Pics-On-Wheels:
Photo Surveillance in the Vehicular Cloud

Mario Gerla, Jui-Ting Weng, Giovanni Pau
University of California, Los Angeles
Computer Science Department
{gerla,jtweng,gpau}@cs.ucla.edu

Abstract - **Cloud computing allows user to access remote hardware, data and software through the network. However, many of these resources today are found on mobiles. For example, modern vehicles provide powerful platforms for computation, data delivery, storage, and sensing. This paper introduces the notion of Mobile Cloud computing to tap mobile resources. As an example, we describe the Pics-on-Wheels service, a Vehicular Cloud service that delivers images on demand to citizens by using vehicles' on board cameras. A server (eg taxi dispatcher or Navigator Server) accepts requests from members and assigns the photo clicking task to vehicles close to the target. The feasibility of the Pics-on-Wheels service is demonstrated in a San Francisco scenario using published taxicab routes.**

Keywords-Mobile Computing Cloud; Cloud Service; Pics Net

I. INTRODUCTION AND RELATED WORK

Vehicles are getting smarter as they carry more onboard processors and devices. For example, Google crowd sourcing application collects road congestion data from on board GPS devices [1]. The Cabspotting.org website [2] tracks San Francisco's taxicabs throughout the Bay Area. These services enable vehicles to gather and exchange information that can improve road mobility.

Modern vehicles are capable of a wide range of applications, including safety, networking, and content distribution. Recently, visionaries have proposed to connect vehicles and turn them into a cloud-like service provider. Olariu et al proposed several architectures to tap vehicles for services as in a cloud in [3-4]. Huang has proposed mobile software structures targeting similar goals in [5]. Gerla projected mobile cloud intelligent transport services in [6].

This paper introduces surveillance as a location based service provided by the "vehicular cloud". The surveillance application exploits the mobility of the service nodes (vehicles) to extend the coverage beyond the reach of static sensors (e.g., video cameras).

The vehicular cloud differs from today's Internet cloud in the nature of its hardware and sensor platform [7-8]. The vehicle cloud builds its platform opportunistically in a totally distributed control, from spare and idle resources made available by participating vehicles. Another major difference is mobility, enabling location based services in the vehicle cloud. In fact, the majority of services provided by the vehicle cloud are expected to be location dependent.

While a single vehicle has the ability to provide location based services on its own, it does not provide enough "interest" for remote users to subscribe to its service. On the other hand, a group of vehicles, forming a vehicle cloud, provides a more widespread, consistent, and reliable service for the urban area. Using collaborative mobile vehicles, the vehicular cloud succeeds where the single vehicles cannot.

Mobile Clouds can be formed also with smart phones. However, smart phones are limited by battery lifetime. Cellular phone users do contribute crowd-sourced images of exceptional events (eg, riots, police abuses, natural disasters) for a civic sense of duty. However, they are not likely to agree to collectively offer crowd-sourcing as an organized service, in part because of the limited resources and also to avoid the risk of exposing themselves. Vehicles, on the other hand, have no power nor exposure concerns, thus they are more suitable to form a mobile cloud that provides surveillance services.

978-1-4673-5288-8/13/$31.00 ©2013 IEEE

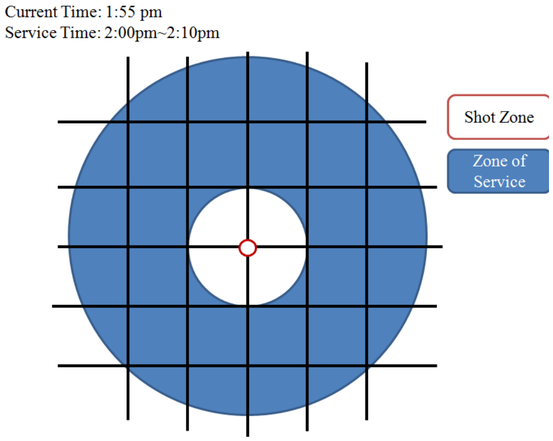

Figure 1. Location Model

One challenge in this Service is vehicle route tracking. In order to provide such a service, the manager must have access to recent location updates from potential witness vehicles. The update interval must less or equal to the response time required by the customer. With current on board GPS technology, the mobile node does the updating. The traffic overhead can be unmanageable when thousands of vehicles must update their positions within seconds. In fact, the authors of RoadTrack [9] suggest that vehicle updates should be no faster than minutes per vehicle, lest the spectrum becomes congested.

