

Addressing Context Dependency using Profile Context in Overlay Networks

Raheel Ali Baloch and Noël Crespi

*Institut TELECOM, TELECOM SudParis,
Evry, France*

raheel.baloch@it-sudparis.eu

noel.crespi@it-sudparis.eu

Abstract—By utilizing various sorts of contextual data acquired from the users' devices, advance context-aware services and applications can be developed. But the current context management systems are based on server-client approach which hinders their widespread adoption in an ad-hoc network. In addition, performance issues introduced because of context dependency in a ubiquitous environment still need to be addressed. This paper presents the idea of profile context to address the problem of context dependency, and also proposes an open framework for context acquisition, management and distribution in a ubiquitous environment. Assembling together the context information and context updates from various sources, support for context-aware decisions can be implemented efficiently in a mobile environment by solving the problem of context dependency using profile context.

Keywords— Context Management, overlay networks, DHT, P2P, ubiquitous computing

I. INTRODUCTION

The growing use of context information in mobile computing has picked up pace in recent times. Though, various types of traits can be utilized to characterize a situation, applications leveraging location are the most common in mainstream pervasive services. The use of context information in the form of user availability to start a conversation has been made particularly popular by instant messaging, both in desktop and mobile environments. Apart from instant messaging, context information utilization has also been of great assistance in decision making modules, helping to conserve already low resources in mobile devices.

Context is usually referred to the attributes in a continually dynamic computing environment in which various aspects like the user's location, social situation and interaction with other resources are alternating constantly. May be because of its more general approach, the definition provided by Dey and Abowd is most widely quoted in the research community: "Context is any information that can be used to characterise the situation of an entity." An entity can be defined as "a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves." [1]

The context acquisition process is a vital feature of any context-aware system. Most approaches try to acquire the device context instead of the "profile context"; each device owned by a user is contacted individually for the purpose of acquiring data relevant to the user situation. Context published

by a device fails to capture the overall picture of user situation in comparison to context obtained from multiple sources that are aware of their mutual existence, and can provide better user context working in collaboration among them. User context is usually dependent upon two or more devices' context.

Despite their potential widespread applications, context-based services are still in infancy because the context gathering architecture is based, more or less, on query-based silo framework. Such a framework is susceptible to numerous issues; most common of them are bottleneck performances and high probability of failure because of single point of contact. If the server-client approach is employed to realize a context-aware service, it can be easily imagined that servers will soon be overloaded with queries, affecting their performance. The dynamic nature of information to be handled in a ubiquitous computing environment is huge, and the numbers of queries increase with time as more users join the network.

In this paper, we aim to combine the idea of profile context and overlay network together for the purpose of proposing an open framework for context acquisition which also address the issue of context dependency. This approach enables the notion "Context is always available", based on the feature that the user's context is accessible even if he is offline or his devices are switched-off. This is possible because profile context keeps the latest context of the device in overlay network, and is available even after the device has been turned-off. Thus, an application can always access latest context avoiding delay resultant of context dependency.

The rest of the paper is organised as follows: Section II briefly explain context dependency, Section III presents the overlay network on which the context management system is based. Section IV describes the profile context management system in detail; Section V presents the simulation results of initial analysis. Section VI gives overview of related work; Section VII and VIII conclude the paper with proposals for future work.

II. CONTEXT DEPENDENCY

Due to highly dynamic nature of agents, too much context dependency can severely affect the performance of a ubiquitous system. Processing trees or graphs that illustrate the path of context information is not a valid option because of their high complexity. With the increasing number of

participants in an ad-hoc network, context dependency induced problems need to be handled efficiently. If the problem is not properly addressed, interdependence on low quality context will increase, leading to poor decision making by context-aware systems. There can be two kinds of context dependencies, as shown in the Fig 1. In the first dependency, a context of type A is being utilized by various applications to generate context of types B and C. In the second, there is interdependence of context between the participating applications. This cyclic context dependency can be serious threat to smooth execution of a context-aware system if it is unintentional.

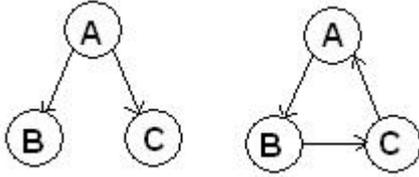


Fig. 1 Types of Context Dependency

Most simple scenario to depict context dependency is the availability of the user on his mobile phone when he is in a meeting. In such a situation, user can only be contacted on his mobile phone through text message, avoiding voice calls altogether. The decision to drop voice calls is based on two contexts; user location and calendar entry of the meeting. The mobile application is dependent upon these two contexts to make correct decision. If any one of the context is not acquired entirely or even acquired late, the usefulness of such mobile application is not valid anymore as the decision is dependent upon these two context.

