Efficiently Managing Location Information with Privacy Requirements in Wi-Fi Networks: a Middleware Approach

Paolo Bellavista, Antonio Corradi, Carlo Giannelli
Dip. Elettronica, Informatica e Sistemistica - Università di Bologna
Viale Risorgimento, 2 - 40136 Bologna - ITALY
Phone: +39-051-2093001; Fax: +39-051-2093073
{pbellavista, acorradi, cgiannelli}@deis.unibo.it

Abstract

The growing availability of wireless portable devices is leveraging the diffusion of Location Based Services (LBSs) that provide service contents depending on the current position of clients, servers, and involved distributed resources. When a wide public of final users will use LBSs, two primary issues will emerge as crucial: how to guarantee the proper level of user privacy given the need to disclose, to some extent, client location information; how to effectively manage the exchange of positioning information (and of its variations) notwithstanding the high heterogeneity of connectivity technologies and device hardware/software capabilities. The paper presents the privacy-related extension of our proxy-based mobile agent middleware to support personalized service provisioning to Wi-Fi portable devices. In particular, our middleware prototype adopts a two-level proxy-based architecture to provide LBSs with middleware-mediated effective access to location data, which are exposed at the proper level of granularity depending on privacy/efficiency requirements dynamically negotiated between clients and LBSs.

1 Introduction

The growing availability of powerful mobile devices with relatively high wireless bandwidth, e.g., via UMTS, IEEE 802.11, and Bluetooth 2.0 connectivity, is going to leverage the widespread diffusion of Location Based Services (LBSs). LBSs significantly increase the expressivity of traditional services by provisioning service contents that depend on current user position, on mutual location of clients and server resources, and on mutual position of users in a group [1]. Some relevant research work on techniques about wireless device positioning has recently emerged, also without the need of additional hardware equipment, as in the case of Global Positioning System (GPS). For instance, some solutions extract approximated information about client positioning from signal characteristics and their time evolutions, in either cellular or Wi-Fi networks [2, 3, 4].

When LBSs will get out from research labs and will involve a wide public of final users, two primary issues will come out as essential: the former is a social issue, i.e., how to guarantee the proper level of user privacy given the need to disclose, to some extent, user location information to enable LBSs; the latter is the technical challenge of how to effectively manage the exchange of positioning information (and of its variations), also by considering the high heterogeneity of currently used connectivity technologies and device hardware/software.

In our previous LBS-related work, we have motivated the need of middleware solutions to support personalized service provisioning to portable devices [5]. Middleware components can locally mediate service access and dynamically adapt service results to device characteristics, location, and runtime resource availability [6-9]. In particular, we have developed a middleware that predicts Wi-Fi inter-cell movements, based only on lightweight and client-side signal monitoring. That middleware is capable of supporting seamless handover in IEEE 802.11 networks, by maintaining service continuity when clients roam between wireless cells at provision time [5].

The original contribution of this paper is the extension of that middleware with functions for efficient handling of location information with privacy requirements. In particular, our extended middleware prototype can provide LBSs with middleware-mediated access to location data. Location data are exposed at the most proper level of precision (*location granularity*) depending on the applicable privacy and efficiency requirements, dynamically negotiated between clients and LBSs.

By focusing on middleware efficiency, let us point out that usual CPU/memory limitations of client devices suggest deploying middleware components over the fixed network, possibly in proximity of the served mobile clients, while portable devices should only host thin clients, loaded by need and automatically discarded after service. By considering privacy, let us stress that the choice of proper location granularity depends not only on user privacy demands and LBS application-logic requirements, but also on the differentiated precision achievable by different positioning techniques.

2 Related Work

Some interesting work in the literature already handles some partial aspects related to client-based positioning and user location privacy. [10] proposes location obfuscation: LBSs can only access a uniformly downscaled location information (with lower precision and lower geographical granularity) instead of exact client positions. [11] realizes user anonymity through a Mediator Agent, i.e., either a user-controlled or a trusted-third-party mediator that separates mobile terminals from service providers on the fixed network. Other work [12, 13] specifically focuses on providing users with strong anonymity, independently of position and movements, by concentrating on real-time Internet services and on malicious attackers capable of observing any communication link.

