Location-Based Ubiquitous Context Exchange in Mobile Environments

Stefan Forsström, Victor Kardeby, Jamie Walters, and Theo Kanter

Mid Sweden University, Sundsvall 851 70, Sweden
{stefan.forsstrom,victor.kardeby,jamie.walters,theo.kanter}@miun.se

Abstract. Context-aware applications and services require ubiquitous access to context information of users. The limited scalability of centralized servers used in the provisioning of context information mandates the search for scalable peer-to-peer protocols. Furthermore, unnecessary signaling must be avoided in large-scale context networks, when location-based services only require nodes in a certain area with which to communicate context. To this end, we propose a lightweight model for composing and maintaining unstructured location-scoped networks of peer-to-peer nodes, which gossips in order to ensure quality of service for each user. The model is implemented in a prototype application running in a mobile environment, which is evaluated with respect to real-time properties. This model can also be extended to include more context dimensions, other than location.

Keywords: Ubiquity, location-based, context exchange, mobile system.

1 Introduction

This paper investigates whether a scalable user centric self organizing context delivery system provides a better solution for location-scoped acquisition of context information in real-time. This paper proposes as such an necessary improvement to previous research [1] regarding structured dissemination of context information to context-aware applications. Scalable personal networks can provide a solid model for ad hoc context-aware applications which can be used in real-time applications. The need to derive context in real-time drives the ability to create applications that can provide real-time user interaction in response to the dynamic nature of the users context. The growth of social communities on the Internet has raised the interest in social infrastructures in support of collaborative computing both mobile and stationary. The creation of services that can utilize ad hoc frameworks include, but is not restricted to, emergency systems, warning services, social gatherings, and general information sharing.

Visitors attending a museum in a city might be able to capture a small percentage of the multitude of photo opportunities that are presented at each time. However, collaboratively they provide an extensive overview of an environment. Each visitor exists as an ubiquitous island of information and potential social

K. Pentikousis et al. (Eds.): MONAMI 2010, LNICST 68, pp. 177–187, 2011.

collaboration waiting to be exploited. Current approaches such as [2,3,4,5] use solutions that build systems around completely decentralized solutions which are implemented as general context exchange solutions, while aiming for minimal hop length towards each peers position. However these solutions still maintain a logarithmic hop count between users as the system grows in scale. Other solutions such as [6] use a centralized infrastructure, which has a constant low hop count, but does not scale well as the system increases.

The reliability of mobile communication cannot be guaranteed and subsequently, neither can dependent services. Decentralized solutions provide an alternative to this problem. Solutions such as [7] and [8] require that the mobile devices contain radios with broadcast capabilities and that there exists some indeterminate means of locating an initial neighboring user. There also exists solutions based on short range communication, such as [9] and [10] solutions, which mostly is based on multi-hop opportunistic communication via short range radios. However as mobile broadband infrastructure has near ubiquitous coverage, wide-area wireless technologies have become mainstream for providing network access. In face of these trends, the need for structuring of data dissemination in the network gains in importance, which is explored in this paper.

Therefore, there exists a need for a simple ad-hoc location-based protocol capable of solving the initial node problem as well as the radio beacon problem. In this paper we present such a model for discovering nodes and maintaining a personal peer-to-peer network enabling a platform for constructing location-based social networks and information exchange.

While we concur with [11] concerning what constitutes context information, our solution is focused on location, and employs a simple structure predicated on maintaining a single hop between a user and its interests, and remain simple enough to be deployed on resource constrained mobile devices.

2 Context Services

Context information is specific information tied to an entity, such as a user or location, which explains the surrounding conditions of that specific entity and is often acquired by using sensors systems such as GPS, or by manual input such as personal preferences. The concept of context networks has been developed to administer intelligent exchange of this context information between users. Three kinds of context networks exist today, centralized solutions, distributed solutions employing structured architectures, and distributed solutions that employ unstructured architectures.

