). The best decay factor is the one that minimizes this error.

[5]The ORBIT tested offers improved evaluation credibility compared to standard simulation environments [27].

B. Mobility Models

We used two mobility models:

1. *Random Waypoint (RWP):* The entry point of the nodes into the region is uniformly at random chosen on the border of the region. RWP then determines the travel path of the nodes. Node speeds are uniformly distributed in the range of 5m/s to 15m/s to emulate vehicular traffic. When a node crosses the regions' border, the node is considered to have exited and thus removes the services (if any) it is carrying. Given that we do not allow extremely low vehicular speeds (0m/s to 5m/s), and that nodes exit the region when they hit its boundary, as opposed to getting deflected back in, RWP reaches steady state within the first 50 periods.

2. *City Buses Traces (CBT)[28]:* Bus traces from a $2216m \times 2216m$ region exactly north of University of Washington were used. These traces capture only buses, so to better approximate the complete vehicular city traffic we created a higher vehicular density scenario by compressing traces from different hours of the day into a single one-hour-long trace. Traces of buses for different hours of the day were treated as traces of different buses moving in the single hour of the experiment.

V. Evaluation

A. Achieving Target Density

We evaluate how well RegReS maintains the target density of carriers against several baselines:

• *Prob:* This is a standard probabilistic spawning scheme [16], [17], [18]. The value of the spawn probability (p=0.0016) was empirically optimized for the default configuration (Table II).

• *Static Decay Factor (SDF):* As opposed to RegReS, the decay factor of this collaborative density-estimation-based scheme is static (does not adapt) and its value (f=0.86) was empirically optimized for the default configuration.

• *Best Decay Factor (BDF):* The best decay factor for each configuration is determined empirically and used.

• *Oracle:* This is a hypothetical scheme. Every P=10 sec, the oracle, knowing exactly how many carriers are missing, makes the necessary spawns.

1) RWP Mobility: In order to provide a thorough evaluation and at the same time show the importance of adaptation, we compare the performance[4] of these schemes across different system configurations by varying one parameter at a time: 1) regional node population (Figure 3(a)), 2) node speed (Figure 3(b)), 3) region size (Figure 3(c)), 4) target density of carriers (Figure 3(d)).

As shown in Figures 3(a) - 3(d), RegReS significantly outperforms Prob and SDF schemes in most scenarios. RegReS is able to adapt to varying system parameters maintaining the targeted carrier density with less than 16% density mean absolute error and 9% mean raw error (not shown in the interest of space). These errors are higher compared to those for the theoretical oracle scheme; carriers in RegReS make spawn decisions based on the limited information they collaboratively measure and exchange, without having global knowledge for the regional density.

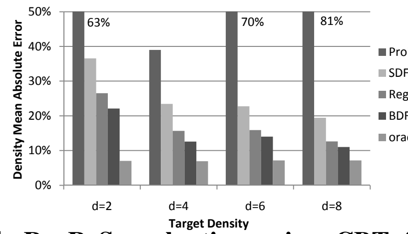


Figure 4: **RegReS evaluation using CBT for different target carrier densities.**

Still collaboration helps keep the errors small enough for ITS applications not needing hard guarantees. Furthermore, the performance of RegReS is very close (within 3%) to that of BDF across all configurations. For the configurations of Figures 3(a) and 3(d), SDF happens to use the best decay factor and hence matches the performance of BDF; the value of the best decay factor is only affected by node speed and region size as only these parameters influence the rate of carrier density changes. Prob is susceptible to the change of all parameters.

This analysis suggests that static schemes like Prob and SDF are very weak at maintaining a target service capacity and RegReS's adaptivity is critical. An adaptive estimation scheme like that proposed by RegReS should be used in dynamic vehicular environments whether the ultimate goal is request routing, power management or service replication.

