# On the Sensitivity of Geo-based Content Sharing to Location Errors

Jörg Ott, Ljubica Kärkkäinen
Technische Universität München

Ermias Andargie Walelgne
Aalto University

Ari Keränen
Ericsson

Esa Hyytiä
University of Iceland

Jussi Kangasharju
University of Helsinki

***Abstract*—A number of opportunistic content sharing services were developed that exploit device-to-device contacts for infrastructure-less operation. All of them depend, like geo-based ad-hoc routing protocols, on mobile devices knowing their respective positions to accurately perform data replication. In this paper, we explore the impact of different types of location errors on the performance of such a service. We use a GPS error distribution for mobiles derived from real-world measurements, consider different frequencies for GPS readings, and account for only subsets of mobile devices actively using GPS. We carry out extensive simulation studies using synthetic mobility models as well as real-world traces to assess the impact of different types of errors. We find that, overall, opportunistic content sharing is quite robust provided that a sufficient number of nodes support GPS and allow the others to have a rough estimate of where they are. Whether or not the GPS position is prone to errors affects some scenarios and is almost negligible in others.**

## I. Introduction

A number of geo-based content sharing systems have been proposed in the past, including Floating Content [1], Hovering Information [2], Locus [3], and Ad Loc [4], among others. While details differ, by and large these systems share the conceptual idea of *anchoring* a piece of content in a physical location and making it available to other nodes within a maximum distance from this location; we jointly refer to these parameters comprising the origin and the replication (and availability radius) as the *anchor zone* [1]. A key property of all the above systems is that they do not rely on infrastructure nodes or cloud services to ensure data availability but rather replicate content items within the anchor zone among mobile nodes in a device-to-device (peer-to-peer) fashion. While this operation does not require infrastructure network access—and thus limits dependencies as well as vulnerability to third party actions such as traceability or censorship—it comes at the cost of unpredictability: there is no guarantee that content "posted" to an anchor zone will remain available. We refer to this property as best-effort (probabilistic) content sharing.

Obviously, the operation of such a system depends on the number of nodes that move through the anchor zone and are thus available for replication (for which we can state lower [5] and practical bounds [6]). But, in addition, it is crucial that the nodes in the system know where they are in order to be able to execute location-based content replication in a fully distributed system in the first place.

In principle, obtaining location information is largely trivial on today's smart mobile devices by using the Global Positioning System (GPS) or Assisted GPS. In practice, however, running GPS all the time consumes a lot of energy, noticeably reducing battery lifetime. As a result, not all mobile nodes may have GPS enabled all the time, or they may activate GPS only in certain intervals. Moreover, depending on the environment, reported GPS positions may be prone to position errors.

In this paper, we explore the impact of position errors on the performance of geo-based information sharing, using *Floating Content* as a reference service. We review related work in section II, briefly summarize the operation of Floating Content in section III, and then introduce our error model and error modes in section IV. We present the evaluation setup in section V and our results in section VI before we conclude in section VII with a brief assessment and hint at future work.

## II. Related Work

Geo-based routing protocols such as GeoCast [7], LAR [8], DREAM [9], and GPSR [10] were proposed based on the assumption that each node knows its location from GPS. However, the GPS signal could be obstructed in many places including inside buildings, tunnels, underground spaces, or wherever else the satellite signal is blocked [11]. In addition, due to the usually limited energy and computing power in mobile devices, it may not be practical to keep the GPS receivers of all nodes active all the time. In this section, we review some of the relevant GPS error modeling techniques and location-based routing protocols for ad-hoc networks.

Several geo-routing protocols have been proposed to address GPS-related errors and to estimate the position of the node of interest in a mobile ad-hoc networks (MANET). There are surveys [11], [12] that summarize, compare, and contrast early efforts on position-based routing in MANETs. Grid-based on-road localization (GOT) [13] was developed to improve location accuracy calculation by allowing message (beacon) exchanges between cars in self-organized vehicular networks, where some of the vehicles have inaccurate or blocked GPS signal. In this system the position accuracy depends highly on the number of vehicles with GPS signal and signal accepting threshold. Similarly, [14] presented a location estimation method for vehicles without GPS using a minimum of three GPS-equipped vehicles as reference points. This latter requirement results in the system suffering from inaccuracy of location estimation in low traffic densities. Another geo-based routing protocol [15] proposes that only a subset of nodes have GPS support (G-nodes). In contrast to the above, only a single

ISBN 978-3-901882-88-3 

G-node is required to allow a node without GPS (S-node) to infer its position from a nearby G-node.

The negative impact of GPS error in geo-based ad-hoc routing in VANETs is analyzed in [16]. The authors show that positioning errors can degrade the performance of geo-based routing by increasing the failure probability of route discovery and the hop-count. [17] studies the effect of GPS error using distance degradation of each hop to find that the performance of VANET is significantly affected by GPS errors.

A modular positioning system that uses various positioning sources (GNSS, GSM, and Wi-Fi), based on the quality of the measured signal is proposed in [18]. If GPS signal is not available or has low quality, Wi-Fi or GSM measurements are read by the mobile device and sent to a localization server. The localization server is responsible for estimating the mobile device's position. GPS error modeling in [19] focuses on localization accuracy by creating a collaboration among mobile nodes that are capable of localizing themselves using GPS or pedestrian dead reckoning. A simulation study in [19] uses a simple random mobility model and shows that the magnitude of GPS error can range up to several tens of meters.

In our work, we focus on geo-based content sharing where content items are kept at rather than routed to a location, and we use extensive measurements for our GPS error model.

## III. Floating Content Operation

We base our work and terminology on the Floating Content model [6]. We assume smart mobile devices equipped with short-range radios such as Bluetooth or WLAN plus suitable communication stacks that allow them to detect peers in their proximity, set up a connection to, and exchange information with them. Despite limitations of mobile operating systems, solutions such as WLAN-Opp [20] can offer connectivity between peers and opportunistic networking platforms such as Haggle [21], PodNet [22], SCAMPI [23], or Twimight [24] implemented the content replication services on top.

