Towards distributed geolocation for large scale disaster management

Florent Coriat∗†, Luciana Arantes∗†‡, Olivier Marin∗†‡, Anne Fladenmuller∗†, Nicolas Hidalgo§ and Erika Rosas§

∗ Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
† CNRS, UMR 7606, LIP6, F-75005, Paris, France
‡ Inria, REGAL project-team, Paris-Rocquencourt, France
§ DIINF, University of Santiago, Chile

E-mail: [florent.coriat, anne.fladenmuller, luciana.arantes, olivier.marin]@lip6.fr
E-mail: [nicolas.hidalgo, erika.rosas]@usach.cl

Abstract—In the aftermath of major disasters such as earthquakes, locating individuals is crucial for passing on vital information, for example warnings and safety announcements. However, large scale disasters cause extensive damage to the network infrastructures and a generalized loss of communications in the chaos that ensues. This position paper presents a preliminary study for a geolocation service that relies on inter-device connections: mobile devices exchange positions of previously encountered devices when they come into contact. Every device thus builds a partial map of device locations and can use it to enforce geographic routing protocols that are resilient to large scale disasters.

Index Terms—disaster management, geolocation, mobile devices

I. INTRODUCTION

When dealing with large scale disasters, real-time geolocation to identify danger zones and to track/contact survivors and rescuers constitutes a major challenge. Indeed, disaster occurrences generally lead to severe communication failures due both to physical damage incurred by the network infrastructures and to overloads caused by increased traffic: official sources striving to propagate emergency announcements while individuals try to communicate with their relatives to exchange news about their safety and well-being.

Continuous geolocation in this context offers many benefits. Persons who are unaware of the danger can be notified and oriented safely towards rescue centers. Competent and able-bodied rescuers can be teamed up and directed towards helpless victims. Other survivors that made it to safety can be tracked to simplify relief efforts. Obviously, all of the above require sustained network routing.

We envision a best effort approach to allow continued routing of data: it builds upon weakly connected networks on top of user devices to share information about device locations. User mobility thus contributes to the generation and the propagation of information. Upon every temporary connection, a mobile device sends its current location along with a list of the locations received from other devices during previous encounters. In this manner, every mobile device builds a partial map of the disaster area. Our approach assumes the existence of rescue centers where servers can aggregate partial maps and correlate them to build a global map, which in turn facilitates routing.

The present paper constitutes a preliminary study towards such an approach. Section II describes the model we designed to simulate the movement of mobile devices in the midst of large scale disasters. Section III presents the evaluation metrics and a simplistic geolocation protocol we intend to use in order to assess our own geolocation scheme. Section IV compares our work with other related research. Section V concludes and details the remaining steps towards the completion of the work proposed in this position paper.

II. MODELLING A LARGE SCALE DISASTER

In the context of large scale disasters, our ultimate goal is to build a solution that facilitates the navigation of device users towards their destination and that enables the routing of messages from one point to another in the absence of legacy network infrastructures. To this end, we intend to conceive communication protocols among mobile devices: exchanges of positioning data lead to the generation of a global view of device locations, which in turn provides a basis for geographic routing. Our first step consists in designing a model that allows assessments and comparisons between such communication protocols. The second step, presented in section III, is to conduct these assessments and comparisons by running simulations on top of this model.

A. Model assumptions

We consider mobile device users that can travel freely along the paths of a map towards their destination. We restrict our study to pedestrians: device users move at different speeds within the range between standing still and running. Users can become incapacitated and will remain motionless. Accidents that block pathways on the map may occur at any moment.

The behavior of device users corresponds to their level of awareness with respect to the situation. We identified three such levels:

1) Oblivious. Oblivious users either have no awareness of the danger they are in or are clueless regarding the behaviour they should adopt towards ensuring their own
safety. As an example, a global power outage combined with the leakage of a toxic gas can lead to the former case: many individuals will remain oblivious until they are warned of the crisis. The latter case could correspond to users who have heard of the leakage on the radio but have not yet acquired information concerning the rescue centers where they will be safe.

2) **Alarmed.** Users become alarmed when they receive warning of the disaster and acquire at least one target destination to reach for safety. We consider two means of notification: an alarmed user will notify an oblivious user upon hearing the danger; otherwise an oblivious user may catch news of the danger and acquire safety information at home as the authorities start delivering such information over the radio.

3) **Safe.** Users consider themselves safe once they have reached their designated target destination.

Mobile devices interact via Bluetooth links, with a range of 10 meters. We chose Bluetooth over WiFi as it is far more energy-efficient, and we believe this is an essential criteria during disasters. Every device is associated with a unique identifier, and incorporates both a physical clock and some positioning feature such as GPS.