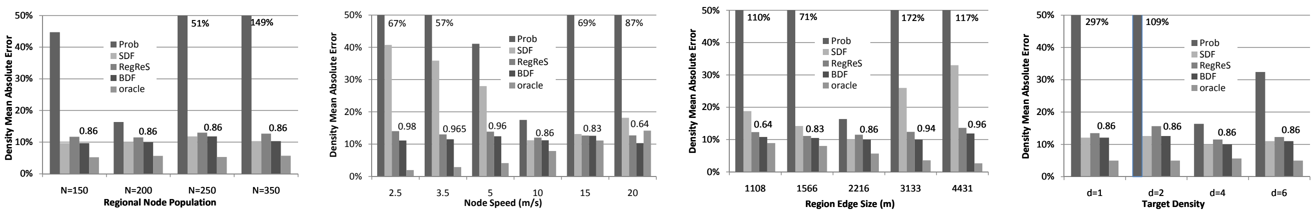
2) CBT Mobility: The density mean absolute errors [4], for CBT mobility and across different target carrier densities, are shown in Figure 4. The desired carrier density for most ITS applications is expected to be d$\geq$4. For such densities, RegReS can maintain the desired density of carriers with less than 16% mean absolute error and 10% mean raw error (not shown in the interest of space). These errors are within 3% of those for BDF.

The schemes that are based on collaborative density estimation (SDF, BDF, RegReS) do not perform as well when very low target carrier densities are combined with the CBT mobility. When node movement is highly correlated (CBT mobility), density estimation-based schemes need a high enough density of collaborating carriers to be able to build an accurate density estimate and robustly sustain clustered carrier exits. In contrast with RWP model even a density of d=0.5 can be sustained.

The bus traces that we used constitute one of the hardest possible cases for RegReS. In a scenario where all types of vehicles are used and many more streets are traversed, vehicles will be mingling better. As a result, the performance of RegReS for CBT will be closer to that for RWP.

B. RegReS Design Space Exploration

1) Spawn policies: In this section we evaluate how well the three spawn policies can maintain the target density. Figure 5 shows the density mean raw and absolute errors. The best spawn policy is P3 with a density mean absolute error of 10%. The mean raw error is even lower (8%). As discussed in Section III-B, spawn policy P1 is highly spontaneous and thus ends up overspawning, leading to a



(a) (b) (c) (d)

Figure 3: **RegReS evaluation using RWP for different** (a) **regional node populations,** (b) **node speeds,** (c) **region sizes,** (d) **target carrier densities. Values of decay factor for BDF shown on top of each bar. Note that some Prob errors exceed the maximum value (50%) of the y-axis. Carrier density fluctuated over time for all evaluated schemes. More specifically for RegReS, the standard deviation of the carrier density is within 14% of the target carrier density.**

density of carriers significantly higher than the target one (64% higher). Policy P2 is not as spontaneous as P1 but still not as measured in its spawn decisions as P3, as it spawns regardless of confidence.

2) Carrier Selection Criteria: In this section the benefits and drawbacks of three carrier selection criteria are evaluated: C1: $ERRT \geqslant NMRRT$, C2: $ERRT \geqslant 1.5 \times NMRRT$, C3: $ERRT \geqslant 2 \times NMRRT$. The number of spawns for these three criteria and the random carrier selection baseline for different node speed scenarios is shown in Figure 6(a). The top of each bar shows the factor by which the number of spawns is decreased compared to the random selection baseline of the same speed.

Figure 6(a) shows that the number of spawns can be greatly reduced by enforcing such smarter carrier selection criteria. C1, C2 and C3 reduce the number of spawns by factors of 1.3 to 1.4, 1.5 to 1.7 and 2.3 to 2.9, respectively, depending on node speed. Furthermore, Figure 6(b) shows that the more relaxed carrier selection criteria C1 and C2 do not hurt the carrier density maintenance as opposed to C3. C3 imposes strict constraints that nodes need to satisfy to become carriers. Therefore, existing carriers are having a hard time finding eligible nodes and end up underspawning. The only exception is the case where speeds are not the same across nodes but uniformly distributed in the range of 5m/s - 15m/s. In this case there is more diversity among nodes and thus carriers can easily find eligible nodes to spawn.

The smarter carrier selection criteria C1, C2 and C3 may hurt the uniformness of the carrier distribution. To evaluate that, we use the density mean absolute spatial error metric. This is calculated by taking at each period a grid of 110×110 sampling points across the region and calculating for each gridpoint the absolute difference between the actual carrier density and the target one. Figure 6(c) plots these absolute differences averaged across all gridpoints and all periods of the experiment.