III. OVERLAY NETWORK

One promising approach for realizing efficient context retrieval framework can be overlay network. An overlay network can efficiently manage network resources, and distribute both the processing load and the network load evenly in comparison to a server-client system. There have been many research efforts regarding information retrieval in an overlay network. Especially, information management methods using Distributed Hash Tables (DHTs) are proposed in numerous studies [2]-[4]. The DHTs can distribute the processing load by an efficient query routing. Techniques that construct a P2P network based on the topics of contents or communication times between peers have also been proposed [5], [6].

The context acquisition system proposed in this paper is to be maintained on a P2P service platform called SCOPE that is being developed by the RS2M team at Institute Telecom SudParis. Its aim is to provide a distributed infrastructure with unified API that enables development of miscellaneous P2P services on ad-hoc networks. The motivation behind SCOPE is to provide an efficient overlay for P2P services over ad-hoc networks [7]. The services framework is based on Bamboo open-sources [8] which provides PASTRY [9] DHT services. Bamboo with its public name OpenDHT [10] has been deployed on Planetlab [11]. Bamboo has been developed to

handle high churn [12], which means the continuous process of node arrival and departure. It also uses proximity neighbour selection [13], [14].

The context information of each device is acquired by the API on one of the super nodes in the overlay network. The super node helps to store the information in the peer nodes along with the context provided by the other devices own by the same user. The profile context is stored in the XML format. Applications requiring user context can enquire one of the super nodes for the address of the peer node storing the XML file in the overlay network. The applications can also have them registered with a super node to receive the profile context whenever it is updated, or when a device context is added to the profile context. There can be other options apart from these two, e.g. the context is regularly provided to the application after an already agreed upon time interval. The storage and distribution of context data is handled by the APIs running on the super nodes. The list of subscribing applications, that want to have context information forwarded to them, is maintained by the API on the super node and notifications are sent to the relevant applications whenever the stored context is updated in the overlay network.

IV. PROFILE CONTEXT MANAGEMENT

Profile context is the main idea of our paper, and this section will highlight its management aspects, specifically how to manage profile context information in a P2P overlay network. But first, we define profile context itself.

A. Profile Context

Almost all of the research literature has viewed gathering context as a server-client problem with context only being obtained from a single device, which can be appropriately called “device context”. The idea of “profile context” is being promoted in this paper which can be rightly considered as a collection of context from various devices and sources whose context information is related to a common single user. Such a context engulfs all the context information efficiently from numerous resources, which can be then successfully utilized to determine current situation of a user precisely.

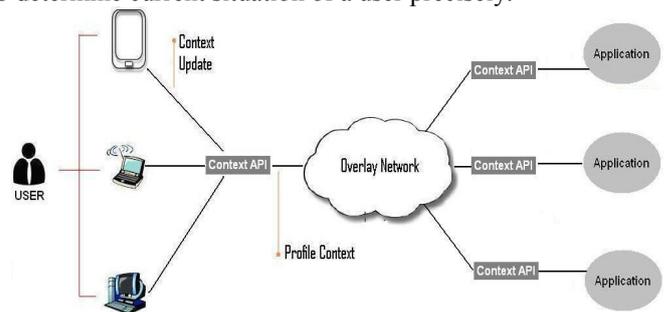


Fig. 2 Profile Context Management Framework based on Overlay Network

Profile context can be formally defined as a finite set, P_C , of all the devices' context belonging to the owner of the profile. If each of the device context is represented by D_{C_i} , and there are N devices providing the user context, then

$$P_C = \{D_{C_1}, D_{C_2}, D_{C_3} \dots D_{C_N}\} \quad (1)$$

The context information present in whole of the overlay network can represent profile context of one or more than one users. The accumulated context, A_C , can be represented as a superset of all the profiles' context in the overlay network. If there are M users in the overlay network, then

$$A_C = \{P_{C1}, P_{C2}, P_{C3} \dots P_{CM}\} \quad (2)$$

B. Context Update

For the purpose of better understanding the context update process, a sequence diagram in Fig. 3 is presented. There are two devices belonging to a user that act as the source of context information. Each device publishes its context whenever there is a change in the context. A super node in the overlay network running the context API is provided with the device context. The API then forward the context update to a peer node using appropriate hashing scheme. The peer node accepts and stores the context file. When another device belonging to the same user publishes its context to the context API, it is forwarded to the same peer node which stored the context of the user's last device. The peer node aggregates the context from the two devices into a single XML file. It should be noted that federated-identity provides the mapping between different profiles of the same person obtained from various devices.