The primary goal of our proposal is to perform location management in a simple and lightweight manner, by providing a partial form of anonymity, suitable and sufficient for most LBSs. However, let us rapidly observe that solutions like [12, 13] are complementary to our proposal and can be integrated with it.

To the best of our knowledge, there are no research activities proposing middleware solutions to simultaneously face location privacy issues and efficient management of positioning data, by exchanging location information with differentiated granularity levels.

3 A Middleware Solution for Effective and Privacy-enabled Location Management

A first element to consider in LBSs with user location privacy requirements is to determine who is in charge of positioning. We claim that privacy-enabled LBSs are simpler to develop and deploy when clients (or trusted positioning servers in client localities) are the only entities fully aware of their location and are responsible for communicating it to LBSs. Therefore, we focus our work on localization solutions where clients estimate their positions either in a completely autonomous decentralized way or via local trusted servers close to them, such as in the case of the Ekahau Positioning Engine (EPE) [3, 4].

Another relevant factor to improve location privacy is to disclose positioning information at the proper granularity, i.e., with the minimum precision needed to satisfy the LBS provisioning requirements. Location representation may be either symbolic or geometric [14]. We have adopted a simple symbolic representation model with variable granularity levels. Table 1 exemplifies possible client positions with different granularity: depending on the precision required by an LBS, the useful position information for a client device may be either α (granularity=3) or β (granularity=4). In particular, if an LBS requires granularity *x*, even if the client can obtain its position with granularity *y*>*x*, the client should only divulge its localization with granularity *x*, i.e., with the minimum possible precision. For instance, locations from 1 to 7 may represent successive positions in a user path with granularity=6, while LBS-required granularity could be lower; that example will be exploited in the experimental result section in the following.

Location ID	Granularity	Location information
α	3	Italy, Tuscany, Florence
β	4	Italy, Emilia, Bologna, EngFaculty
l1	6	Italy, Emilia, Bologna, EngFaculty, Lab2, PhDZone
l2	6	Italy, Emilia, Bologna, EngFaculty, Lab2, Office
l3	6	Italy, Emilia, Bologna, EngFaculty, Lab2, StudZone
l4	6	Italy, Emilia, Bologna, EngFaculty, CommLab, BTStation
l5	6	Italy, Emilia, Bologna, EngFaculty, CommLab, Admin
l6	6	Italy, Emilia, Bologna, MathFaculty, Floor1, Room12
l7	6	Italy, Emilia, Bologna, MathFaculty, Floor1, Room5

Table 1. Our Granularity-differentiated Symbolic Location Model.

The proper location granularity should be negotiated, for any client-LBS pair, depending on both user preferences and LBS requirements. Our primary solution guideline is to adopt middleware-level proxies, which execute on the fixed network in client proximity, for granularity negotiation and location obfuscation on behalf of their associated clients. Proxies can alleviate resource-limited devices from location management operations and, most important, can enforce location privacy requirements with no impact on client application logic. In addition, by choosing appropriate granularity, they can significantly reduce the network traffic exchanged due to position modifications. For instance, in the case of LBSs with results to update at location changes with granularity=4, our proxies can inform LBSs about the movements of their associated clients only when changing faculty buildings and not when entering new rooms in the same building.

We have identified two different proxy-based architectures for privacy-enhanced efficient management of location data: with proxies only on client side (*CProxies*) and with both client- and server-sided proxies (*CProxies* and *SProxies*). Middleware solutions based on CProxies only are simpler to deploy since there is no need for server-sided support infrastructure; however, the achievable privacy is intrinsically limited by the fact that LBSs could identify clients by tracking associated proxies (the countermeasure is overloading clients by forcing them to continuously change exploited proxy instances). A double level of middleware proxies can achieve greater privacy and anonymity: when using CProxies and SProxies, the middleware can mediate any communication between clients and LBSs; in addition, CProxies/SProxies can be the only entities to know the specific privacy preferences of their associated clients/LBSs.