The Floating Content model defines for each content item $I$ an anchor zone comprising its origin (or: anchor point) $P$, a replication range $r$, and an availability range $a$ (cf. Figure 1a). Within radius $r$ from $P$, the item is replicated by a node when it meets another node. Between $r$ and $a$, a node holding I will be passive, i.e., continue carrying the item but not replicating it anymore. When a holder is further away from $P$ than $a$ item I is deleted (upon encountering another node).[1]

Each item $I$ has a size $s_I$ and an associated time-to-live (TTL) $T_I$; after its lifetime expires, it will automatically be deleted ("garbage collection"). Together with the availability range $a$, these two properties define the potential total resource usage in the distributed system: all nodes within $a_I$ from $P_I$ may hold a copy of $I$ for $T_I$ seconds and use up $s_I$ memory during this period. We have used this total resource usage to prioritize content items for replication: those with the least demands would be replicated first. This policy ("STF2" in [6])

[1]Experiments with different replication and deletion policies reveal that the simple one here yields very good results and is practical to implement. [6]

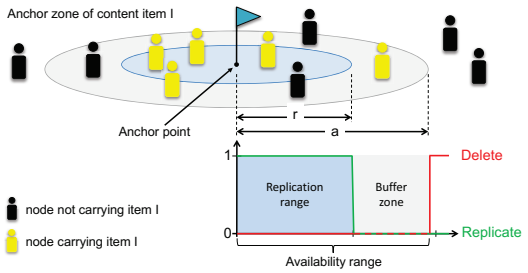



Fig. 1. An anchor zone of an item, mobile nodes and their communication ranges: a node will replicate a content within $r$ from the anchor point, delete an item when further away than $a$ and just hold on to the item in-between.

computes per content item a priority $p_I = a_I^2 \times s_I \times T_I$ and then sends content items in ascending order of $p_I$. This policy encourages users to use the minimum necessary resources (anchor zone area and TTL) when posting an item.

In this paper, we extend this policy further to break ties between messages of equal size. The reason is that to focus the evaluation on GPS errors and remove other factors, we consider equal-sized messages with identical TTLs and equal anchor zones, so that all $p_I$s are equal. Let $d_I$ denote the distance of a node from the origin $P_I$ when encountering another node. We then define $p'_I = d_I \times p_I$ so that items further away from its origin have a lower replication priority. All our simulations use this refined policy termed "SDTF2".

## IV. Position Errors

We consider three types of position errors discussed in the following: 1) GPS not active or unavailable; 2) GPS update (or: reading) interval; and 3) GPS position errors.

### A. GPS Availability

We define $\rho$ to be the fraction of nodes that have an active GPS device, which may or may not have position errors as per 3). A node without GPS is initialized to have no valid position. As long as it does not have a valid position, the node will not create content items, nor exchange items with other nodes. However, a node may obtain a valid position when it comes in contact with at least one other node that does have a valid position. Let $n$ denote the number of peer nodes with a valid position node $A$ comes in contact with. Thus, node $A$ obtains the positions $(x_i, y_i), 1 \leq i \leq n$, from which it computes its own position: $(x_a, y_a) = (1/n \sum_{i=1}^{n} x_i, 1/n \sum_{i=1}^{n} y_i)$.

We define two policies for obtaining a valid position from another node; they define the expectation on the accuracy of the other node's position information: 1) *GPS-only* only accepts a valid position from a peer node if that peer has an active GPS and thus delivers first-hand information. This policy yields higher accuracy for the position information inferred from others but limits sourcing location information to other nodes to those with GPS. 2) *Indirect* just requires the other node to have any valid position, including having obtained it from a third party. We do not impose any limit on how old the last update to the valid position is and we do not differentiate between GPS-based and indirect positions when computing the position average. This second policy allows transitive propagation of position information—and

thus introduces potentially faster spreading of valid position at the risk of this information being less accurate or outdated.

### B. GPS Reading Interval

We define a parameter $\delta$ as the interval in seconds at which the GPS device will be (activated and) read. If the device is read at $t_0$ the position information is cached. Requests from the Floating Content system will always use the cached value for all operations, i.e., even if a node moves, the retrieved position information will not change during $[t_0, t_0 + \delta)$.

### C. GPS Errors

We finally define a simple GPS error model. The accuracy of GPS position information depends on many factors, including: the GPS device itself and its firmware; the operating environment, especially how well the sky is visible and how many GPS satellites can be seen; and, for assisted GPS, how many additional cues are available. Here we focus on plain GPS.

Because of the above dependencies, we use a trace-based approach to estimate reasonable GPS errors. To this end, we rely on the *NetRadar* [25] measurement platform. NetRadar provides mobile users with a tool to obtain instant performance measurements of their cellular (or WLAN) connectivity. These measurements are usually taken on-demand by the user (even though automated background operation is possible), so that they are irregular in time and space. Along with the results of the performance measurements (such as uplink/downlink bit rate, RTT) the system collects in an anonymized form the GPS location and the GPS location error as reported by the mobile device, plus further metadata such as the device type. The NetRadar platform aggregates the anonymized data and offers a geographic overview of the mobile network performance.

We have mined the NetRadar database for the GPS data for 2014 and 2015. To be in line with our simulation environment (we use Helsinki for synthetic mobility traces, see below) we extracted only those data points from within the Helsinki region. Moreover, we restrict ourselves to 2015 for the most recent data of a complete year so that we likely have more modern mobile devices with better GPS capabilities. Figure 2a) depicts the distribution of GPS errors as per the above measurements and figure 2b) how the accuracy differs across different device types (which we deliberately do not name). Table I summarizes the statistics per operating system. While there are differences across the device models, the ranges are sufficiently similar so that we opt for a single error model (rather than device-specific ones) across all measurements.

We cap the distribution at 1925m below which fall 99% of the measurements (the maximum value recorded was 74.5km!). We quantify the distribution into steps of 0.1% and create a table within the simulator using those steps. We validate the process by generating a errors from the table, shown to match the measured values well in fig. 2a) as dots.

