AD LOC: Location-based Infrastructure-free Annotation

Derek J. Corbett and Daniel Cutting

School of Information Technologies
University of Sydney, Australia
{dcorbett,dcutting}@it.usyd.edu.au

ABSTRACT

We introduce AD LOC, a system for mobile users to collaboratively tie persistent virtual notes to physical locations without the need for any servers embedded in the environment or accessed via the Internet. Instead, all notes are proactively cached solely on the mobile devices of passing participants and served up to others in the vicinity via ad hoc wireless protocols like WiFi. By making use of any of the increasingly ubiquitous positioning technologies, such as GPS, devices attempt to ensure notes remain cached at the physical locations they were published. Through simulation we demonstrate AD LOC's behaviour under different scenarios and caching strategies. Our results show that despite the unpredictable and inconstant storage medium, surprisingly few participants are required for remarkably robust note persistence. Furthermore, the number of transmissions between devices needed to realise the caching strategies is relatively low and increases linearly with the number of participants.

Keywords: Location-aware services, pervasive computing, physical annotation.

1 INTRODUCTION

People are using mobile devices for an increasingly diverse collection of activities including gaming, storing and sharing contact details and schedules, purchasing products from vending machines, and even social organisation and coordination [1]. These devices are quickly becoming an important part of our lives, accompanying us everywhere from home and work, to movies, bars and bus stops.

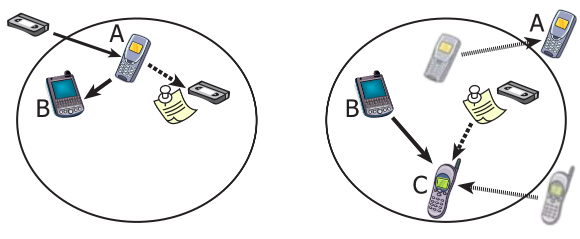
Location-based services such as finding cafés in foreign cities or locating nearby train stations are appearing [2], [3], as are systems for annotating the physical environment with virtual "post-it notes" [4]–[6]. Typically these require either server infrastructure deployed by providers into the physical environment, or mobile connectivity to the Internet available through expensive contracts.

With the technological convergence of devices, a plethora of features are now offered in the latest mobile phones including cameras, high capacity storage cards, keyboards, large screens and wireless capabilities like WiFi and Bluetooth which can be used for ad hoc connections to nearby devices without the cost of a service provider. Furthermore, technologies such as GPS and PlaceLab [7] are becoming more pervasive, allowing ever finer detection of a mobile user's position.

This paper introduces AD LOC, a platform that enables users to tie persistent virtual notes to physical locations without the need for embedded infrastructure in the environment or access to the Internet. We outline a usage scenario in section 1.1 and detail the model in sections 2 and 3. In section 4 we construct an Internet object cache upon the basic design allowing users to save on mobile Internet costs by first probing AD LOC for content. Section 5 contains our evaluation of the system and highlights its advantages and disadvantages. We conclude with related work in section 6 and future work in section 7.

1.1 Motivating Scenario

Imagine visiting a cinema and trying to decide which movie to see. You could use your mobile device to download the trailers and reviews of the films that are currently showing, but it's expensive to do so. Instead you first query AD LOC to see if anybody has published copies at the cinema's location. Unfortunately nobody has yet downloaded the trailer you're interested in. However you did find some reviews that people have published and it comes highly recommended. Your interest further piqued, you download the trailer of the movie from the internet and publish it as an AD LOC note for the benefit of future movie-goers. Based on the reviews and the trailer you decide to watch the movie, afterwards you vote for the AD LOC reviews that were the most helpful to you and add your own. As you leave the cinema, the notes you have published remain, cached on the devices of patrons arriving for the next session and other devices in the vicinity.



(a) **A** publishes notes which are cached on bystander **B**.

(b) **A** leaves. **C** arrives and retrieves all notes at this location.

Figure 1: AD LOC devices cooperatively publishing, caching and retrieving notes at a physical location.

Figure 1(a) shows user **A** downloading a movie trailer and publishing it along with her opinion of the movie in the foyer of her local cinema. Another user **B** is participating in the system and caches copies of the notes. When **A** leaves and user **C** arrives in figure 1(b), he is able to retrieve the notes from **B**.

2 AD LOC

AD LOC is an infrastructure-free, localised, persistent and asynchronous platform for collaboratively annotating the physical environment with "notes".

- Localised: notes are specific to physical locations.
- Persistent: notes remain in the environment.
- Asynchronous: notes are published and retrieved without requiring the concurrent presence of both publishers and consumers.
- Collaborative: anyone can publish or read any notes.
- Infrastructure-free: no servers or Internet connections.