Figure 1 depicts the architecture of our middleware solution, based on two level of proxies (CProxies and SProxies). Each client device hosts the execution of a lightweight Mobile Node Stub (MNStub) and is assisted by one CProxy running on the fixed network, in the same network locality of the Wi-Fi access point that currently provides client connectivity. MNStub works to achieve seamless roaming by prefetching data when its MN is expected to perform a handover, by alleviating problems due to temporary network unavailability. CProxy is a mobile agent that migrates by following MN changes of access points, thus maintaining colocality with the served MNStub (the dotted line in the figure represents wireless communications between MNStub and its CProxy). CProxy colocality with its associated MNStub notwithstanding client roaming permits to reduce network latency and overhead during service provisioning. CProxy is in charge of client-side privacy maintenance and granularity negotiation. Finally, each deployed LBS interworks with one server-sided SProxy that transparently enhances its LBS with privacy negotiation functionality.

CProxy and SProxy exploit a secure SSL communication channel to negotiate the appropriate location granularity and to exchange the needed position modifications. Note that, in our middleware solution, users should exclusively trust their CProxies and the same applies to LBSs with their SProxies, thus requiring the establishment of only local trust relationships with middleware components.

To better detail how our middleware works, MN informs CProxy about its privacy level requirements, i.e., the maximum granularity at which it agrees on exposing its location. CProxy possibly decreases exposed location granularity in the case that LBS requirements are compatible (lower than privacy-related ones). Then, CProxy invokes service execution to SProxy, which finally contacts the actual LBS component. Let us note that each proxy level can contribute to reduce network traffic to preserve wireless link bandwidth: CProxy does not communicate location changes not relevant for LBS granularity requirements; similarly, SProxy does not notify service result variations with finer granularity than client privacy requirements in the case of publish/subscribe model of interaction.

Our flexible middleware architecture can simply enable alternative solutions for location granularity filtering: for instance, CProxy could forward MN location at maximum precision, together with user privacy requirements, to SProxy; SProxy could be the only responsible for granularity reduction, by increasing location update traffic but potentially enlarging the usability of location data if SProxy serves different LBS components with differentiated granularity requirements in its locality. It is also possible to downscale location granularity only to respect LBS desiderata, thus obtaining only a form of user anonymity and not location obfuscation.

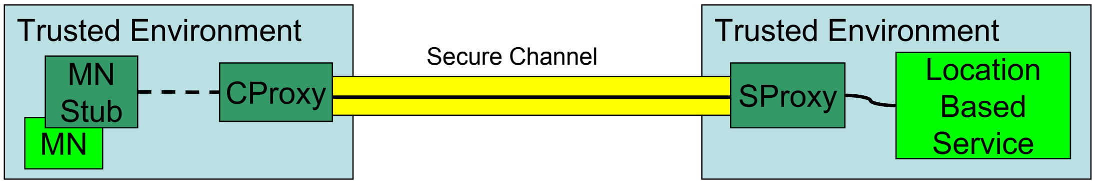

Figure 1. The Architecture of our Proxy-based Middleware for Location Management.

Let us finally point out that, to achieve stronger user anonymity, our middleware can be easily extended by implementing either Onion or Mix mechanisms in CProxy [12, 13]. Another possibility is to associate one CProxy with all MNs served by one access point, thus mixing the requests of different users to the same LBS but introducing a potential performance bottleneck.

4 Experimental Results

To quantitatively evaluate the performance of our two-level proxy-based middleware, we have considered the case of a simple LBS that provides clients with the list of all resources available in their locality. Suppose that clients move from l1 to l7 (see Table 1) at provision time; the positioning system can estimate user location with granularity 6, the LBS requests granularity 5, and users desire to disclose their position with granularity 4; each location has the same number of resources (10) and resource descriptions have all the same size (3.3KB each).