We finally need to transform the GPS error ranges into a model useful for our simulations. Since the measurements taken by NetRadar are mostly point measurements, we cannot observe how stable the position reported by the GPS device

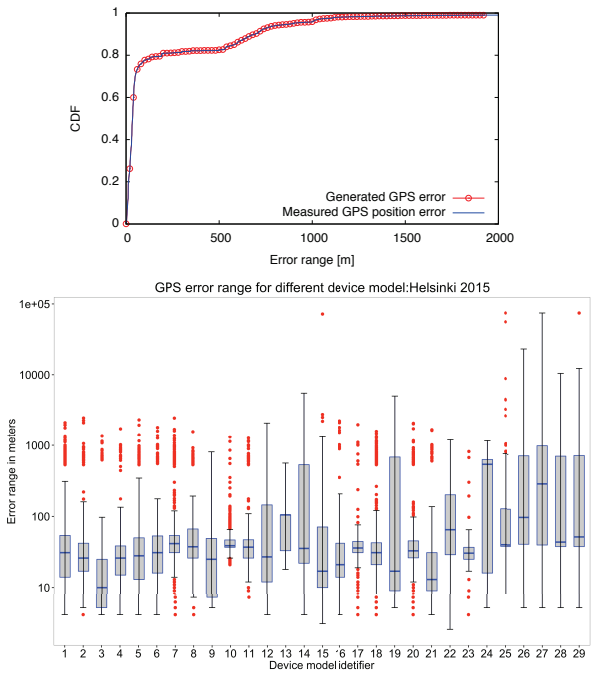



Fig. 2. a) Measured GPS error ranges and errors generated in the simulator (top); b) GPS errors observed across different devices (bottom).

| OS platform | Median | Mean | # measurements |
|---|---|---|---|
| Android | 34.5 m | 160.1 m | 922,418 |
| iOS | 71.0 m | 1265.0 m | 68,008 |
| Windows phone | 15.0 m | 206.6 m | 173,929 |
| Qt | 99.0 m | 174.5 m | 39,263 |
| Total | 36.0 m | 314.4 m | 1,203,619 |

TABLE I
STATISTICS OF NETRADAR GPS MEASUREMENTS FOR HELSINKI IN 2015 INDICATING THE MEDIAN AND MEAN GPS POSITION ERROR RANGE

is and how the error range changes over time. Therefore, we define a two-stage error process. In the first stage, every $\Delta U_1$ seconds, a node determines its present GPS error range according to the above distribution. Let $r_e$ be the error range determined in this way. The node then picks, uniformly distributed within $[0, r_e]$ a distance and, uniformly distributed within $[0, 2\pi]$, an angle $\alpha$. These are used to compute a *base error offset* $(\Delta x, \Delta y)$ to the node's actual position $(x_0, y_0)$ and obtain a base error position $(x_1, y_1)$ by adding the base error offset to the actual position. Refer to figure 3. This base error offset remains stable for $\Delta U_1$ to prevent heavy oscillation.

In the second stage, we provide for a smaller-scale oscillation around the base error position. We define a stability interval $\Delta U_2$ at which a second level offset $(\delta x, \delta y)$ is updated. We use the same mechanism as above, choosing a distance and an angle using uniform distributions to compute the second level offset, but we use only a fraction of $r_e$ for the distance: $[0, f \times r_e]$. Using these, we finally define a node's GPS position with error as: $(x_e, y_e) = (x_0, y_0) + (\Delta x, \Delta y) + (\delta x, \delta y)$. This means that within $\Delta U_2$, the error position accurately tracks the node movement, but "jumps" whenever $\Delta U_1$ or $\Delta U_2$ are updated. To illustrate the operation, figure 4a depicts the actual movement of a single node in the Helsinki City Scenario with only pedestrians (HOP), see next section, along with the perceived positions due to GPS errors.

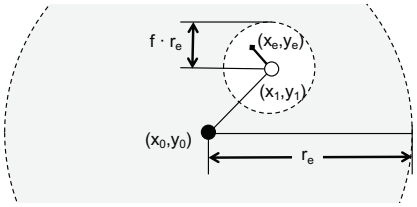



Fig. 3. GPS error generation model with two offsets being computed at different time scales.

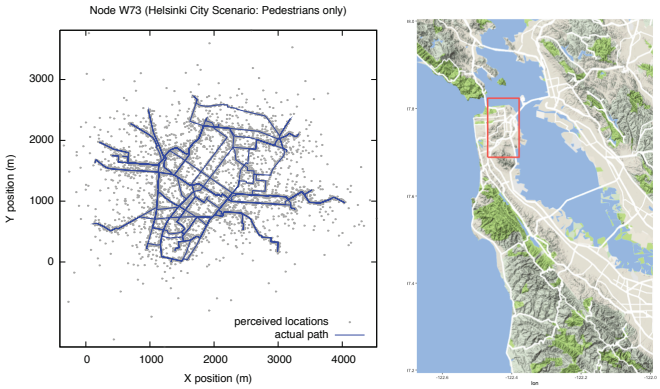



Fig. 4. a) Sample position errors of an individual node (left); b) Area of the San Francisco taxi cab traces (right)

## V. Simulation Environment

For our simulation-based evaluation, we use the ONE simulator [26], version 1.5.1 RC2 includes the code for Floating Content. We revise and extend the rudimentary code for generating position errors to implement the above model, to be included in a github branch. We also implement a new report class to create regular snapshots of ongoing simulations, `SnapshotReport`, and provide a specific subclass to compute absolute position errors of nodes. We use two classes of mobility patterns as described in the following.

### A. Synthetic mobility

We choose two different synthetic mobility models: 1) We pick *Random Waypoint (RWP)* in spite of its known deficiencies because its resulting movement, especially with lower node densities, exhibit fewer and shorter contacts than map-based mobility models and thus provide a simple way to explore one extreme in poor connectivity. 2) We also choose the more sophisticated *Helsinki City Scenario (HCS)* [26] based upon a city map ($4500 \times 3400$) of downtown Helsinki, which we modify to feature only one class of mobile nodes: pedestrians (HOP). They roam the city area like restless tourists following streets and walkways when moving to randomly chosen points on the map following a shortest path using pedestrian speeds of $0.5 - 1.5$ m/s. For both scenarios, we use 100, 200, 500, and 1000 nodes.