The rest of the chapter is organized as followed. Section II explains the Pic-on-Wheels photography service. Section III and IV define the Service Implementation. Section V presents performance results for the San Francisco Taxi based Service. Conclusions and future work are in Section VI.

II. Pics-on-Wheels Service

The Pics-on-Wheels service selects a group of vehicles to take photo shots of a given urban landscape within a given timeframe as requested by a customer. To participate in this Service, vehicles register to the centralized cloud manager as opt-ins. They also upload their own GPS location periodically to the cloud manager. The on board navigation system in each vehicle maintains the trace of the vehicle for a predefined time period. We will use the following terminology: **Service Request**, the time and location of photo request; **Service Time**, the photo delivery deadline. **Shot Zone**, the area where vehicle is able to shoot the photo; **Zone of Service (ZoS)**, the zone comprising the vehicles that can reach the Shot Zone within the Service Time; **Cloud Management Unit:** The cloud service manager responsible for tracking vNodes and predict their locations.

Figure 1 shows an example of Pics-on-Wheel service. This picture is a snapshot at time 1:55pm, The service request starts 5 minutes later, and ends 15 minutes later. The blue circle indicates the Zone of Service.

The Customer requests service with a message (Time, Location)) to the cloud management unit. The latter searches for candidate vehicles in ZoS as shown in Fig 1. It is possible that no vehicle exists in the area. In such case, the service request is rejected.

III. The Photo Shot Algorithm

In this study we assume that the POW service is centrally controlled via an Internet Server. Server, vehicles and customers communicate via 3G. For example, taxicabs talk to dispatcher (once every few minutes, say). In the same way they also talk (via an interactive on board navigator) to the Navigation Server to get a better route. Since the 3G or LTE system does not support broadcast from an Internet Server to cellular users, the Server must wait and be contacted by vehicles directly before it can invite them to provide the photo service. It is thus clear that one of the main performance factors of this service is the frequency by which vehicles call the Server. The ensuing delay is a penalty of the vehicle calling model with respect to the broadcast model. There are however also two immediate advantages. The centralized Server can screen vehicles' requests for sensitive services, before the service is announced to the entire vehicle cloud. Also, the Server, in both navigator and dispatcher case, learns the final destination of the vehicle and can infer the path to such destination.

The next step is to find the candidates that can perform the service. These candidates by definition are in the ZoS. Note that only a subset of vehicles in this region can do the job, namely the ones that will pass by the target. With the ZoS knowledge, the Server can estimate how long it will take to satisfy a customer request. Assuming no a priori knowledge of vehicle trajectories, the Server can compute the probability P that a vehicle randomly picked within the circle drives by the central target, under the assumption of uniform vehicle direction selection. From P and using the knowledge of average urban traffic density and query frequency, the Server can estimate the success rate and thus announce this rate to the perspective customer. For example, if the customer wants a picture of the crowd in front of the City Hall within 5 minutes, the Server will respond that it can deliver the picture with 65% probability. The customer may then either relax the time and accept the service; or, it may request that a taxi drives by the target at extra cost. In the experiments we have also evaluate a very simple (lower bound) case where the vehicle does not know where it is going next. Then, when it receives the invite from the Server, either takes the picture if it is in the Shot Zone, or it drops the invite (ie it does not memorize in case it lands on SZ later).

The vehicle selection procedure for the photo shot can be summarized in four steps:

Step 1: the Server upon receiving the request from a customer computes ZoI

Step 2: Server estimates the number of qualified candidates in ZoI and accepts/rejects the request

Step 3: If accepted, the Server awaits for calls within ZoI and invites authorized vehicles to provide the service.

Step 4: The vehicle compares target with own route and accepts/rejects the invitation.

IV Forensic Picture Service

The Pics-on-Wheels service can also be used to retrieve pictures taken spontaneously by cars as part of background ambient surveillance. Suppose vehicles routinely capture license plates, car makes and other surrounding images as they roam through city streets. Then, a valuable service for forensic investigators is for the cloud to identify "witness" vehicles that took pictures in Zone of Interest (ZoI), say, and in the interval of interest DT. For example, if the insurance company is investigating a vehicle collision, it will request pictures taken in a given intersection at the time of the accident.