Centralized solutions such as 3GPP's IMS Presence [12], build on using the SIP protocol, to establish sessions between end users. The SIP protocol is simple and real-time capable, but the architecture for the underlying IMS system is cumbersome and was not created for real-time context-aware applications, because of its control over the data flows and the presence architecture. Structured distributed systems such as DCXP [13] utilize a strict structure and layout of all nodes. Which is aimed towards scalability and real-time applications, but require overhead messaging to maintain the underlying DHT. In addition to this, there is unstructured distributed systems such as Gnutella [14], which employs unstructured layouts which organizes nodes into an unoptimized, but much simpler topology. Requiring much less overhead to maintain the network, but has longer hop length.

The area of context networks have been revisited many times in research, by the CONTEXT [15], SenseWeb [16] and MobiLife [17] projects. Where the CONTEXT, and Mobilife project aimed towards using a centralized approach, using the SIP protocol in similary to 3GPP IMS. SenseWeb on the other hand uses an web-service oriented approach and the SOA oriented architectures used in that area, however still using a centralized approach.

Most research has been conducted in node traversal and communication optimization, which detracts from the problems associated with running the applications on mobile and limited devices, which have highly volatile connections, long communication delays, real-time demands, and a massive user base.

2.1 Context Architectures

Context-aware applications that can be enabled by personal self organizing networks are often aimed towards location-based services, such as group conversations (among people on a train) and social networks (finding people with similar interest), and requests to enable more location-based public services (location-based tourist information). However the possibilities of services based on this approach is almost endless, ranging from simple data sharing, such as location-based instant messaging, to complex services such as matchmaking services and location-based social networking systems employing multimedia delivery services.

General peer-to-peer networks do not natively support location-based services. To enable such services, message overhead is required in the form of flood searches or random walks. This mandates the need for an underlying structure which enables the location-based services natively and is lightweight enough to be run on a resource constrained device. Location-based grouping of information have existed for some time [18]. Such solutions point out the advantage gained by having a limited search area, resulting in decreased latency and bandwidth consumption.

Structured peer-to-peer networks, such as Chord [19], are often built using distributed hash tables and these networks enable many advantages such as optimized node searches. However structured networks, also introduce higher computational requirements for each node and increased signaling for maintaining the distributed hash table, especially with increased node volatility. Unstructured peer-to-peer networks are often flat architectures and without any globally structured connections among the nodes, which enables easy access for nodes to join and leave the network. Unstructured networks gain simplicity and low computational requirements, at the expense of bandwidth. Unstructured

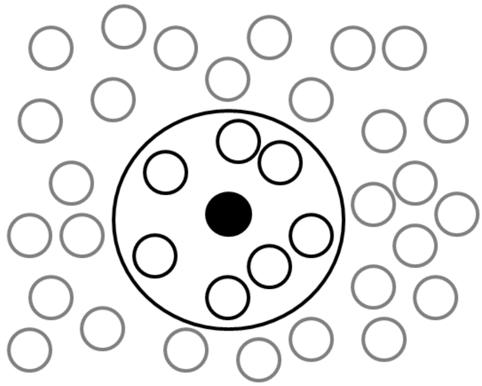

Fig. 1. Location-based personal network

peer-to-peer networks employ simple search algorithms, such as variants of flood or random walk searches. However such algorithms consume a considerable amount of network overhead even when optimized.

2.2 Context Proximity

Proximity can be defined as the physical or geographical nearness, i.e a kind of closeness or vicinity. However in regards to context, we want to utilize this concept of closeness to create the concept of context proximity. This concept means the proximity between users context, i.e. not only by location. For example, two people in two different towns, working at the same company, are not in geographical proximity to each other, but their context is in proximity, since they both working for the same employer. This proximity in regards to context has been seen in [20], although at that point it was quite simple. Although the focus in this paper is on location, the idea of location-based context proximity is used to create a form of interest between users.