AD LOC does not rely on any systems or services specifically installed at locations where users would like to publish notes. Not all devices are expected to have access to the Internet, so we cannot make use of Internet-based servers to store the notes. Besides, storing the content on servers embedded either in the environment or on the Internet may make them susceptible to legal censorship or administrative bias. Instead, all notes stored in the AD LOC system are stored on the constituent mobile devices.

As users with mobile devices pass through locations, they automatically accrete notes that have been published there, and make them available to other users that wish to find and read them. As users leave, their cached notes can be transferred to other devices remaining at the location. AD LOC acts as a platform for storing and retrieving data over an inherently very unreliable and dynamic substrate.

Each device is assumed to have the following necessary components: some form of localised, wireless communication technology such as Bluetooth or WiFi; a persistent storage cache of some size; and some way of sensing its location, be this GPS, PlaceLab [7], or any other means. We place no restrictions on how location is determined, just that it must be possible to convert it into global latitude and longitude coordinates.

2.1 Publishing

A *note* is the unit of annotation in the AD LOC system. It is composed of an identifier, a timestamp, a location described by an "area of relevance" or AOR (see section 2.3), a descriptive subject, and a MIME data component :

$$< id, timestamp, AOR, subject, data >$$

A note's ID is constructed by creating a digest of the entire subject and data segments. Notes are conceptually stored at the user's location at the time of publication where they persist for others to view, even after the publisher has left the area. A note is physically distributed to other AD LOC devices by caching them according to the cache replication policy (see section 3).

2.2 Querying

A query is made by requesting notes matching a given template. The template can be empty, for a general *location query*, or specify certain constraints regarding the notes to be returned. For example, a timestamp can be given to ensure only recently published notes are returned, or a particular subject field can be required.

If a query can potentially be resolved locally the device first examines its own cache. If no match is found, or the template is empty, the query is broadcast to all AD LOC devices within range. Each checks its cache and broadcasts a reply if matches are found.

Notes are considered matches if and only if they match the template and are relevant to the location of the query device or replying device. By including notes that are relevant to the location of the replying device, the system ensures that notes can be propagated to all devices that may have an interest in them. Section 2.3 details this mechanism.

The list of notes resulting from a query can be displayed to the user according to some ranking function (see section 7). If only a single note matches a query, it can be automatically displayed to the user.

2.3 Area of Relevance

Notes are relevant to specific locations. However some notes are relevant to very small locations, such as a storefront, while others are relevant to much larger areas, e.g., an entire football ground. Each note is thus associated with an *area of relevance* (AOR) defining this region.

The AOR of a note is not fixed. It begins small when a note is first published and can grow if more distant users also find the note to be relevant. In this way, the AOR of a note is able to approximate the size and position of the conceptual location originally intended by the publisher. The act of querying for a specific note or selecting and reading a note resulting from a general location query implicitly indicates it to be relevant to the user's current location, and hence grows the AOR.

The AOR is used to aid caching policies (see section 3) and when responding to queries from other devices. A reply to a location query includes all notes that match the given template and that have an AOR including themselves or the query device.

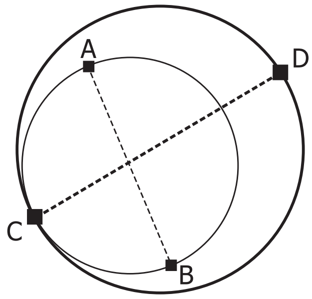


Figure 2: An area of relevance growing from AB to CD.

A note's area of relevance is modeled by two coordinates that bisect a circle containing all points where users have indicated the note to be relevant. When a new note is published, its AOR is initially centered at the publisher with a small, arbitrary radius. When another device outside the area receives the note and marks it relevant, the area is increased to minimally include both the original area and the new point. This results in an area which always encompasses all points that have ever been a part of it. Figure 2 shows an area described by points A and B being increased to encompass a new point D (and henceforth being described by points C and D).

Since notes are cached on many devices, it is possible for some caches to retain stale AORs. We remedy this by stating that an AOR can never shrink in size. Any device that learns of a conflicting AOR for a note in its cache simply selects the largest as the "true" value.

3 NOTE CACHING

The aim of AD LOC is to tie virtual notes to physical locations by physically caching them on mobile devices that are continually passing through. Since such a storage medium is so volatile, a good caching policy is of paramount importance for note availability.

3.1 Device Density

Clearly, in some situations it is impossible for AD LOC to guarantee note availability. For example, if all mobile devices move away from a location, there is no way for previously published AD LOC content to be available to devices that subsequently arrive. Similarly, if users sometimes refuse to cache notes or respond to queries, availability will suffer.