According to Figure 6(c), C1 increases the density mean absolute spatial error at most by 3% making it a very promising criterion to get savings without significant carrier distribution degradation. For C2 this error can be at most 10% and C3 may make this error even double compared to the random selection baseline.

The stricter criteria select nodes that stay longer in the region. These tend to be nodes that traverse the region mostly diagonally and/or pass close to the center. Therefore, the stricter carrier selection criteria tend to accumulate more carriers towards the center of the region and less close to the edges making the distribution of carriers less uniform.

C. Sample Application: Parking Service Deployment

Previously proposed smart parking applications depend on the existence of smart parking meter infrastructure [4] or other special onboard sensors [5]. To illustrate the potential of our grassroots platform, we developed a Parking Availability Service (PAS) that does not require any additional infrastructure or hardware beyond a GPS device and vehicle-to-vehicle communications. It should be noted that unlike [5], our PAS is a very small-scale single-street deployment.

When a vehicle moves out of a parking spot, it broadcasts the release of the specific spot with a <Latitude, Longitude, RT> triplet. The latitude and longitude constitute the geographic location of the released spot, and RT is the Release Timestamp *i.e.,* the time that the vehicle released the spot. The release of the parking spot is detected as a combination of the vehicle engine switching on and the vehicle gaining a speed of over 5km/h. The former is detected with the use of a power inverter and the latter with speed information from the GPS receiver. This by no means constitutes a robust parking release detection scheme and needs further refinement.

PAS service carriers maintain a list of such <Latitude, Longitude, RT> triplet entries. These entries are obtained either from vehicles while releasing parking spots or from other PAS service carriers via the epidemic service updates mechanism. An upper limit N is set on the number of entries and the most recent entries (based on time elapsed since the parking spot was released) are kept.

To demonstrate the PAS service and the ability of RegReS to maintain such a region-resident service we carried out a five-node deployment using Ubuntu 8.04 laptops (four Dell Latitude D610 and one IBM Thinkpad x40) equipped with Globalsat BU-353 GPS receivers and 802.11g interfaces. Three of the laptops were carried by humans and the other two were mounted on vehicles and powered by Jensen JP30 inverters via the cigarette lighter. The region was defined to be a 200 meter-long road segment and participants were asked to move freely in and out.

The service was maintained for 20 minutes using on average 2.7 carriers when the specified target density was set

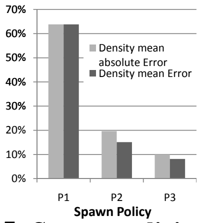


Figure 5: **Spawn policies evaluation.**

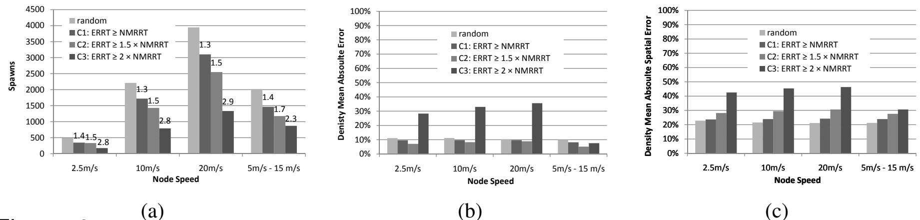


Figure 6: **Carrier selection criteria evaluation: (a) Number of spawns, (b) density mean absolute error, (c) density mean absolute spatial error.**

to three carriers within this region. To maintain this density RegReS performed 53 spawns in total as a result of carrier exits from our small deployment area; carriers removed the service after exiting. A parking release event was also triggered after the first five minutes of the experiment. The information about the released parking spot was received by the other two carriers that were in the region at that point in time and maintained as part of the service.

VI. Conclusions

This paper has presented RegReS, a middleware system that allows services to specify their desired carrier density along with their region and lifetime. RegReS then maintains this targeted regional carrier density throughout the lifetime of the service in a fully distributed fashion; service carriers opportunistically collaborate to estimate the current service density and spawn additional carriers where necessary directed by a confidence-based policy. RegReS adapts dynamically to different system parameters, effectively maintaining the required service density across a wide range of environments with less than 10% mean raw error and 16% mean absolute error. Such a regional service middleware can form an important foundation for low-infrastructure ITS applications in particular, and mobile ad hoc networks in general.