### B. Trace-based mobility

We use the *San Francisco cab traces (SFO)*[2] as those offer GPS coordinates (unlike pure contact-based traces) and thus allow evaluating geo-based protocols. The traces were collected from taxi cabs serving in the San Francisco-Oakland area from May 17 to June 10, 2008 (=25 days, the first and last day having only half a day of records). The traces contain records of some 500 vehicles. Each taxi was equipped with a GPS receiver, which was programmed to send location-updates (timestamp, identifier, geo coordinates) to a central server. The location-updates are quite fine-grained, with the average time interval between two consecutive location updates around 1 minute. Figure 4b depicts the area covered by the traces.

Since the original dataset contains many irregularities, we preprocess the traces and remove inaccurate locations. First, all data points that correspond to locations in the water are removed. Next, we prune "jumping" location points, identified as the location updates which would require taxis moving at unusually high speeds between two consecutive points (our threshold is 50 m/s). Two taxis (numbers 493 and 517) are entirely removed from the trace as they reported alternatingly accurate (or approximately accurate) locations and a few stationary locations, which did not allow filtering or smoothing the trajectories. [3] Finally, if two consecutive timestamps of one taxi were more than 30 minutes apart, the interval between those timestamps was considered to be an inactivity period.

These mobility traces are then formatted in accordance with the syntax of the path movement format and fed to the simulator through the ExternalPathMovementReader. The path reader uses two files: one for the paths and one for specifying activity times. Nodes follow the paths in the trace file, and pause between paths. Activity times refer to the periods of time when there is valid trace data about the node. While a node is not active, it pauses and will not generate any messages, but it will participate in content replication.

The path reader expects Cartesian coordinates with distance values in meters. Hence, the GPS coordinates have to be transformed into this format. We set the origin of the new coordinate system at point SFcenter (lon=-122.446747, lat=37.733795) and translate the trace GPS coordinates into distances from this point. This procedure causes some coordinates to become negative, but the ONE simulator translates and scales the coordinates according to the positive World dimensions. As the traces provide GPS positions but no error ranges we use the above error model also here. For the simulations, we choose the following days from the traces with the corresponding number of mobile nodes (taxi cabs):

| seed | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| day# | 2 | 4 | 5 | 8 | 9 | 10 | 15 | 17 | 21 | 22 |
| #cabs | 496 | 497 | 495 | 497 | 490 | 480 | 487 | 493 | 500 | 504 |

### C. Simulation parameters

Table II summarizes the simulation parameters we use on our evaluation. The area size for HOP is defined by the Helsinki City Map used in the ONE [26], and we use the same dimensions for RWP. The SFO map size is inferred from the taxi traces. We use a radio range of 50 m for smart mobile

[2] http://crawdad.org/epfl/mobility

[3] Some minor issues remain after processing, e.g., apparently less accurate location records when vehicles pass through mountain areas. This could be improved by using, e.g., "snap to roads" from the Google Maps API.

devices for all scenarios, but also explore the larger-scale SFO scenario with 250 m because the mobile devices would be car-based. We choose 250 MB message buffer across all scenarios as, empirically, those do not cause message drops due to buffer overflow for up to 500 nodes at 1 message/node/hour. We define three different loads: 1, 2, and 4 messages per node per hour and three settings for message parameters: *variable* varies message size, TTL, and anchor zone for each message within a given range, whereas *fixed-500* and *fixed-2000* use constant parameters. Messages are generated across the entire simulation area for RWP and HOP, but their generation is limited to a central ("core") part as indicated by the solid line square in figure 4; the latter is to avoid areas with in little cab traffic influencing the results.

| | |
|---|---|
| *General parameters* | |
| Area size | 4500m (RWP, HOP), $60 \times 90$km (SFO) |
| Radio range | 50 m (RWP, HOP), 50 m, 250 m (SFO) |
| Node buffer | 250 MB |
| *Position errors* | |
| Fraction | $\rho \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ |
| Interval | $\delta \in \{0, 15, 30, 45, 60\}$ s |
| GPS error | *none*, ($\Delta U_1 = 60$ s, $\Delta U_2 = 10$ s, $f = 0.1$) |
| *Messages* | |
| variable | $r \in [200; 500]$ m, $a \in [500; 2000]$ m<br>$T \in [1800; 10800]$ s, $s \in [100; 1000]$ KB |
| fixed-500 | $r = a = 500$ m, $T = 3600$ s, $s = 500$ KB |
| fixed-2000 | $r = a = 2000$ m, $T = 3600$ s, $s = 500$ KB |
| *Traffic load* | |
| low | 1 message/node/hour |
| medium | 2 messages/node/hour |
| high | 4 messages/node/hour |

TABLE II
SIMULATION PARAMETERS

For the position errors, we run scenarios without GPS errors ("none") and with errors as per section IV-C with the parameters in table II ("$60/10/0.1$"). We vary the fraction of nodes having their own GPS between 10% and 100% and the intervals of reading the GPS position between constantly and once per minute in intervals of 15 s. We conduct 10 runs for each simulations with different random seeds and, for SFO, using different days from the traces for variable node mobility.

## VI. EVALUATION

We first present the results of our synthetic mobility scenarios to explore the parameter space and then validate these findings using the traces. We ran 9,600 simulation settings for the synthetic models and 1800 for the trace-based model, with 10 seeds each, so that we have to constrain the following presentation to selected—representative—results.

### A. Synthetic scenarios

Figure 5 depicts the simulation results for RWP with 100 and 1000 nodes; we begin with RWP because here some of the observed effects are particularly pronounced. Each point indicates the fraction of messages reaching 95% of its intended lifetime, the error bars show the range from 75% (upper) to 100% (lower). Two expectations are obviously confirmed: increasing the message load reduces the success probability and denser networks yield better performance. We consistently see that having accurate GPS information improves performance compared to applying our error model. But we also find that delaying reading the GPS by different intervals does not appear to have a strong influence as the curves are fairly similar, shown as performance metric for different delays relative to $\delta = 0$ in figure 7a; we also note that in this regard it barely matter if GPS errors are present or not.