However, by assuming a uniform distribution of cooperative mobile users in a square area, it is possible to calculate the average number of users required for connected networks of various degrees. This in turn allows an estimate of the number of participants needed for AD LOC to effectively store notes (since all caching is achieved by broadcasting notes to neighbouring devices). It is worth noting that users in real scenarios are unlikely to be uniformly distributed; rather they would be expected to congregate in more popular locations. A uniform distribution does however provide an important minimum estimate for the number of necessary users. We adapt our equation from [8]:

$$n = \frac{-3dl^4}{(12\pi - 16 - 4\pi\sqrt{3})r^4 + 8lr^3 - 3\pi l^2 r^2} \qquad (1)$$

where n is the number of necessary users, r is the broadcast radius, l is the side length of the square area, and d is the average number of neighbours desired for each user. Our evaluation in section 5 uses square areas 500 metres on a side and assumes a broadcast radius of 82m. With these parameters, at least 14 users are required for each to have an average of one neighbour, and 28 for an average of two neighbours. Thus, we can expect AD LOC to perform poorly with less than 14 users in our experiments, since notes will not often have an opportunity to be replicated before devices move away from locations. With at least 28 users however, AD LOC should exhibit good storage properties, leading to few lost notes.

3.2 Cache Replication Policy

In general, the goal of the cache replication policy is to maximize the availability of notes and minimis the number of messages between devices. In the basic caching policy, which underlies all other policies, all messages between devices (including responses to queries) are broadcast such that any device overhearing the transmission of a note can cache it, even if it is not specifically intended to be a part of the exchange. Beyond this, there are two main decisions to make: when to broadcast notes, and which notes to broadcast. We identify several possible policies below.

3.2.1 Publish

The publish caching policy immediately publishes any new note via broadcast when it is generated, for caching on neighbouring devices. The advantage is that the note is highly replicated at the location very quickly.

3.2.2 Periodic

In the periodic policy, each device periodically broadcasts some or all of their notes for caching by other devices. The subset to be broadcast is determined by the time at which they were cached; notes that have been recently received are less likely to be broadcast because they are likely to have also been recently overheard by neighbours. For locations with low user turnover, periodic broadcasting results in many unnecessary transmissions, and for high turnover the period must be frequent enough to ensure notes are not lost.

3.2.3 Location-aware Periodic

The location-aware periodic caching policy periodically broadcasts the subset of its cached notes which are relevant to the area within its broadcast radius. Any note that has an area of relevance that is at least partially within the broadcast radius of the device is selected to be sent. As with the periodic policy, the set of notes is pruned such that more recently received notes are not broadcast.

3.2.4 All

This caching policy combines both the publish and the location-aware periodic caching policies (as well as the default basic caching policy). The device will broadcast a note upon publication and also periodically broadcast location-relevant notes for caching.

4 THE AD LOC INTERNET CACHE

Section 2 described the basic AD LOC platform. As it stands, the system can be used to store personal opinions, reviews and points of interest in the environment and make them available to others. In this section, we present an extension that enables AD LOC to also function as an Internet object cache, saving users the cost of downloading content over expensive mobile Internet connections.

For some locations, there will be relevant material available on the Internet, e.g., trailers for movies are useful for people at cinemas selecting which movie to see. To use AD LOC as an Internet object cache, a user can simply download the object over their mobile Internet connection (if they have one) and republish it as a note for others to find (who either don't have an Internet connection or would prefer not to use it).

The only difference between an Internet object and any other note is that the same object can be published separately by many users. It is preferable to treat them as replicas of one another rather than distinct notes in order to save on storage and broadcast costs. This is trivially achieved by using the URL of the object as the subject of the note, which results in all notes published for a particular Internet object having the same identifier regardless of which devices publish it. Once notes have been published with the same identifiers, they are assumed to be replicas and will not be cached more than once on any particular device.

To use the Internet object cache, a user performs a query with the URL of the object they are trying to find as the subject. If the object has already been published in AD LOC it will be returned to them. Otherwise, they will need to download it themselves and publish it for others.

5 EVALUATION

We evaluated the behaviour of AD LOC according to two different scenarios by varying the number of users and measuring the availability of published notes, and the amount of traffic introduced by caching and requests. All experiments had a similar basic configuration. We constructed an OMNeT++ simulation using the Mobility Framework to simulate users with WiFi-enabled mobile devices roaming a square area of 500x500 metres. Each user had a broadcast range of 82 metres using a path loss alpha of 2.5, and followed a random way–point model of movement with a speed of 1m/s and no pause time. The first scenario (A) simulated several city blocks, with 400 small locations of radius 10m randomly placed to represent points of interest throughout the city. The second scenario (B) comprised 4 large locations of radius 100m, to represent 4 large points of interest. In each scenario the area covered by the locations was approximately half of the total network area. A user would expect to be in one or more of these locations on average half of the time. All periodic cache replication policies employ a 20s period. To avoid a broadcast storm these periods are randomly offset. Notes are not rebroadcast until at least 100s after they are received.