How can a customer request and receive this service? Let us consider the Navigator model this time. The customer sends the service request (street #, time of accident) to the Navigator Server (which serves also as the Cloud Forensic Service Manager). From the difference {current time - time of accident}, the Navigator Server figures out how far the "witness" vehicles can have travelled. This is called the Zone of Witnesses (ZoW), similar to the previous ZoS . When a vehicle sends a query to the dispatcher via 3G, the latter checks if the vehicle is in the ZoW. If so, it includes the service request (street #, time of accident) in its response to the vehicle's query inviting it to submit the evidence. If the vehicle traversed ZoI (ie street # in the query) in interval DT, it returns an OK committing to upload the images taken in ZoI before the deadline.

The Forensic Picture Service implementation is very similar to the Photo Shoot service. There are some differences in performance, however. The response time requirement is not as critical as in Photo Shoot, since the event has already happened. Rerouting vehicles to take the picture for an extra fee does not apply here. The success rate is proportional to number of vehicles that witness the accident, ie, that are in ZoI during DT. The DT time constraint greatly restricts the number of successful matches with respect to the Photo Shoot service. To obtain a reasonable success rate one must rely on a large population as the one controlled by the Navigator Server (as opposed to the taxicab population controlled by the dispatcher).

In our study we will evaluate the performance of Forensic Photo Service as a function of vehicle density, query frequency, accident window DT, for given response time.

The careful reader will note that the Forensic Service is similar to Mobeyes [10]. However, Mobeyes was totally distributed. It required motorized agents to investigate the accident. In such cases, epidemic dissemination of vehicle metadata (ie, place, time, vehicle ID) was proposed to increase the probability of discovering witnesses. In our case, epidemic creation of proxy records is not required since the probing of the vehicle cloud is centralized, via 3G.

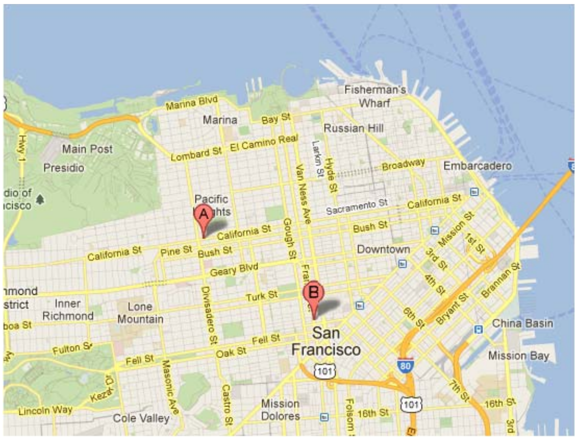

Figure 5 (a) SF taxi map, with shot zone

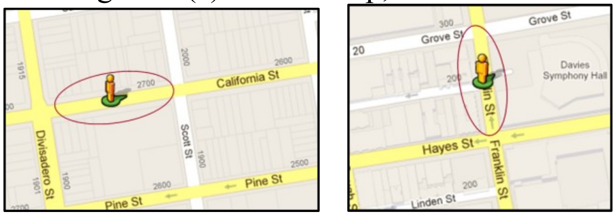

Figure 5 (b) targeted photo shot area

V. Simulation Results

We evaluate PoW performance using San Francisco taxicab traces collected via cabspotting [2] and archived on CRAWDAD [11]. This dataset contains 500 taxi traces from 5/17/2008 to 6/10/2008. Each taxi contacts the Server and updates its location about once per minute. At each contact the Server can deputize the vehicle for PoW service.

Figure 5 shows the targeted region. The Shot Zone is a 100 meter road section. We take the traces from 9:00 am to 9:00pm, and aggregate them into five 2-hour intervals.