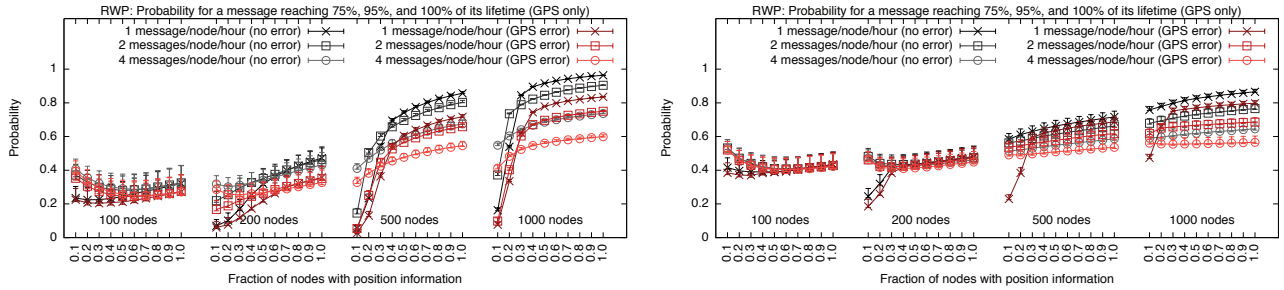
The most interesting findings are (1) the effect when varying the fraction of nodes supporting GPS; (2) the strong impact of restricting nodes learning their position to direct contacts with GPS-enabled devices (*GPS-only*) vs. allowing nodes to obtain indirect positions as well (*indirect*); and (3) a difference between fixed and variable messages.

(1) For lower node densities, the performance appears to be better if fewer nodes have GPS enabled. This is because, especially in RWP, nodes meet rarely due to their unconstrained motion. So once a node without GPS carries a message, it depends on meeting another node before it can make a deletion decision if the node has moved out of the area. The net result is that the message survives *somewhere* in the system. Investigating the number of copies for each message clearly confirms this, with an average of less than two copies created per message. So, while messages may survive, they are not useful because they are not in their anchor zone. Having more nodes with GPS enabled leads to even less replication because now messages are more frequently deleted upon the rare encounters as nodes determine that they have left a message's anchor zone. Moreover, nodes only generate messages when they have a GPS reading (no matter how old), so they may create a message based upon a reading just to delete it upon the next encounter as it was created "outside" its anchor zone. With increasing node density, this anomaly disappears as nodes have a chance of updating their position more frequently so that we see the expected behavior for 1000 nodes (which gradually emerges already for 200 nodes and is clear for 500). For 500 and 1000 nodes, the marginal performance increase reduces as we surpass 50% of nodes supporting GPS.

(2) We find that enabling nodes to learn position from arbitrary peers in a transitive fashion rather than just from authoritative sources, i.e., GPS-enabled devices, makes the above anomaly effect more pronounced as incorrect positions propagate quicker, more nodes are ready to generate messages, and the accuracy of information goes down. Nodes may also delete messages more readily if they encounter other nodes with outdated position information so that not meeting any nodes becomes a plus. As noted above, this means that useless (because ill-positioned) messages may stay around.

(3) We observe a more pronounced performance difference across different parameters when using fixed messages compared to variable ones, especially in the denser scenarios. This is because our variable messages may have larger anchor zones and are, on average, thus less susceptible to being deleted due to position errors, which is confirmed as we see more mean copies of individual messages (some 20–30% for 1000 nodes, not shown). In addition, differing message attributes yield more flexibility when prioritizing messages for replication (than just distance from the anchor point).

(a) GPS location acquisition only: fixed-500 (left) vs. variable (right)




(b) Allowing Indirect location acquisition: fixed-500 (left) vs. varied (right)

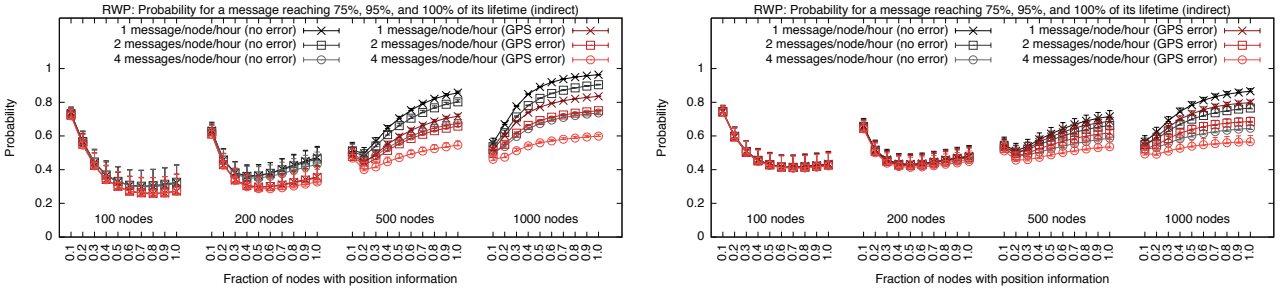
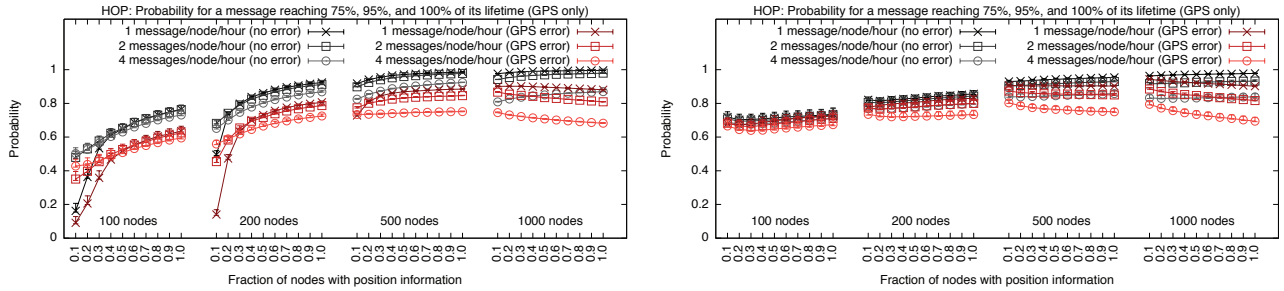