Each simulation ran for 3,000 simulated seconds, the results of the first 1,000 seconds lead in time were not included in the results in order to let them stabilise. All results presented here are the average of 10 runs for each experiment, and the error bars represent a 90% confidence interval.

Each device in the simulation flushed its entire note cache based on a normal distribution centred at 500 seconds with a standard deviation of 100 seconds. On average a device erased its cache 6 times in each simulation run. This was done to simulate a turbulent environment, where users either leave the network entirely or choose to flush their device's cache. If a device were to never flush its cache it would eventually cache all available notes published in AD LOC (dependent upon its mobility).

We investigated both the effectiveness of AD LOC for publishing and storing content created by users *in situ* and its ability to cache Internet objects for limiting the use of Internet connections. In these experiments the essential difference between *Internet* objects and *user-created* content notes is that user-created notes are unique, and therefore must be discovered whereas Internet objects can be specifically requested by URL.

5.1 Metrics

The availability of notes that have been published at a location is of primary importance in AD LOC. In these experiments we calculated availability by keeping track of the notes that had been published at each location, and comparing this to what was obtainable by a user when they performed queries (either already in their cache or returned by other users). We call this ratio the Recall metric. It is a value between 0 and 1 derived from equation 2.

$$Recall = \frac{Relevant\ Notes\ Found\ on\ Query}{Total\ Published\ Relevant\ Notes} \quad (2)$$

Since AD LOC relies on proactively caching content over local devices, there is a potential overhead in terms of the number of messages the average user must send and receive compared to the ideal case of only querying Internet-based servers or servers embedded in the environment. We calculated this overhead by finding the Traffic Overhead Ratio (TOR), which is the ratio between the number of messages a user sends as part of AD LOC (including queries, replies, and any other broadcast messages) and the number they would need to send if using only a client/server architecture (i.e., only the number of queries issued) as shown in equation 3.

$$TOR = \frac{Total\ Packets\ Sent}{Total\ Queries\ Made} \quad (3)$$

Ideally, AD LOC should exhibit a high recall and low TOR, although we expect the TOR to increase as recall increases (since AD LOC trades Internet queries for local queries and caching). Recall should increase as the number of users increases, since more content should be cached and be more readily available. The TOR should also rise, since the average user will be required to cache notes and respond to queries increasingly more often than they perform their own queries.

AD LOC can be used to cache Internet data. A key metric to evaluate its performance in this respect is the ratio of queries that can be fulfilled by AD LOC to the total number of queries made. We refer to this metric as the AD LOC Satisfied Internet Queries (ASIQ) ratio (equation 4). There are two extremes to this ratio: 1 means all queries are satisfied by AD LOC and 0 means that all queries had to be resolved by the devices' Internet connections. This metric shows how well the system is able to cache content and reduce Internet traffic. We expect the ASIQ ratio to increase as more data is cached in AD LOC. This will occur as more nodes participate in AD LOC and the longer AD LOC operates.

$$ASIQ = \frac{Queries\ Resolved\ By\ \text{AD LOC}}{Total\ Queries} \quad (4)$$

5.2 Content Note Availability

This experiment measured the general availability of content notes created by users of the AD LOC system.

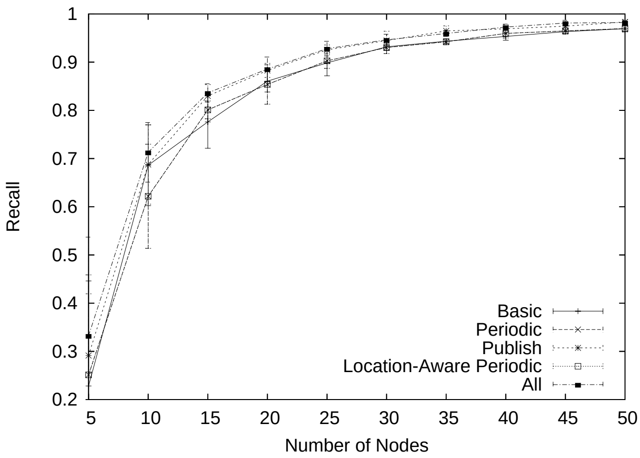


Figure 3: Recall of user-created notes in scenario A.