To identify the isolated overhead contribution due to our middleware, we have decided to consider four possible working modes:

- *PrivacyOff* – the proposed middleware is not used, i.e., MNs request service provisioning directly to LBS;
- *Anonymous* - the middleware does not perform any location granularity downscaling;
- *Server Side Privacy Management (SSPM)* – the middleware SProxy performs granularity downscaling and resource filtering;
- *Client Side Privacy Management (CSPM)* - the same as SSPM but with CProxy in charge of downscaling and filtering.

The experimental results reported in the following are specific for the above scenario. However, similar results can be obtained in any deployment environment where potential positioning granularity is greater than LBS requirements (that condition applies to most LBSs, such as in city/museum guide assistants based on Wi-Fi positioning estimation [5]).

From the deployment point of view, in the experimental testbed CProxy, SProxy, and LBS run on different nodes with different available bandwidths. In particular, MN and CProxy communicate through a wireless link with limited bandwidth of 500Kbit/s; CProxy and SProxy can exploit a 2Mbit/s wired connection to mimic geographic distribution; the bandwidth between SProxy and LBS is 8Mbit/s. We have deployed our middleware components on Pentium4 2.8GHz desktops with 1GB RAM connected to the same 100 Mbit/s LAN; differentiated bandwidths are obtained via emulation.

Since LBS granularity is greater than user privacy level, when the middleware performs location downscaling for privacy requirements, LBS tends to send more service results than needed. For instance, when a client is in l1, LBS provides all the objects in EngFaculty and not only Lab2 objects. To reduce useless traffic and service response time, the middleware tailors LBS results accordingly to actual user location, independently of enforced privacy. Moreover, when a client moves from l1 to l2/l3, or from l4 to l5, or from l6 to l7, the middleware does not propagate new MN service requests (*request dropping*) since it is aware that no location variation of interest for LBS has occurred.

We have identified one synthetic performance indicator, *Cumulative Service Time* (CST), defined as the sum of all service response times experienced in the current and already visited locations. For instance, CST at l3 is the sum of response times measured in l1, l2, and l3. The CST indicator is relevant to understand middleware performance while used in the typical usage scenario of clients continuously accessing their LBS while moving along a path, where it is sometimes possible to reduce response time and network traffic thanks to request dropping.

Figure 2 reports CST for the different middleware working modes (not including the delay for SSL channel instantiation between CProxy and SProxy - about 547ms – which has to be sustained only once at CProxy startup).

In the case of PrivacyOff, the MN client directly contacts LBS and performs service requests anytime MN changes location, regardless LBS granularity. Therefore, CST exhibits an almost linear growth when increasing location ID. When the working mode is Anonymous, instead, service response time in each location greatly depends on current and already visited locations: the two proxies introduce a non-negligible delay when MN does its first request to LBS (l1) due to request/response propagation through our middleware components (1438ms instead of 813ms for PrivacyOff). However, successive responses, e.g., the ones from l2 and l3, are prompter (about 200ms in place of more than 700ms) because CProxy can also perform request dropping.

In all cases where location management has the twofold goal of privacy enforcement and traffic reduction, the most interesting middleware working modes are SSPM and CSPM. In both modes, LBS sends more objects than strictly needed because our middleware provides it with downscaled client location. In SSPM mode, it is the SProxy that performs service tailoring and unfiltered data only overload the LBS-to-SProxy link: CST at l7 is only 140ms higher than in Anonymous mode. In CSPM mode, since CProxy is in charge of service tailoring, unfiltered results also overload the SProxy-to-CProxy communication link, by introducing additional delay. However, actual user location is visible only at the client side in CSPM, thus achieving a stronger and more secure level of privacy.

Finally, let us note that the middleware performance can be further increased in any deployment case where i) the difference between location granularity and LBS requirements is large, thus enabling frequent request dropping at CProxy, and ii) the caching of either client location data or service results makes sense (for instance, result caching at SProxy when it serves multiple clients and at CProxy when it deals with successive requests of the same user).