Figure 6 shows the number of successful visits to SZ (and thus photo shots) that can be deputized in a two hour period using cabspotting locations or full path knowledge, respectively. We carry out this analysis in two different locations in Downtown San Francisco. We assume that the vehicle contacts the Server at exactly each cabspotting time. The vehicle must either immediately commit to take the shot, or decline. If the vehicle has no precise knowledge of the path (for example, the driver has not selected a route yet), then it can commit only if it happens to be in the Shot Zone. In this experiment, Service Time constraint is relaxed (ie, it is set to infinity). Thus, the path aware vehicle can commit for any future visit to the Shot Zone at the first invite. The path non aware vehicle must inspect at each checkpoint and takes a photo shot if there is a match. Each vehicle can take only one shot (no repetitions). The number of photo shots taken by the path unaware vehicles, (ie cabspotting time) is smaller than that of the path aware ones. In fact, it is about 3 to 6 times smaller depending on traffic congestion for both case A and case B. This is because the path aware vehicle can check every 10 to 20s for a Shot Zone match (as it takes 10 to 20s on average to traverse a Shot Zone in light or heavy traffic conditions), while the less path aware vehicle has such opportunity only every 60 seconds, at cabspotting instants. These results show that path

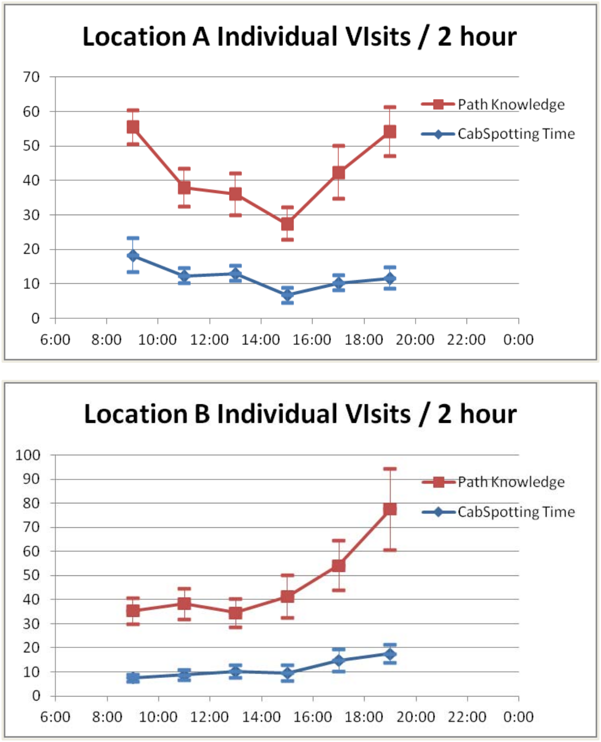

Figure 6 Visits per 2 hour

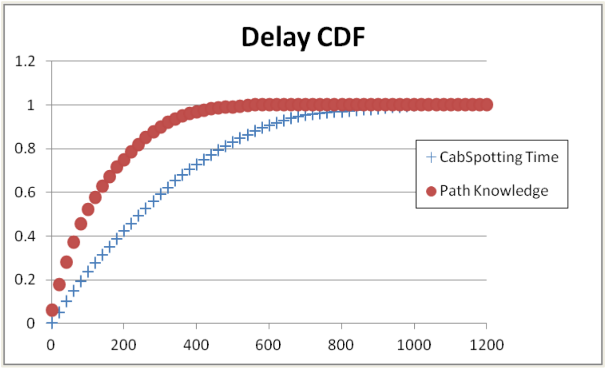

Figure 7 Delay CDF

awareness is important for efficient PoW service delivery. In fact, the longer the update interval, the more critical path awareness becomes

Figure 7 shows the CDF of service response time. Without path knowledge, more than 50% of the requests take longer than 4 minutes to complete. However, with path knowledge, 80% of the requests are completed within 2 minutes. This result shows that even with just 500 taxicabs, Pic-on-wheel service is able to provide good service coverage to a road section. The results confirm the importance of path awareness. Of course, path awareness will be the rule when the Service is offered by the Navigator Server.

Figure 8 show the simulation results for Forensic Picture Service. The service is limited not only by the service area, but also by the service time window when the accident occurred. Each vehicle sends now an update once every 2~10 minutes, less than before because the number of vehicles has increased and overhead must be kept at a reasonable level. We ran the simulation using the same target as in Figure 5 requesting a picture that was taken within the last 5 minutes (ie ToS = 5 min), with a 30 second time window.