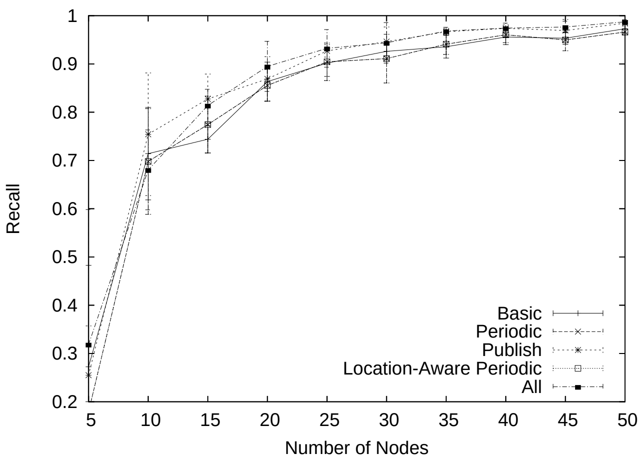


Figure 4: Recall of user-created notes in scenario B.

Figures 3 and 4 show that regardless of which AD LOC caching policy is chosen, greater than 50% recall can be expected from just 10 nodes, and just 25 are needed for 90% recall. This shows that AD LOC style broadcast based cache is resilient to flux of the users in the network and that its reliability increases with the number of participants. The smoother curve under scenario A is an indirect effect of the smaller location areas.

The traffic overhead graphs 5 and 6 show that the TOR increases linearly with the number of devices in the network. This increase is due to the increased number of replies a device must make as the density of its one-hop neighbourhood increases. These graphs show that there are two distinct bands of results. While protocols using a publish–on–creation caching policy (All and Publish) are better able to recall published content, they incur a traffic overhead penalty to do so.

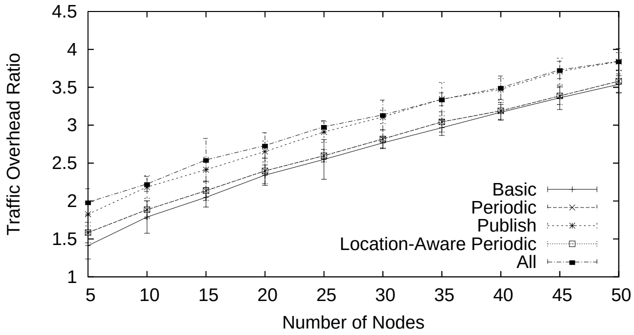


Figure 5: TOR for user-created notes in scenario A.

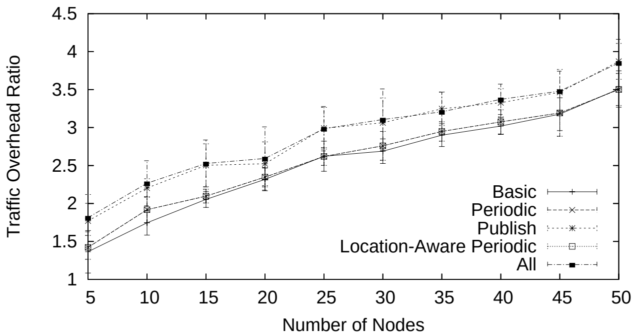


Figure 6: TOR for user-created notes in scenario B.

5.3 Internet Cache

The Internet cache feature of AD LOC (section 4) allows Internet objects to be stored at relevant locations in order to reduce users' mobile Internet costs. This experiment determined the degree to which Internet traffic was replaced by local AD LOC traffic as the number of users increased. We omitted the recall metric for this experiment as it was very similar to the results of the previous experiment.

To make the simulation tractable we modeled the Internet in the following way. Each location in each scenario had a list of data items that could be queried by a user. These data items were assumed to be the sum total of data items on the Internet relevant to the particular location. The user chose an item from this list based on a probability derived from a Zipfian distribution [9]. The total number of Internet data items was kept constant in each scenario. In scenario A there were a Poisson average of 20 data items in each of the 400 locations, while in scenario B there were a Poisson average of 2,000 data items in each of the 4 locations. This allowed a variation in the number of data items in different locations and assumed that bigger locations would have more relevant data items.

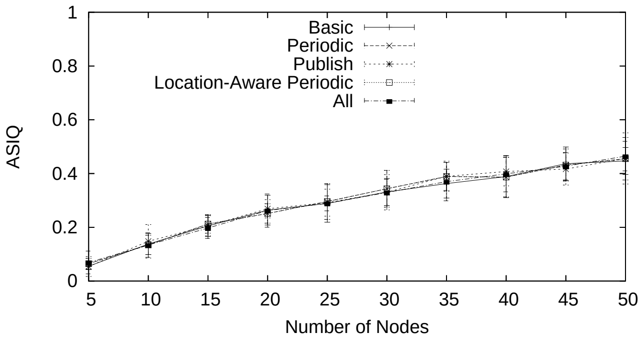


Figure 7: Internet caching in scenario A.