Fig. 5. RWP: Contrast of GPS errors vs. no errors for different node densities

(a) GPS location acquisition only: fixed-500 (left) vs. variable (right)




(b) Allowing indirect location acquisition: fixed-500 (left) vs. varied (right)

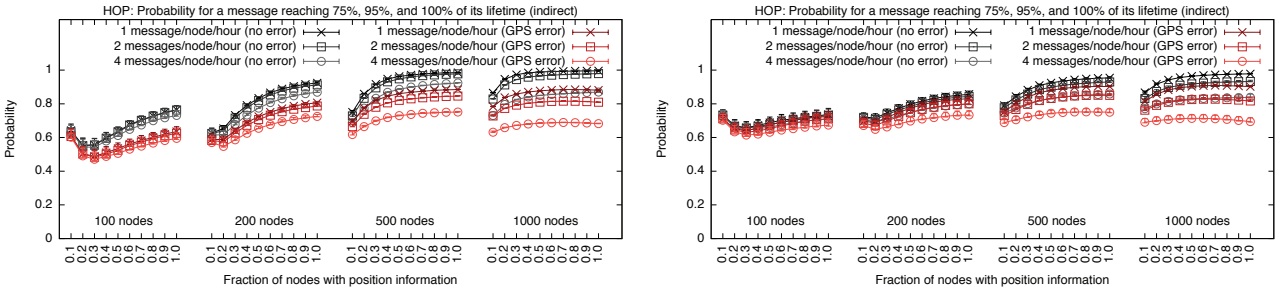



Fig. 6. HOP: Contrast of GPS errors vs. no errors for different node densities

The findings for HOP mobility are qualitatively similar, but the overall performance is much better due to constrained node movement and thus more frequent encounters (see figure 6). The more frequent encounters also reduce the anomaly observed for RWP and lead to the performance stabilizing at a smaller fraction of nodes with GPS support. It is most interesting to observe that the performance degrades with an increasing number of nodes supporting GPS with GPS errors (which does not happen in the absence of GPS errors). We attribute this to the "jumpiness" of our GPS error model. GPS-enabled nodes will exhibit less stable position properties and may occasionally jump out of the anchor zone, which then may cause content deletion. In contrast, nodes without GPS errors exhibit a more stable behavior as they only jump upon encounters. Introducing GPS reading intervals $\delta$ shows slightly less spread than with RWP, (figure 7b) and may even have a slightly more positive impact as nodes are more constrained and meet more frequently so that the delayed reading may support further replication instead of deletion.

Finally, we look at $d_e = \sqrt{(x_0 - x_e)^2 + (y_0 - y_e)^2}$, the distance between the perceived and actual anchor point when generating a message, to investigate position errors and the above anomaly. Figure 8 shows the CDFs across all messages for $d_e/a$, with $a$=500m (we also indicate $a$=2000m) for 100 nodes, indicating that for small $\rho$ only a small fraction of messages is actually generated within their anchor zones due

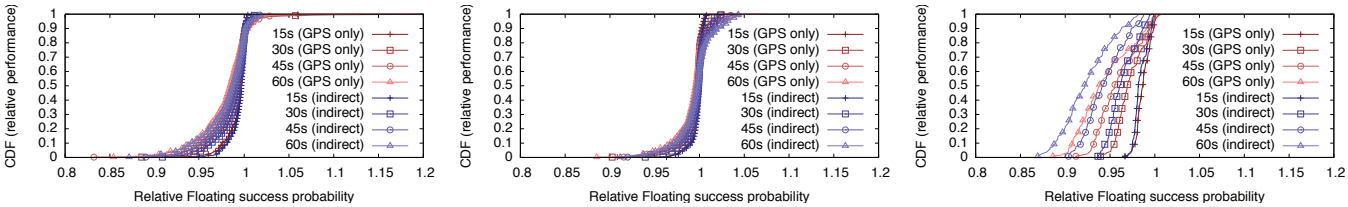



Fig. 7. Impact of GPS reading intervals on Floating Content across all densities, loads and message types: a) RWP (left), b) HOP (middle), c) SFO (right)

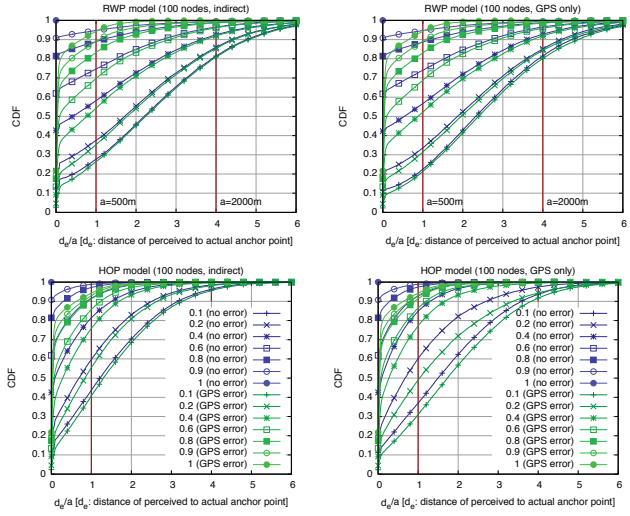



Fig. 8. Position error when posting a content item: distance $d_e$ of the perceived to the actual anchor point relative to the availability range $a$=500m

to location errors (e.g., 20-25% for RWP and 30-40% for HOP with $\rho = 0.1$). With growing $\rho \geq 0.6$, this fraction shrinks to less than 10% even for 100 nodes in HOP.