Geolocation servers, called "hotspots", are accessible at the appointed rescue zones on the map. These locations are fixed and are considered common knowledge: mobile devices store these locations in advance, and orient the device owner towards a selected hotspot upon receiving a notification of a disaster occurrence. Every hotspot dedicates significant CPU and storage resources to the geolocation service. It can communicate with mobile devices via Bluetooth but cannot exchange data with other hotspots since we assume that legacy network infrastructures are down.

**B. Emergency Movement Model**

Given the above assumptions, we designed a movement model to represent the mobility of device users in the midst of a large scale crisis.

Our Emergency Movement model reflects the different behaviours of a device user during a crisis. Figure 1 gives the state diagram that regulates these behaviors. Users either start as oblivious or alarmed. Oblivious users can start inside or outside their homes. Those outside walk randomly around the map for some time, and then head home. When inside their homes, oblivious users remain there until they receive warning of the danger. Warnings come from two possible sources: either an alarmed user passes by and notifies the oblivious user when they connect, or the oblivious user becomes self-warned (with a statistical probability that can be set). Alarmed users know of at least one safe target destination and try to reach it. Our model allows accidents that block pathways dynamically, so an alarmed user may become stranded if there are no more accessible paths left towards its target. To ensure that simulations based on our model do end, we do not require all users to reach the safe state. Once a set percentage of users are safe, all remaining users become stranded. Finally, upon reaching its target safety destination a safe user remains stationary somewhere within the bounds of the safety area.

![Emergency Movement Model - State Diagram](image)

**Fig. 1. Emergency movement model - state diagram.**

![Emergency Movement Model - Class Diagram](image)

**Fig. 2. Emergency movement model - class diagram.**

The Emergency Movement model combines a couple of classic movement models with a few of our own. Each of these models corresponds to a user state. Figure 2 represents the class diagram associated to the state diagram of figure 1. Oblivious users regain their homes with a random path map-based movement (RandomPathMapBased). At home, they levy-walk randomly within a very limited area (Home). Alarmed users try to reach their designated evacuation center with a variation of the shortest path map-based POI movement (ShortestPathMapBasedPOI): since the shortest path may be cut off at any moment, it is recomputed every time the user stumbles upon a dead-end. Once they reach their designated safety area, safe users pick a location at random within the area, walk straight to it and remain stationary from then on (EvacuationCenter). Finally, alarmed users who are stranded for lack of any path towards their safety destination stop moving altogether (SOS).

Figure 3 shows an example of our model applied to a simulation. This portion of a larger map contains only one
safety area; several pathways are inaccessible due to accidents. In this snapshot of the simulation, two device users (p100 and p43 – not very readable because they’re standing very close to each other) have actually reached safety. p22 and p42 are still unaware of the disaster. p22 awaits notification at home while p42 is walking around randomly. p3, p4, p61, and p99 are alarmed and trying to reach the safety area. p30 and p84 are stranded between two accidents that prevent any possible escape.

III. ASSESSING GEOLOCATION SCHEMES

Along with our Emergency Movement model, we need a tool to implement and evaluate distributed protocols to geolocate devices during large scale disasters. The ONE simulator [1] fits our requirements: it is specifically designed for evaluating Delay-Tolerant Network (DTN) routing and application protocols, it provides implementations for a wide range of classic movement models, and it can import real-world maps as well as mobility data from real-world traces to produce more realistic simulations.

The question remains about how to determine the efficiency and appropriateness of a geolocation scheme. The present section discusses metrics for this purpose, and assesses a naïve protocol that we deem worthy of comparison.

A. Evaluation metrics

In our opinion, a geolocation scheme must aim towards piecing together information that is both accurate and up to date. Another goal is to make this information as available as possible to all users, whilst minimizing the global cost of communications. With this in mind, we propose the following metrics to measure the performance of any given geolocation protocol.

- **Exploration speed.** This first metric computes the proportion of devices that have exchanged locations with at least one other device as time passes. It reflects efficiency when locating individual devices. Ideally, a geolocation scheme ought to find as many devices as physically possible over a given timespan.

- **Graph coverage.** Our second metric calculates the proportion of links of acquaintance created since the start of the experiment. A new link of acquaintance $\Lambda_{A\rightarrow B}$ is spawned every time a device $A$ acquires a location for a previously unknown device $B$. This metric verifies how much knowledge of the global map is actually disseminated among the devices. Given a scenario with $N$ devices, the total number of possible links of acquaintance is $N^2$ (by assumption, every device is constantly aware of its own location). A good geolocation scheme should tend towards a large graph coverage as fast as possible.