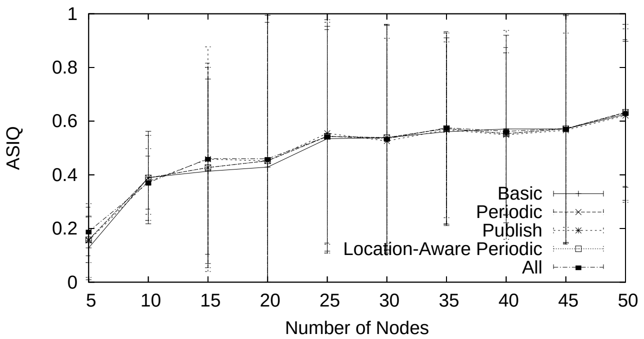


Figure 8: Internet caching in scenario B.

Figures 7 and 8 show that as the number of nodes in the network increases, the fraction of queries that can be satisfied wholly within the AD LOC system also increases. This is a direct consequence of the fact that as the number of users grows, the number of requests for Internet objects grows, and more objects are cached. Scenario A had a low variance because each location had only approximately 20 Internet objects associated with it, such that it would not take long for all of these to be requested and cached for future users. This would lead to almost all queries in a location being resolved by AD LOC after an initial phase of requests. In scenario B however, there were around 2,000 objects for each location. Many of these would be requested only once and there would be many cache misses. However, as a result of the Zipfian distribution of requests, a few objects would be requested very often, resulting in many cache hits. The combination of these effects produced a very high variance, though on average the Internet cache was able to resolve around half of the Internet object queries in scenario B.

The ability of AD LOC to recall Internet data is similar to its ability to recall user-created notes. However, it incurs less

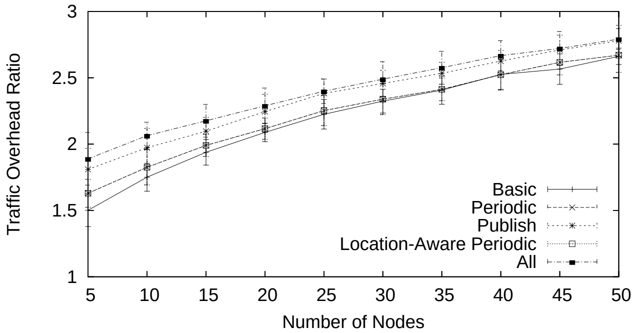


Figure 9: TOR for Internet objects in scenario A.

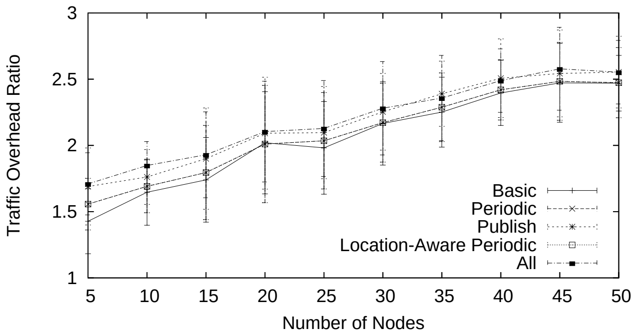


Figure 10: TOR for Internet objects in scenario B.

of a traffic overhead penalty. This is shown in figures 9 and 10. The reason for the lower overhead is the non-uniqueness of Internet data and the use of a Zipf distribution to model the probabilities of querying certain pieces of Internet data. This means that a device is more likely to get a cache hit for the data it is seeking than in the user scenario.

As AD LOC is essentially a loosely associated opportunistic network architecture, we see that even as we add more nodes to the network we never reach full recall or ASIQ. Still, AD LOC provides essentially free access to serendipitously available content with reasonably low network overhead.

6 RELATED WORK

There is increasing interest in location-based services [10] and a critical mass of users with powerful and capable mobile devices [1]. There have been many systems that allow users of mobile devices to annotate the physical environment using the "post-it note" metaphor. One of the initial systems was Stick-e notes [4], but similar ideas have existed since the dawn of ubiquitous computing with devices such as the ParcTab [11] and Chameleon [12].

Stick-e notes present the notion of using context to trigger events. Note authors attach a contextual "trigger" (often including a location) to a note that describes under which circumstances it shall be displayed to the user. More recently, this concept has been extended to aid collaboration on city and worldwide scales [5], [6], [13] with public annotation of large tracts of physical space.