To quantify the impact on message availability within the anchor zone, we consider the period a message is available, $T_f \leq T_I$. We divide the time axis into 10s intervals and check for every message in every interval between its creation $t_c$ and disappearance time $t_c + T_f$ how many message copies $k$ are within the *actual* anchor zone. We define $\tau$ as the fraction of intervals in $[t_c; t_c + T_f]$ for which $k \geq 1$ and plot $P[x \leq \tau]$ in figure 9 for different $\rho$ with and without GPS errors. For just 200 nodes in HOP and $\rho \geq 0.6$, we find that more than 90% of the messages have at least one message copy in their anchor zone for 98% of the time (mean 98.2%) with *GPS-only* and for 96% of the time (mean 97.7%) with *indirect*. This confirms that when performance stabilizes so does the availability in the actual anchor zone. For 500 nodes, we get 98% of the messages for 98% of the time (mean 99%) for both position sharing modes, i.e., messages are virtually always in the anchor zone. We see in both figures how badly RWP positioning and availability are impacted due to infrequent encounters. Finally, while noticeable in figure 8, the practical impact of GPS errors appears marginal (cf. figure 9).

### B. San Francisco cab traces

We also look at the GPS impact in a real world setting, the result summary for which is shown in figure 10. The overall performance is lower than RWP and HOP, which is not a surprise given the much larger area. To compensate this partly, we also run simulations with messages that have a fix anchor radius of 2 km (fixed-2000). We see the same basic patterns as the fraction of GPS-enabled nodes increases (we would expect GPS in every cab) and that the impact of GPS errors is smaller (vehicles would probably have more accurate positions). If only GPS-enabled devices share positions, we need half of the node population to have GPS for the performance to roughly stabilize, otherwise the impact of the fraction of GPS-enabled nodes is less pronounced. The same reasoning as above applies, with cars moving faster making up for the lower density. The impact of the increased node velocity also becomes visible when looking at GPS reading intervals (see figure 7c). There is virtually no positive effect of the error and the different reading intervals clearly separate.

For SFO mobility, we also explored a larger radio range (250 m) for seeds 1, 2, and 3 with low message loads. We find the following mean Floating Content success probabilities for fixed-500 messages: When allowing transitive (indirect) position sharing, we obtain 0.47–0.61 with GPS errors and 0.50–0.69 without GPS errors; if only GPS-enabled nodes share positions (GPS-only) the success rates are and 0.26–0.61 with and 0.29–0.69 without GPS errors. For fixed-2000 messages, the impact of GPS errors is negligible and we get success rates of 0.76–0.86 (indirect) and 0.61–0.86 (GPS-only) in either case. Thus performance improves expectedly with transitivity and with growing radio range and anchor zone, the impact of GPS errors disappears. Moreover, the larger radio range (and thus more frequent encounters) appears to make the anomaly observed above disappear: we get monotonically increasing performance as more nodes have GPS support.

## VII. Conclusion

Our exploration of how three different types of position errors impact a location-based content sharing service shows that such a service can be fairly robust against location errors provided that a sufficiently large fraction of nodes are GPS-enabled. GPS error ranges do play a role, but the impact of other factors—such as the aforementioned fraction and the strategy for sharing location information—is more pronounced. We also find that synthetic models that approximate reality poorly (such as RWP) may provide overly poor results, nevertheless useful for understanding the issues. But content sharing services seem to suffer less from GPS errors when using models closer to or traces from reality.

After the extensive simulation studies presented here (which still only scratches the surface), our future directions are threefold: 1) evolving the GPS location sharing methods beyond simple ones used here; 2) devising an analytical model to properly characterize the underlying interdependencies of dif-

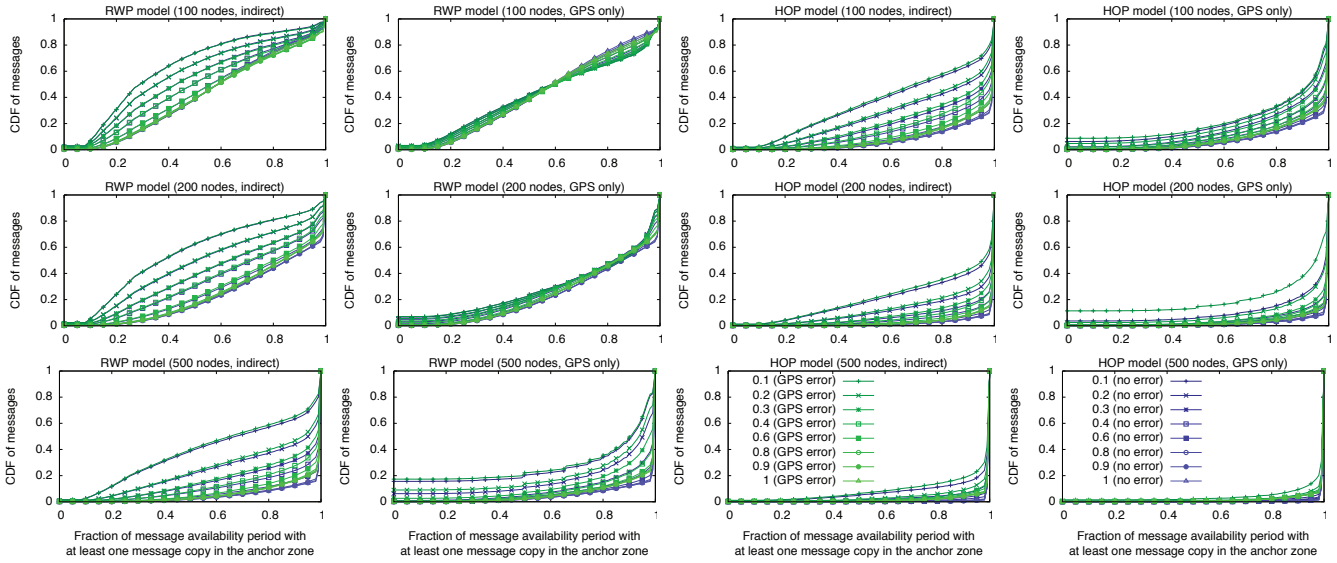



Fig. 9. Share of message copies inside the anchor zone over time (fixed-500 messages)