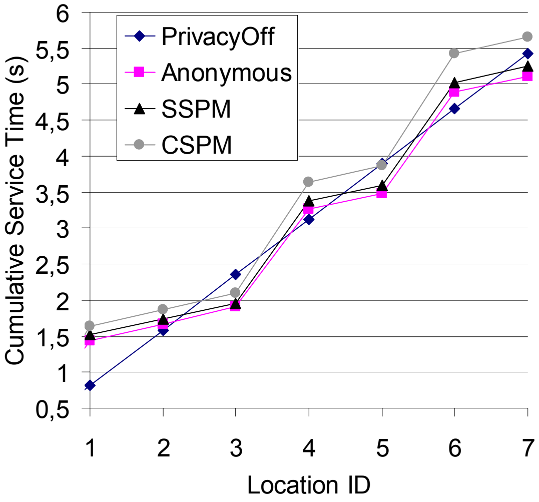

Figure 2. CST for different working modes.

5 Conclusive Remarks

The spreading of LBSs is drawing attention to the need for support infrastructures to effectively manage location information and to disclose it properly depending on both privacy and application requirements. We claim the need for middleware-level solutions for privacy-enabled location management in order to decouple the responsibility of location maintenance/processing from service-side application components, also to simplify the design and implementation of LBSs. Our middleware prototype demonstrates that it is possible to achieve feasible performance even without sacrificing portability, by adopting decentralized proxy-based solutions capable of reducing network traffic via proper management of different location granularities.

Acknowledgements

Work supported by MIUR FIRB WEB-MINDS and CNR Strategic IS-MANET Projects.

References

[1] G. Chen, D. Kotz, "A Survey of Context-Aware Mobile Computing Research", *Dartmouth College Technical Report TR2000-381*, http://www.cs.dartmouth.edu/reports/

[2] C. Drane, M. Macnaughtan, C. Scott, “Positioning GSM Telephones”, *IEEE Communications*, Vol. 36, No. 4, Apr. 1998.

[3] Ekahau Inc. - *Ekahau Positioning Engine 2.0, Technology White Paper*, 2002.

[4] W.N. Schilit, et al., “Ubiquitous Location-Aware Computing and the Place Lab Initiative”, *1st ACM Int. Workshop Wireless Mobile Applications and Services on WLAN*, USA, Sep. 2003.

[5] P. Bellavista, A. Corradi, C. Giannelli, “Mobile Proxies for Proactive Buffering in Wireless Internet Multimedia Streaming”, *Int. Workshop Services and Infrastructure for the Ubiquitous and Mobile Internet (SIUMI)*, USA, June 2005.

[6] W. Stallings, *Wireless Communications and Networks*, Pearson Education, Aug. 2001.

[7] P. Ramanathan, et al., “Dynamic Resource Allocation Schemes during Handoff for Mobile Multimedia Wireless Networks”, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 7, July 1999.

[8] S. Saha, M. Jamtgaard, J. Villasenor, “Bringing the Wireless Internet to Mobile Devices”, *IEEE Computer*, Vol. 34, No. 6, June 2001.

[9] K. Curran, G. Parr, “A Middleware Architecture for Streaming Media over IP Networks to Mobile Devices”, *IEEE Int. Conf. Wireless Communications and Networking*, Mar. 2003.

[10] A. Quigley. B. Ward, C. Ottrey, D. Cutting, R. Kummerfeld, “BlueStar, a Privacy Centric Location Aware System”, *IEEE Int. Symp. Position, Location and Navigation*, USA, Apr. 2004.

[11] L. Titkov, et al., “Privacy Conscious Brokering in Personalized Location-Aware Applications”, *Int. Conf. Autonomous Agents and Multi Agent Systems (AAMAS)*, Australia, July 2003.

[12] R. Dingledine, N. Mathewson, P. Syverson, "Tor: The Second-Generation Onion Router", in *13th USENIX Security Symposium*, Aug. 2004.

[13] O. Berthold, H. Federrath, M. Kohntopp, Project ”Anonymity and Unobservability in the Internet”, *Int. Workshop on Freedom and Privacy by Design*, Canada, Apr. 2000.

[14] J. Hightower, G. Borriello, “Location Systems for Ubiquitous Computing”, *IEEE Computer*, Vol. 34, No. 8, Aug. 2001.