Other systems have taken a more introverted approach, using location-based services only to aid individual users. Web-enhanced GPS [3] uses GPS to find a user's location, maps this via a database to a street address, then uses this as a search term to find relevant web pages. AURA [14] enables users to scan physical objects such as books or DVDs with their mobile devices to retrieve detailed information or commentary regarding these artifacts via the web.

Two recent studies [15], [16] found that location-based services accessed by mobile devices would be well-received by users in general. They identified the importance of location in informal social activity and the importance of available topical, dynamic content. They also found that people liked and were willing to contribute additional content of their own. AD LOC's ability to cache location-relevant web information and publish personal content fits this profile well. The E–graffiti [17] and CampusAware [2] projects found that many people enjoyed the social aspect of a collaborative location annotation system and contributed an unexpectedly high number of notes of advice or opinion specific to their locations.

Key to our system is the ability for devices to accurately determine their location. There are several possible ways of achieving this on mobile devices. GPS is now a very mature technology that has overcome many of its early limitations. Assisted GPS, for instance, works well both outdoors and indoors, where additional beacons have been installed to mimic satellite signals. Other nascent technologies such as PlaceLab [7] map WiFi signals to geographic locations and enable devices to triangulate their position with relatively high accuracy. For metropolitan environments where these sorts of technology are infeasible or not deployed, mobile phone towers offer some coarse-grained positioning capabilities. For example, the Place-Its project [18] used this approach to implement location-based reminders on mobile phones. They found the technique to be adequate for detecting arrival at a location, though less useful for detecting departure.

The main difference between our system and other similar systems is the storage medium. Most collaborative note systems thus far have focused on usability or the context- and location-aware aspects, and simply relegated the storage to servers. In contrast, AD LOC focuses specifically on an alternative storage approach that requires no such infrastructure to be deployed.

There are similarities in fields such as sensor networks. The family of SPIN [19] protocols diffuse information from sources to interested nodes by first forwarding metadata. Other nodes examine this metadata and request the actual data if desired. This is conceptually similar to the way notes are cached by users around a location. However, SPIN is intended to disseminate information throughout a sensor network to sensors where it is needed without the high overhead of flooding, and is not designed for location-based services. Bayou [20] permits high degrees of replication to make data available to mobile devices even after they have disconnected from primary servers and other clients. This data can be modified and reintegrated over time to a convergent state by sharing copies with other clients. Unlike AD LOC however, Bayou does not incorporate a notion of physical location.

7 CONCLUSION AND FUTURE WORK

The experiments we have conducted show that a critical mass of users is necessary for AD LOC to function in a useful manner. It is encouraging to see that this number is surprisingly small. However, the simulation models conducted were only in unobstructed free space. In a real world scenario the penetration of WiFi may be much lower than 82m. This is both a blessing and a curse. With a lower broadcast radius, there is less interference and more users can be supported. However, more users will be needed in order to reach critical mass. The mobility and note publication/query models employed were also simplified. Future work could investigate more realistic models based on real mobility traces.

A possible criticism of AD LOC is that without environmental infrastructure or Internet-based servers to store notes, the reliance on transient devices will result in much published content being permanently lost when users leave a location. However, in many cases users are likely to repeatedly return to the same locations, e.g. workplaces or lunch spots. If their caches are not erased immediately upon leaving a location, much of the content will return with them the next day as they follow their ordinary routine. Indeed, location annotation systems have found users to often access the system from the same small subset of locations [17].

Another possible cache replication policy not discussed earlier is cache-on-departure. When a user is aware they are leaving the AOR of one or more of the notes in their cache, they can broadcast them for caching on other devices. The advantage of this approach is that notes that are not cached elsewhere at a location can be replicated before they are permanently lost, without the need for additional queries. However, mobile users may potentially move through many areas of relevance in a short period, triggering excessive broadcasts and there are subtle problems with determining when a user is departing an area.

The cache replication policy tries to widely cache as many notes as possible, but real caches are of limited capacity. A cache replacement policy decides which notes to eject when the cache becomes full. For this paper, we assumed caches were large enough never to require replacement, but in practice the policy could use the age of the note, the distance from its area of relevance, and the likelihood of the user returning to the location in future. Clearly the willingness of a node to participate is also important.

AD LOC cannot absolutely guarantee a note will be persisted for others to find, due to the inherently unreliable storage medium. Locations that need stronger guarantees could deploy a proxy server that acted like any other AD LOC device. The system's main strength is that such proxies are not required, though they can augment it if desired.

We have alluded to a ranking function for presenting matching notes to users after a general location query. One possible way of doing this would be to have a voting system allowing AD LOC users to vote for notes they find, affecting their rank in the results of others' location queries. This would add an extra participatory component that could be combined with other aspects, such as the age of notes, to make the system more manageable for users in locations with a high number of users and notes.