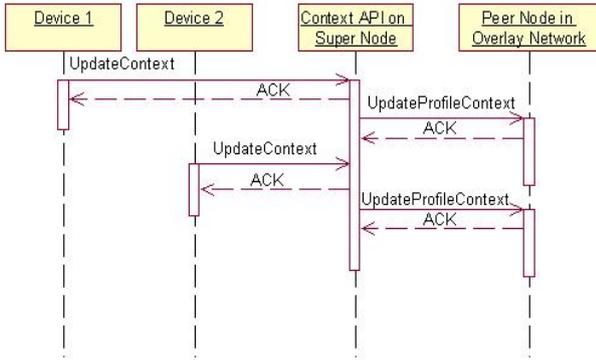


Fig. 3 Sequence Diagram of Context Update.

C. Context Retrieval

Context retrieval in an overlay network is quite simple in a sense that applications only need to send their query to one of the super nodes that take care of context retrieval for the respective applications. In the sequence diagram, Fig. 4, the application requests the context of the user from the super node running context API. By using appropriate hash function, the super node finds the address of the peer node storing the profile context. The peer node is informed of the context request from the particular application. The profile context of that particular user is then forwarded to the concern application.

When there are applications that have subscribed to receive context information when an update occurs, the super node takes the responsibility of multicast-like broadcast as it has more resources than a peer node, e.g. in terms of battery life, network bandwidth, etc.

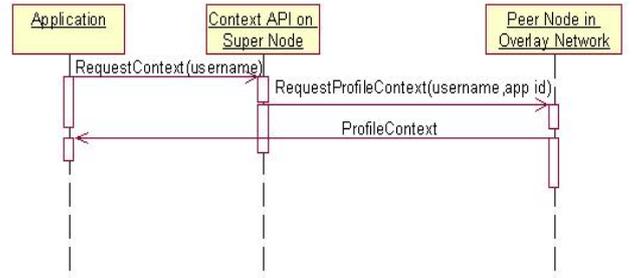


Fig. 4 Sequence Diagram of Context Retrieval.

V. EVALUATION

At the time of writing this paper, the simulation is conducted to incorporate initial results in the paper as the proposed framework is still being integrated with the SCOPE architecture. In the simulation, the churn rate is assumed to be zero. The churn rate is the rate at which the nodes join or leave the overlay network. If the churn rate is high, then the overlay network's performance drops down as the context information on the nodes, that are leaving the overlay network, needs to be copied or saved on to other peer nodes. The packet drop rate is also set to zero, as the simulation was conducted to isolate the difference between device context approach and profile context approach. The simulation was developed in .NET programming environment. For the purpose of evaluation, we have concentrated on the number of queries in the system, each time when a service or application requires updated context. We have observed that our proposed framework shows better performance when the services or applications sent queries to all the devices owned by same user. If a query is aimed at only one of the devices, the performance gains in terms of network traffic remains negligible.

Replies Generated in Response to Queries

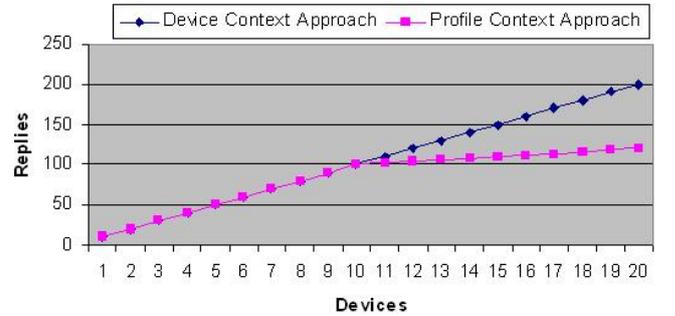


Fig. 5 Comparison between Queries generated using Device Context and Profile Context.

Performance in terms of traffic generated as the result of queries is plotted in Fig 5. The number of users in the simulation was set at 10, and each user can simultaneously own up to 2 devices. Initially, when the overlay network is being populated by the introduction of users, the performance of overlay network using profile context remains almost the same as that of device context approach. The number of responses for profile context approach starts to divert from device context approach when the user starts to have more than one device in their use. This is due to the introduction of

profile context which replaces device context. Performance in terms of traffic generated when a single user introduces more than one device in the network is plotted in Fig 6. The number of devices in the simulation was gradually incremented up to 10 devices. The responses for profile context approach are significantly less than device context approach as the user has more than one device in his use.