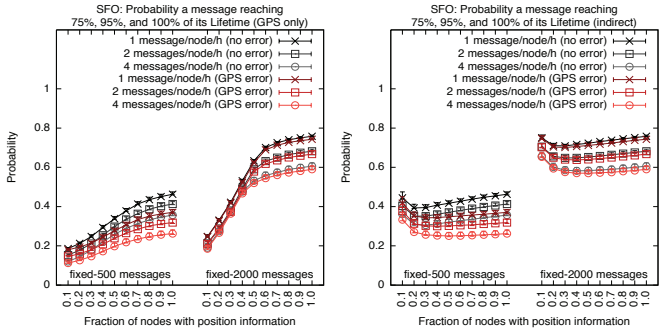



Fig. 10. SFO: Contrast of GPS errors vs. no errors for different message types and position acquisitions: GPS-only (left), indirect (right)

ferent parameters; and 3) exploring GPS error characterization further, for more locations and more in detail for individual device behavior through measurement studies.

## Acknowledgments

This work was supported by the European Commission Horizon 2020 Programme RIFE Project Grant No. 644663.

## References

[1] J. Ott, E. Hyytiä, P. Lassila, T. Vaegs, and J. Kangasharju, "Floating Content: Information Sharing in Urban Areas," in *Proc. of IEEE PerCom*, March 2011.

[2] A. Villalba Castro, G. Di Marzo Serugendo, and D. Konstantas, "Hovering information: Self-organizing information that finds its own storage," School of Computer Science and Information Systems, Birkbeck College, London, UK, Tech. Rep. BBKCS707, Nov. 2007.

[3] N. Thompson, R. Crepaldi, and R. Kravets, "Locus: A location-based data overlay for disruption-tolerant networks," in *ACM MobiCom Workshop on Challenged Networks (CHANTS)*, 2010.

[4] D. Corbet and D. Cutting, "Ad loc: Location-based infrastructure-free annotation," in *International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, 2006.

[5] E. Hyytiä, J. Virtamo, P. Lassila, J. Kangasharju, and J. Ott, "When does content float? characterizing availability of anchored information in opportunistic content sharing," in *Proc. IEEE INFOCOM*, Apr. 2011.

[6] J. Ott, E. Hyytiä, P. Lassila, T. Vaegs, and J. Kangasharju, "Floating Content: Information Sharing in Urban Areas," *Elsevier Personal Wireless Communications (PMC)*, vol. 7, no. 6, pp. 671–689, 12 2011.

[7] J. C. Navas and T. Imielinski, "Geocast—geographic addressing and routing," in *Proc. ACM/IEEE MobiCom*, 1997.

[8] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," *Wireless networks*, vol. 6, no. 4, pp. 307–321, 2000.

[9] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (dream)," in *Proc. ACM/IEEE MobiCom*, 1998.

[10] B. Karp and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. ACM/IEEE MobiCom*, 2000.

[11] J. Liu, J. Wan, Q. Wang, P. Deng, K. Zhou, and Y. Qiao, "A survey on position-based routing for vehicular ad hoc networks," *Telecommunication Systems*, vol. 62, no. 1, pp. 15–30, 2016.

[12] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network*, vol. 15, no. 6, 2001.

[13] T. Yan, W. Zhang, G. Wang, and Y. Zhang, "Got: Grid-based on-road localization through inter-vehicle collaboration," in *Proc. IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, 2011.

[14] A. Benslimane, "Localization in vehicular ad hoc networks," in *Proc. Communication Systems Conference*, 2005.

[15] G. V. Záruba, V. K. Chaluvadi, and A. M. Suleman, "Labar: Location area based ad hoc routing for gps-scarce wide-area ad hoc networks," in *Proc. IEEE PerCom*, 2003.

[16] X. Wu and B. Bhargava, "Ao2p: Ad hoc on-demand position-based private routing protocol," *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, pp. 335–348, 2005.

[17] W.-H. Kuo and S.-H. Fang, "The impact of GPS positioning errors on the hop distance in Vehicular Adhoc Networks (VANETs)," in *Proc. IEEE ICNC*, 2013.

[18] P. Brida, J. Machaj, and J. Benikovsky, "A modular localization system as a positioning service for road transport," *Sensors*, vol. 14, no. 11, pp. 20 274–20 296, 2014.

[19] J. Hemmes, D. Thain, and C. Poellabauer, "Cooperative localization in gps-limited urban environments," in *International Conference on Ad Hoc Networks*, 2009.

[20] S. Trifunovic, B. Distl, D. Schatzmann, and F. Legendre, "WiFi-Opp: Ad-Hoc-less Opportunistic Networking," in *Proc. of ACM MobiCom CHANTS workshop*, Sep 2011.

[21] J. Su, J. Scott, P. Hui, J. Crowcroft, E. de Lara, C. Diot, A. Goel, M. H. Lim, and E. Upton, "Haggle: Seamless Networking for Mobile Applications," in *Proc. of UbiComp*, 2007.

[22] V. Lenders, M. May, G. Karlsson, and C. Wacha, "Wireless ad hoc podcasting," *ACM/SIGMOBILE Mobile Comp. and Comm. Rev.*, 2008.

[23] T. Kärkkäinen, M. Pitkänen, P. Houghton, and J. Ott, "SCAMPI Application Platform," in *Proc. of ACM MobiCom CHANTS workshop*, 2012.

[24] T. Hossmann, F. Legendre, P. Carta, P. Gunningberg, and C. Rohner, "Twitter in Disaster Mode: Opportunistic Communication and Distribution of Sensor Data in Emergencies," in *Proc. of ExtremCom*, 2011.

[25] S. Sonntag, J. Manner, and L. Schulte, "Netradar – Measuring the Wireless World," in *Proc. of WiNMee*, 2013.

[26] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *Proc. SIMUTools*, 2009.