References

[1] *IEEE 1609: Family of Standards for Wireless Access in Vehicular Environments (WAVE)*, U.S. Department of Transportation, Jan. 9 2006.

[2] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: using mobile smartphones for rich monitoring of road and traffic conditions," in *SenSys*, 2008.

[3] 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 *MobiSys*, 2008.

[4] M. Caliskan, D. Graupner, and M. Mauve, "Decentralized discovery of free parking places," in *VANET*, 2006.

[5] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "Parknet: drive-by sensing of road-side parking statistics," in *MobiSys '10*.

[6] R. K. Rana, C. T. Chou, S. S. Kanhere *et al.*, "Ear-phone: an end-to-end participatory urban noise mapping system," in *IPSN*, 2010.

[7] U. Lee, E. Magistretti, M. Gerla, Bellavista, and A. Corradi, "Dissemination and harvesting of urban data using vehicular sensor platforms," in *IEEE Transaction on Vehicular Technology*, 2008.

[8] G. S. Kasbekar, Y. Bejerano, and S. Sarkar, "Lifetime and coverage guarantees through distributed coordinate-free sensor activation," in *MobiCom*, 2009.

[9] W. W. V. Srinivasan and K.-C. Chua, "Trade-offs between mobility and density for coverage in wireless sensor networks," in *MobiCom*, 2007.

[10] N. Bisnik, A. Abouzeid, and V. Isler, "Stochastic event capture using mobile sensors subject to a quality metric," in *MobiCom*, 2006.

[11] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, and A. Bestavros, "Distributed placement of service facilities in large-scale networks," in *INFOCOM*, 2007.

[12] P. Reynolds, J. Von Behren, R. B. Gunier *et al.*, *Cancer Causes Control*. Springer Netherlands, 2002, ch. Traffic patterns and childhood cancer incidence rates in California, United States.

[13] C. Maihöfer, T. Leinmüller, and E. Schoch, "Abiding geocast: time–stable geocast for ad hoc networks," in *VANET*, 2005.

[14] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *NSDI*, 2004.

[15] T. Liu and M. Martonosi, "Impala: a middleware system for managing autonomic, parallel sensor systems," in *PPoPP '03*.

[16] K. Lee, S.-H. Lee, R. Cheung, U. Lee, and M. Gerla, "First experience with CarTorrent in a real vehicular ad hoc network testbed," in *MOVE*, 2007.

[17] P. Zhou, T. Nadeem, P. Kang, C. Borcea, and L. Iftode, "Ezcab: A cab booking application using short-range wireless communication," in *PERCOM '05*.

[18] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Sanadidi, "Co-operative downloading in vehicular ad-hoc wireless networks," in *WONS '05*.

[19] Y. Kim, V. Lesser, D. Ganesan, and R. Sitaraman, "Cluster-swap: A distributed k-median algorithm for sensor networks," in *WI-IAT*, 2009.

[20] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *WDTN '05*.

[21] P. Bellavista, A. Corradi, and E. Magistretti, "Redman: A decentralized middleware solution for cooperative replication in dense manets," in *PerComW '05*.

[22] I. Leontiadis, P. Costa, and C. Mascolo, "Persistent content-based information dissemination in hybrid vehicular networks," in *PERCOM '09*.

[23] M. H. Raza, L. Hughes, and I. Raza, *Trends in Intelligent Systems and Computer Engineering*. Springer US, 2008, ch. Density: A Context Parameter of Ad Hoc Networks.

[24] P. Zhang and M. Martonosi, "CA-TSL: Energy adaptation for targeted system lifetime in sparse mobile ad-hoc networks," *IEEE Transactions on Mobile Computing*, 2010.

[25] M. Fiore, C. Casetti, and C.-F. Chiasserini, "Efficient retrieval of user contents in MANETs," in *INFOCOM*, 2007.

[26] R. Friedman and N. Mori, "3DLS: Density-driven data location service for mobile ad-hoc networks," in *MobiHoc*, 2009.

[27] D. Raychaudhuri, I. Seskar, M. Ott *et al.*, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," 2005.

[28] J. G. Jetcheva, Y.-C. Hu, S. PalChaudhuri, A. K. Saha, and D. B. Johnson, "CRAWDAD data set rice/ad-hoc-city (v. 2003-09-11)."