- **Positioning accuracy.** The error associated with the link of acquaintance from a device $B$ to a device $A$ equals the euclidian distance between the stored location on $B$ and the real-world location of $A$. When $A$ sends its location to $B$ and then both devices move on in separate directions, the error associated with $\Lambda_{B\rightarrow A}$ increases as $A$ and $B$ proceed farther away from each other. Our third metric computes the average error for all links of acquaintance. Thus it represents how close the average partial view of a device is to the real situation on the map: the smaller the accuracy value, the better.

- **Network load.** Our final metric calculates the total number of messages, as well as the total quantity of data exchanged amongst mobile devices in order to carry out the geolocation protocol. It measures the cost of the protocol on top of the DTN: a protocol that generates too much overhead has a negative impact on energy consumption and hinders the availability of the network for other applications.

B. Comparison with a naïve geolocation scheme

We have defined a first, simplistic geolocation protocol based on the epidemic propagation of the estimated locations among mobile devices. Every device maintains a counter that it increments upon every connection, and a table that associates a counter value and a location with every link of acquaintance. When two devices connect, they exchange their locations, their counter values and their tables. This allows every device to bring its table up to date: it inserts new links of acquaintance acquired during the encounter into its table, and it updates locations associated with a known device but with a fresher counter value.

Additionally, devices maintain a list of locations of unexpected roadblocks: every time a device comes across a roadblock it adds it to its list and will propagate the information upon every ulterior connection. This reduces the time it takes for device users to reach safety zones.

We implemented this protocol for simulation purposes on the ONE, and ran experiments detailed hereafter. Our experimental setup assumes that there are no bounds on the storage capacity of mobile devices and on the amount of
information that can be exchanged between devices during a single connection. We are aware that these assumptions are not realistic, however they provide an experimental upper bound on the exploration speed and on the graph coverage.

We used a map of a portion of Santiago de Chile (figure 4) for our experiments, set the real addresses of medical facilities as safety zones and ran simulations with 500, 1000, and 2000 nodes initially distributed at random throughout the map. We produced our results by running each simulation 20 times and by computing the average of every output.

Figure 5 shows the exploration speeds obtained with our epidemic propagation protocol for different densities of users. As expected, the larger the number of devices on the map, the faster they explore the map and encounter isolated devices. An interesting result is that all three curves converge (> 90%) after an hour, which does not seem that long given the significant surface of the map (29 km²). The remaining 10% of devices either take much longer to come across a first device or fail to encounter any device at all, producing a tail effect visible on all three curves. This small fraction of devices prevents the map exploration from reaching 100%. To ensure that simulation runs will actually end, our experimental settings include a threshold for the number of devices that make it to safety. Above this threshold, all remaining devices are set to stranded and the simulation stops.

Figure 6 illustrates the progression of the graph coverage obtained with our epidemic propagation protocol for different densities of users. Here also, the results confirm intuitive expectations: the higher the density of devices, the faster they propagate knowledge among themselves. A result that is less trivial is that a higher density of devices guarantees a higher upper bound for the coverage, regardless of the time that passes. We imagine that there is a critical mass of devices beyond which this upper bound cannot increase, but confirming this intuition will require further experimentations.

Figure 7 provides the progression of the positioning accuracies obtained with our epidemic propagation protocol for different densities of users. Obviously, even though the results are better for higher densities, they are overall very bad. Devices do not discard links of acquaintance that they acquired in the distant past. In a more realistic setup, our epidemic protocol could prioritize the locations to exchange reversely to their order of encounter. Indeed, if device $A$ meets $B$ and then $C$, the location for $C$ is more likely up to date than the location for $B$. Coupled to a simple LRU cache policy, we believe it could improve these results and plan on integrating this solution in our next implementation.
IV. RELATED WORK

Geographic routing protocols prevent the maintenance of routing tables on mobile nodes and thus appear as a good routing alternative for DTNs. These protocols rely on an accurate location service to retrieve the exact position of the destination. Quantifying the accuracy of the position estimate thus constitutes a first step to evaluate the performances of routing protocols in a crisis scenario.

Several geographic routing protocols have been proposed in the literature. In these kinds of systems, location servers uphold the mapping between nodes and location: they update the map periodically and handle lookups from other nodes. Each node usually determines its own geographic location with some global positioning system like GPS. As we aim to tackle reliability and scalability issues, we are particularly interested in location services which are decentralized. By exploiting node locations, effective and scalable global geographic routing protocols can be implemented and present better performance than other types of solutions: mobile ad hoc networks (MANETs) topology-based protocols such as DSR, DSDV, AODV, intermittent connectivity network (DTN) protocols that often use store-carry-forward or contact-based solutions or vehicular ad-hoc network (VANET) protocols.