3 The Model

This paper proposes a peer to peer model with limited signaling to enable a solution that natively support location-based services. The basis of the prototype comes from the fact that each node creates a personal network of nodes which are within its context proximity by gossiping with its peers. The interest area moves with the user, constantly changing when other users come within range, see Fig. 1 for an illustration of the context proximity interest area. The area can also be dynamically resized, based on the quality of service required by the applications. This quality of service can be determined on a per application basis, by choosing a radius of interest. This choice of quality of service comes from the fact that some applications might want to limit their interest area to a bus or a train on which the user is currently traveling, while some applications might require a wider interest area such as an entire city block in order to provide an adequate service.

The network is built up by having one single persistent bootstrap node, called tracker. The bootstrap procedure is simple but centralized, therefore the amount of bootstrapping should be kept to a minimum. A node bootstrap by contacting the tracker and asking for a list of nodes, which contain interesting nodes for the bootstrapping node. As the node acquires this list it can start its gossiping, and given that the other nodes can provide adequate gossip information, the node will not need to bootstrap ever again. One unfortunate disadvantage of the centralized tracker, other than the obvious scalability problem, is that the tracker is required to be updated with context information for it to provide reliable bootstrapping information. Therefore the nodes has to contact the tracker once in a while to provide it with the nodes current context.

The nodes communicate and exchanges information with other nodes via wide area Internet access. Therefore, after the bootstrap procedure, they continually gossip with each other, exchanging context information. The information exchanged in the gossip is kept limited, as the request only includes a node's current context, and the response only contains a list of other nodes which might be interesting for the requester, in similarity to the bootstrap procedure. Although the gossiping procedure is simple, it requires some management and evaluation of the acquired information.

In detail, the gossip procedure is conducted in the following manner. A node evaluates it's own list of known nodes, chooses a remote node and gossip information. The answer from the remote node will update the list of current known nodes. However, each node also has to evaluate this list of nodes, removing obsolete nodes or uninteresting ones. Such a process is required due to the volatility of context with respect to location and size of the interest areas. Therefore, all nodes continuously evaluate their list of known nodes, maintaining an updated list of other known nodes. They also continually review this own list, removing nodes which are no longer required.

The created network is dynamic, since the nodes in the interest area are constantly changing to reflect their changes in context in the real world. It is also ubiquitous as it runs on mobile devices which can contact each other using mobile Internet access. The network is also highly volatile, which can be observed when a node leaves the network and then later rejoins, then the previous interest area is almost certainly outdated and of no use. This is because the personal network composition is never in a stable state for an extended period, since the nodes context is always changing.

The key point about this model is that there exists no need to maintain overlays and expensive routing protocols which can undermine the real-time characteristics. Instead all nodes relevant to a user are kept within one hop in the overlay. This differs from other solutions, because of the distributed approach, the gossiping method, the real-time properties, and the way context is handled.

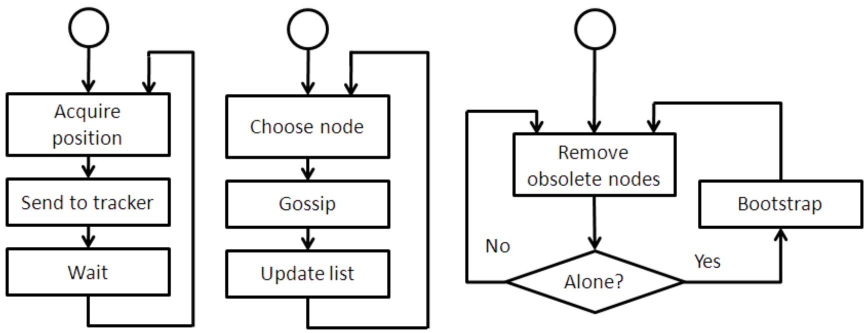

Fig. 2. Flowchart over the prototype

4 Prototype

A prototype has been developed, which resides on an implementation of the proposed model. It is however limited to only consider location as context, because of the difficulty in determining context proximity. The prototype is developed as a mobile application on the Android platform, but can also be run on a home computer. In addition to this, a prototype implementation of the tracker was implemented for home or server computers. The purpose of this application is to act as a proof-of-concept, that location-based peer-to-peer networks can run and provide suitable service even on limited mobile devices. The applications was also used to perform preliminary results and evaluations on how such a system will perform over mobile Internet and how it scales with multiple entities.