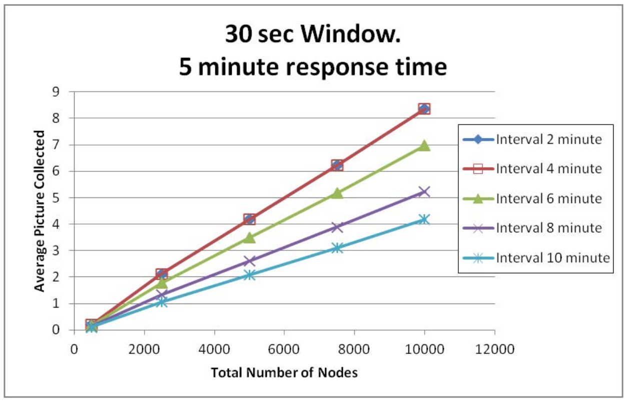

Figure 8(a) Average picture collected

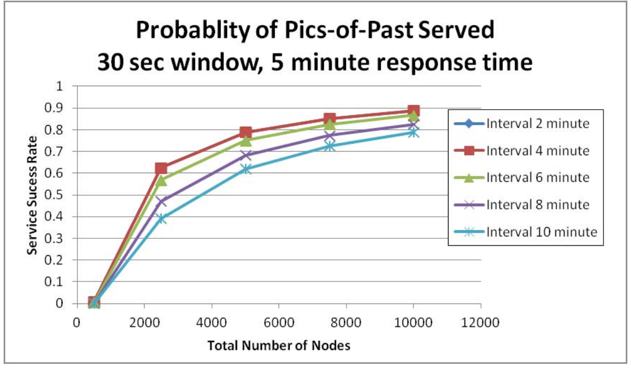

Figure 8(b) Service success rate

In order to increase traffic density, we take taxi traces collected from different dates and fold them into the same day. For example, by aggregating 10 2-hr traces, we emulate a 5000 vehicle scenario. By increasing the number of vehicles, the number of pictures increases proportionally. The left most datapoint in Figure 8(a) shows that with taxicabs only, the picture retrieval success rate = 0.9 %. With 5000 vehicles, the success rate climbs to 78%.

VI Conclusion and Future work

This paper has described a vehicle cloud service that exploits vehicle predicted routes to provide photoshots of target locations on demand. Key to performance is the ability to identify the candidate vehicles (vNodes) that take the shots. The paper proposes an algorithm to minimize the number of vehicles sufficient to guarantee the service.

Acknowlegdment

This research was sponsored by NSF project CNS-1111971: "Collaborative research: closing the loop between traffic/pollution sensing and vehicle route control using traffic lights and navigators."

References

[1] Google mobile blog, "The bright side of sitting in traffic: Crowdsourcing road congestion data", August, 2009

[2] http://www.cabspotting.org

[3] M. Abuelela, S. Olariu "Taking VANET to the clouds", Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia, 2010

[4] M. Eltoweissy, S. Olariu, M. Younis," Towards Autonomous Vehicular Clouds", Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2010, Part 1, 1-16

[5] D. Huang, X. Zhang, M. Kang, J. Lou,"MobiCLoud: Building Secure Cloud Framework for Mobile Computing and Communicaton", 5th IEEE International Symposium on Service Oriented System Engineering, 2010

[6] M. Gerla. Vehicular Cloud Computing, VCA 2012 Proceedings, Cyprus, June 2012

[7] "A Break in the Cluds: Towards a Cloud Definition"

[8] R Buyya, CS Yeo, S Venugopal, J Broberg, I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality ford delivering computing as the 5th utility", Journal of Future Generation Computing System, June 2009

[9] P Pesti, L Liu, B Bamba, A Iyengar, M Weber "ROADTRACK: Scaling Location Updates for Mobile Clients on Road Networks with Query Awareness", Proeeding of the VLDB Endowment, Volume 3, Issue 1-2, September 2010

[10] U. Lee, B Zhou, M Gerla, E Magistretti, "Mobeyes: smart mobs for urban monitoring with a vehicular sensor network", IEEE Wireless Communication, 2006

[11] CRAWDAD, http:// crawdad.cs.dartmouth.edu