The area of relevance technique will not work well for locations that are not approximately circular, as the area will need to encompass more and more irrelevant space to contain its extents. In the case of a river for example, the area of relevance would likely encompass vastly more area than it should. This is a known drawback of our approach, but to more closely model the actual area would require more data to be stored for each note, and we feel this is a good compromise.

There are some interesting properties of the AD LOC storage mechanism that would not occur with server-based approaches. For example, notes cannot be added to locations without physically being there which could present a problem if trying to seed the system with information. Also, notes will vanish from an area once everybody has left but may return the next day, for example, when people return with their caches intact. Curiously, it would even be possible for the same locations to harbour radically different sets of notes over the course of the day depending upon their changing social functions.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the ongoing support of the Smart Internet Technology CRC.

REFERENCES

[1] H. Rheingold. Smart mobs. *Shift: At The Frontiers of Consciousness*, pages 30–33, March 2004.

[2] Jenna Burrell, Geri K. Gay, and Kiyo Kubo. Context-aware computing: A test case. In *Proceedings of UbiComp*, Göteborg, Sweden, 2002.

[3] R. Hariharan, J. Krumm, and E. Horvitz. Web-enhanced GPS. In Thomas Strang and Claudia Linnhoff-Popien, editors, *International Workshop on Location- and Context-Awareness, LoCA*, page 95, 2005.

[4] J. Pascoe. The Stick-e Note architecture: Extending the interface beyond the user. In *Proceedings of the conference on Intelligent user interfaces*, pages 261–264, 1997.

[5] A. Dearle, G. N. C. Kirby, R. Morrison, A. McCarthy, K. Mullen, Y. Yang, R. C. H. Connor, P. Welen, and A. Wilson. Architectural support for global smart spaces. In *Lecture Notes in Computer Science 2574*, pages 153–164. Springer, 2003.

[6] S. Carter, E. Churchill, L. Denoue, J. Helfman, and L. Nelson. Digital graffiti: public annotation of multimedia content. In *CHI extended abstracts on Human factors in computing systems*, pages 1207–1210, 2004.

[7] B. N. Schilit et al. Challenge: ubiquitous location-aware computing and the "place lab" initiative. In *Proceedings of the ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, pages 29–35, 2003.

[8] D. J. Corbett, A. G. Ruzzelli, D. Everitt, and G. O'Hare. A procedure for benchmarking MAC protocols used in wireless sensor networks. Technical Report 593, School of I.T., The University of Sydney, August 2006.

[9] L. A. Adamic and B. A. Huberman. Zipfs law and the internet. *Glottometrics*, 3:143–150, 2002.

[10] M. Hazas, J. Scott, and J. Krumm. Location-aware computing comes of age. *Computer*, 37(2):95–97, February 2004.

[11] R. Want et al. The PARCTAB ubiquitous computing experiment. Technical Report CSL-95-1, Xerox Palo Alto Research Center, March 1995.

[12] G. W. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Commun. ACM*, 36(7):39–49, 1993.

[13] M. Mankins. Location linked information: A framework for emergent, location-based content deployment. Master's thesis, Massachusetts Institute of Technology, 2003.

[14] M. Smith, D. Davenport, and H. Hwa. Aura: A mobile platform for object and location annotation. Demonstration at UbiComp, October 2003.

[15] E. Kaasinen. User needs for location-aware mobile services. *Personal and Ubiquitous Computing*, 7(1):70–79, 2003.

[16] Q. Jones, S. A. Grandhi, S. Whittaker, K. Chivakula, and L. Terveen. Putting systems into place: a qualitative study of design requirements for location-aware community systems. In *Proceedings of the ACM conference on Computer supported cooperative work*, pages 202–211, 2004.

[17] Jenna Burrell and Geri Gay. E-graffiti: evaluating real-world use of a context-aware system. *Interacting with Computers*, 14(4):301–312, 2002.

[18] T. Sohn, K. Li, G. Lee, I. Smith, J. Scott, and W. Griswold. Place-its: Location-based reminders on mobile phones. Technical Report CS2005-0820, University of California, San Diego, March 2005.

[19] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In *International Conference on Mobile Computing and Networks*, pages 189–199, 2001.

[20] A. J. Demers, K. Petersen, M. J. Spreitzer, D. B. Terry, M. M. Theimer, and B. B. Welch. The Bayou architecture: Support for data sharing among mobile users. In *Proceedings IEEE Workshop on Mobile Computing Systems & Applications*, pages 2–7, Santa Cruz, California, 8-9 1994.