4.1 Implementation

Our prototype uses three independent algorithms, see Fig. 2. The first one, continuously updates the location of the user from available sensors, and updates it to the tracker. The second algorithm contacts a random peer chosen from a list of active nodes and queries that peer for more suitable nodes. The third algorithm manages the list of suitable nodes and removes obsolete nodes which are no longer within context proximity. It also manages the fall back situation where it will bootstrap again, which will happen if the user finds itself alone and unable to gossip.

In detail, the first algorithm acquires the GPS position of the mobile phone, and stores the location for later usage in the other algorithms, while also sending it to the tracker. The tracker will store the location, and use it when other users want to bootstrap. This algorithm is interrupt based, because of the location events created by the GPS. However the time between sending the location to the tracker can be varied, depending on the sought after quality of service. Although for the evaluation, the GPS was disabled and replaced with manual input for movement.

The second algorithm is the primary gossiping procedure. Which relies on a local list of current nodes which are in context proximity, which in the prototype is only based on physical location. The algorithm choose a random node in the list and tries to contact and gossip information with it. The request contains the current location of the user and the interest radius, which the user is interested in. As the remote node receives this information, it evaluates the request and compares to it's own local list. This process creates a new list of interesting nodes, including their location, which will be sent to the requesting user. The requesting user will receive this list and update it's own list, including the node's location. This algorithm can be performed as often as possible, depending on the quality of service required by the application.

The third algorithm relies on the local list of nodes and the current context of the user. This algorithm evaluate the context of all the nodes in the list, to determine if they still are in context proximity. As this prototype only takes into regards location as context, the evaluation for the context is quite simple. If a node is found to be outside of interest area of the user, it is removed from the list. However if the algorithm has removed all other nodes from the list, effectively making the user impossible to continue gossiping, the algorithm will contact the tracker again and bootstrap, to find new suitable nodes.

4.2 Preliminary Results

The application was deployed on Android mobile phones, but due to operator-side problems with peer-to-peer communication between mobile phones, the service runs best in emulated environments where all devices have public IP and without a firewall. Therefore the evaluation environment was setup with an active tracker with a public IP on the SUNET Internet backbone. In addition to this, a computer was setup with multiple nodes running, capable of operating up to 50 nodes at the same time. This computer operated these nodes just like a real node, and simulated their movement in the world. To this a single HTC Hero phone was added, a mobile Android device with access to the TeliaSonera 3G mobile Internet. This device runs the application as a real node, without any knowledge that the other nodes are simulated. Figure 3 shows a screenshot on how the application looks like, when running on the mobile phone. Where the node in the middle is oneself, which can be moved around on the screen in addition to the simulated movement of the other nodes.

The application also included a view with debug information with timers on how long the different actions in the algorithms have taken to be processed. These measurements was used to create an preliminary evaluation on how each of the algorithms operated. The actions which were available was bootstrap, position update, node management, and query. These evaluations are summarized in table 1, and was conducted in three different scenarios, with 10, 25, and 50 active nodes in the system. The values are measured in milliseconds using the current millis function in Java, and the values are averaged values over a small set of 10 samples each.

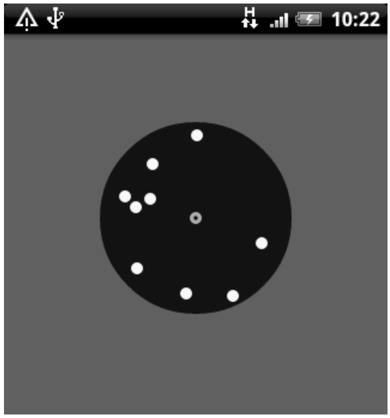

Fig. 3. Screenshot of the prototype on an Android phone

Table 1. Prototype response times

	10 nodes	**25 nodes**	**50 nodes**
Bootstrap	367ms	387ms	410ms
Position update	103ms	136ms	141ms
Node management	<1ms	<1ms	3ms
Query	310ms	313ms	335ms

Bootstrap is when a client bootstraps from the tracker. It is measured from the point when a peer begins to send a bootstrap request to the tracker, to the point in which it receives an answer, this time includes location evaluation on the tracker.