Distributed location service protocols for MANETs can steer towards different approaches: a quorum approach ([2] and [3]), a greedy approach such as GPSR [4] or (GPCR) [5], a hierarchical-based location service such as GLS [6], HLS [7] or a hash-based approach like SLURP [8] or GHLS [9].

In a quorum-based location service [2] [3], each node updates its position by sending the information to a subset (update quorum) of nodes. When a node wants to know the location of a target node, it requests its location from a subset (query quorum) of nodes. The two subsets must be designed so as to have a non empty intersection. Node location is, in average, disseminated to $O(\sqrt{N})$ other nodes.

Greedy Perimeter Stateless Routing (GPSR) [4] is a routing protocol for wireless networks that uses router positions and message destination to make decisions about message forwarding. GPSR makes greedy forward choices based only on information about the locations of nodes that are one hop away from the routers in the network topology. Thus, in case of topology changes due to node mobility, GPSR can use local topology information to compute new routes quickly. Another example of routing protocol that uses a greedy algorithm to forward messages is the Greedy Perimeter Coordinator Routing (GPCR) [5]. However, both protocols can forward messages efficiently provided that the underlying network is fully connected.

In a hierarchical-based location service like GLS [6], the area is recursively divided into a hierarchy of grids. For each node, one or more nodes in each grid are chosen as its location server. Therefore, the density of location servers for a node is high in areas close to the node and decreases exponentially as the distance to the node increases. Each mobile node periodically updates its current location on a set of location servers which are determined according to the geographic grid and the node identifier, GLS disseminates each node location to $O(\log N)$ other nodes.

In a hashing-based approach (SLURP [8], GHLS [9]), a known function maps each node identifier to a fixed home region composed of one or more nodes. The nodes of the region store the location information about the node and, by hashing the node identifier, other nodes can obtain its home region and therefore its location. Each node location is disseminated to $O(1)$ location servers.

All the above location service protocols are decentralized. However they target MANETs, so their assumptions are inconsistent with large scale disasters: relatively small communication latency, no network partitioning, and a mobility model which does not characterize crisis/disaster scenarios.

The DTN-based Price [10] protocol proposes a hybrid strategy that firstly exploits a geographic approach where messages are propagated towards nodes along the direction of the destination. It switches to a contact-based approach when the message is close to the destination. Price predicts future contacts based on history of past contacts and assumes repetitive human behaviors. In [11], the authors also argue that the short-term future locations of vehicles can be predicted and develop a geographic routing protocol based on this assumption. GeOpps [12] is also a delay tolerant network routing algorithm that uses GPS information from vehicles. It exploits routes suggested by the GPS of every vehicle to piggyback message routing information on the vehicles. Other VANET examples that use node positions are GPSR-DTN [13], which adds a store-carry-forward mechanism to GPRS, and GeoDTN+Nav [14], an extension of VANET Cross Link Corrected Routing (VCLCR) [15] which improves VCLCR by exploiting the vehicular mobility and on-board vehicular navigation systems. Notice that all these systems are either for VANETs or do not consider any mobility model in a crisis/disaster scenario.

V. CONCLUSION

This position paper presents a preliminary study for a distributed geolocation service that is resilient to large scale disasters. Mobile devices maintain local lists of the positions where they encountered roadblocks and other devices; upon every connection, devices share their respective lists in order to improve their partial views of the global map. As part of this undergoing work, we designed a mobility model that is specific to large scale disasters, we formulated metrics to assess the performance of a distributed geolocation model, and we devised a very simplistic algorithm based on epidemic propagation to test our model and to calibrate our future experimentations.

In the context of this project, our next step is to conceive a full-fledged protocol that: proves resilient, explores the map and discovers devices efficiently, and positions devices accurately without inducing excessive overhead. Our ideas for such a protocol include using the hotspots located in safe areas to centralize information and build a more accurate version of
the global map that can then be used for optimized geographic routing. We shall evaluate these properties by implementing and running the protocol on our experimental testbed. Concurrently, we intend to implement a revised version of GLS [6] for comparison with our own solution.

ACKNOWLEDGEMENT

This work is supported by the STIC-AmSud project RESPOND, Reputation and Energy-Aware Search for Supporting Natural Disasters.

REFERENCES