Position update is the operation performed by the clients when it sends its current location to the tracker. It is measured from the point that the node starts to send an update to the tracker, until it has received an acknowledgment that this position have been successfully updated on the tracker.

Node management is the evaluation that the client performs on the local list of nodes, which purpose is to remove obsolete nodes. It is the time it took for the client to go through the local list once.

Query is the gossiping operation, where a client asks another node for interesting nodes. It is measured from the point when the client starts to ask another node for other interesting nodes that is within vicinity, to the point in which it has received the list and updated it's own local list of interesting nodes.

4.3 Preliminary Analysis

With respect to the acquired values, one can observe that the preliminary response times is on par with normal TCP traffic over 3G Internet access, for the values which include traffic over the mobile Internet connection. But in the steps which included contacting the tracker, i.e. *Position update* and *Bootstrap*, a noticeable scalability problem of the tracker is exposed. The small increase in *Node management* is probably accounted the increase in interesting nodes, because of larger set of available nodes. While the small increase in *Query* is probably accounted the simulation environment which is running multiple peer nodes on a single physical machine, in addition to the increased set of nodes.

Because of this, a more complete simulation environment is required, taking into account both the 3G latencies of the gossiping, and a more verifiable evaluation method. Such an environment can also include even more nodes and scalability evaluations.

But in comparison to other systems such as [2,5,3,4], whose approach follows the routing principle and therefore increasing the latency with the hop count, which in those cases increase logarithmic in regards to the amount of nodes. Our system maintains a single hop, and therefore the major drawback of the 3G mobile Internet system, the delay and latency, is circumvented.

5 Conclusions

This paper presented a peer to peer based model for organizing users into dynamic, unstructured, location-based per user unique groups. The network is ubiquitous and has highly volatile connections, but is lightweight enough to be handled by each user's mobile device. The model is aimed towards context services, running on limited mobile environments with Internet access. The model is an overlay but running under different applications, therefore the model is open-ended with respect to applications.

Therefore, this model addresses the need for a solution which is capable of discovering nodes and maintaining a personal peer-to-peer network, which enables location-based social networks and information exchange in real-time. The key point of this model is that there exists no need to maintain global overlays and expensive routing protocols which can undermine the real-time characteristics. Instead all nodes that are in context proximity are kept within one hop in the overlay. This model differs from other existing solutions, because of the distribution, the gossiping method, and the way context is handled in the model. Therefore, by utilizing this approach, one can enable location-based ubiquitous context exchange in mobile environments.

The models preliminary evaluation show that the system is capable of keeping as good as possible real-time properties, and that the major drawback of the system is the centralized tracker. However, even with the tracker, it is still comparable to other related solutions.

In the future, the possibilities of enabling more complex context-ware applications on this model is going to be explored, for example enabling a multi-dimensional context-aware database, which enables the possibility of complex context queries. Future work include scalability measurements and possibilities of increasing the performance. Other interesting areas which is being explored, is to remove the tracker and see what kind of quality of service would be preserved. In addition to this, there is the possibility of arranging the members in each of the personalized networks to an alternative structure that increases the quality of service, without introducing overhead. Further future work include utilizing multiple dimensions of context to progressively build personalized networks of related sensors and context dependent entities. This will then enable more optimized real-time searching and browsing of context information sources and entities allowing more time critical applications and services to be deployed.

Acknowledgment

This research is a part of the MediaSense project, which is partially financed by the EU Regional Fund and the County Administrative Board of Västernorrland.

References

1. Kanter, T., Österberg, P., Walters, J., Kardeby, V., Forsström, S., Pettersson, S.: The mediasense framework. In: Proceedings of 4th IARIA International Conference on Digital Telecommunications (ICDT), Colmar, France, pp. 144–147 (July 2009)
2. Araújo, F., Rodrigues, L.: Geopeer: A location-aware peer-to-peer system. In: IEEE International Symposium on Network Computing and Applications, pp. 39–46 (2004)
3. Kaneko, Y., Harumoto, K., Fukumura, S., Shimojo, S., Nishio, S.: A location-based peer-to-peer network for context-aware services in a ubiquitous environment. In: The 2005 Symposium on Applications and the Internet Workshops, Saint Workshops 2005, pp. 208–211 (2005)
4. Sripanidkulchai, K., Maggs, B., Zhang, H.: Efficient content location using interest-based locality in peer-to-peer systems. In: DEF, vol. 3(3) (2002)
5. Asaduzzaman, S., von Bochmann, G.: Geop2p: An adaptive peer-to-peer overlay for efficient search and update of spatial information. In: CoRR, vol. abs/0903.3759, p. 13 (2009)
6. José Viterbo, F., Endler, M., Sacramento, V.: Discovering services with restricted location scope in ubiquitous environments. In: MPAC 2007: Proceedings of the 5th International Workshop on Middleware for Pervasive and Ad-Hoc Computing, pp. 55–60. ACM, New York (2007)
7. Li, X., Calinescu, G., Wan, P.: Distributed construction of a planar spanner and routing for ad hoc wireless networks. In: IEEE INFOCOM, vol. 3, pp. 1268–1277 (2002)
8. Gao, J., Guibas, L., Hershberger, J., Zhang, L., Zhu, A.: Geometric spanner for routing in mobile networks. In: Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, pp. 45–55. ACM, New York (2001)
9. Marsan, M., Chiasserini, C., Nucci, A., Carello, G., De Giovanni, L.: Optimizing the topology of Bluetooth wireless personal area networks. In: IEEE Infocom, vol. 2, pp. 572–579. Citeseer (2002)
10. Basagni, S., Conti, M., Giordano, S., Stojmenović, I.: Mobile ad hoc networking. Wiley-IEEE Press (2004)

11. Schmidt, A., Beigl, M., Gellersen, H.: There is more to context than location. Computers & Graphics 23(6), 893–901 (1999)
12. P. 3gp, Ts 23 228: Ip multimedia subsystem (ims); stage 2 (release 9). 3GPP (December 2009), http://www.3gpp.org/ftp/Specs/html-info/23228.htm
13. Kanter, T., Pettersson, S., Forsstrom, S., Kardeby, V., Norling, R., Walters, J., Osterberg, P.: Distributed context support for ubiquitous mobile awareness services. In: Fourth International Conference on Communications and Networking in China, ChinaCOM 2009, pp. 1–5 (August 2009)
14. Ripeanu, M.: Peer-to-peer architecture case study: Gnutella network. In: Proceedings of First International Conference on Peer-to-Peer Computing 2001, pp. 99–100 (August 2001)
15. Raz, D., Juhola, A.T., Serrat-Fernandez, J., Galis, A.: Fast and Efficient Context-Aware Services. Wiley Series on Communications Networking & Distributed Systems. John Wiley & Sons, Chichester (2006)
16. Microsoft. Senseweb, http://research.microsoft.com/en-us/projects/senseweb/
17. Klemettinen, M.: Enabling Technologies for Mobile Services: The MobiLife Book. John Wiley and Sons Ltd., Chichester (2007)
18. Li, M., Lee, W.-C., Sivasubramaniam, A.: Semantic small world: An overlay network for peer-to-peer search. In: IEEE International Conference on Network Protocols, pp. 228–238 (2004)
19. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, vol. 31, pp. 149–160. ACM Press, New York (2001)
20. Holmquist, L., Mattern, F., Schiele, B., Alahuhta, P., Beigl5, M., Gellersen, H.: Smart-its friends: A technique for users to easily establish connections between smart artefacts. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) UbiComp 2001. LNCS, vol. 2201, pp. 116–122. Springer, Heidelberg (2001)