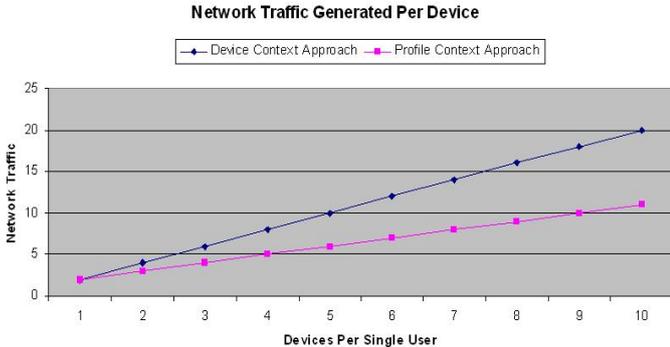


Fig. 6 Comparison between Network Traffic generated for each device using Device Context and Profile Context.

VI. RELATED WORK

Implementation of multi-tier architectures is a prevailing trend in pervasive computing. Such architectures provide abstraction layers which help to delegate details pertaining to context management. The various approaches presented in recent research literature are based on the concept of separation of context gathering and distribution processes from the focal application logic. A well known server-based platform, the Context Toolkit (CT) [15], try to hide details related to sensors, and for the representation of contextual data, XML structures are employed. The context providers and context interpreters serve the purpose of context abstraction layer. The context can be utilized with services and other components present in the system through the context harvester. For the reason of privacy protection, only basic access control procedure is made available in the Context Toolkit. Privacy guidelines and rules are stated but lack any further details. The vision propagated by our approach is to promote open infrastructure for context. The same is proposed in [16], trying to utilize delegation to external applications and components for an integrated infrastructure. Analogous to Context Toolkit, control over the usage of context with external applications is also restricted in [17]. The aggregation and sharing of context over an open infrastructure is one of the main ideas of our scheme to enable truly context-aware services. Context aggregators with the help of context widgets and interpreters collect the entire context data from sensors in an architecture described in [18], [19]. In [20], a similar infrastructure has been introduced in which context information is provided by sensors who forward it to context applications through the medium of context providers and context services. Applications' requests are forwarded to relevant context drivers in a comparable architecture proposed in [21]. The context drivers can deal with only a particular type of context data and act as an abstraction layer for the context sources. A Service Oriented Architecture (SoA) has

been presented in [22], called Service-Oriented Context-Aware Middleware. SoA is a preferred approach to develop applications that require context management services to form ad-hoc network. Further work done related to context aggregation can be found in [23]-[26]. The main drawback in these studies is that they lack the consideration to various performance issues such as number of queries generated, number of context updates, etc.

VII. DISCUSSION AND FUTURE WORK

The proposed context management framework is merely intended as a demonstration of the concept of profile context, and its acquisition and distribution through overlay network. As can be imagined, the concept currently lacks solutions to various obstacles that are faced in practical implementation. One of the most prominent features of any context management system is the provision of adequate user privacy and security of user's data, and we concede that our proposal lack such assurances at the moment. The question of how to make sure that only the relevant information from the profile context is disclosed to the service or application remains to be addressed. If an application is only authorized to access a single device context, there should be some mechanism to ensure that only that information is disclosed to the application, not the whole context which comprises information from more than one device.

The issue of scalability is another hurdle that needs to be dealt with. The information overflow due to high churn rate can clog up the performance of the overlay network. In our opinion, the proposed framework is more than capable to handle context information with in a considerable community of users. Stress testing of SCOPE still needs to be done to determine its failure point in terms of the number of users in the overlay network. The increasing wireless network bandwidth availability and the idea of cloud computing can combine to provide a practical solution to the problem of scalability in the near future. Convergence of flexibility of Web 2.0 and the reliability of network infrastructure can result in an interesting future for cloud computing in terms of context management solutions.

The distributed nature of overlay networks provide improved robustness in terms of failures by making the availability of context information possible over multiple peers. It provides the advantage of low latency and high resilience in regards to fault tolerance.

Our goal is to provide an open platform for context management in a ubiquitous environment, and overlay networks have the advantage that permits both application users and developers to design and implement their own context-aware services. The end nodes can always communicate with one another through overlay network, so robustness in overlay networks is another attractive feature for our system. The high connectivity of end nodes allows effective sharing of context information in a distributed environment lacking centralized authority.

The proposed framework, as mentioned earlier, is a workable solution that already shows with the help of results,

though quite initial themselves, that it can sufficiently provide gains in terms of reducing the additional network load commonly associated with other context management systems. Further detailed study is essentially required to assess the viability of the system in common real life situations.

For the near future, we will try to propose a workable solution to the privacy issue. One approach can be directed in a way that an unauthorized application can not access the context information, through the introduction of authentication dialogue between the owner of the context and one of the super nodes. This is a very crude solution, as it results in extra information flow that we are trying to reduce in the first place. Introducing some privacy features in XML can be a feasible solution [27], [28].

Further testing in real environment is on our agenda, and we hope our system will be able to perform as better as its simulation results have indicated. The evaluation will be done with rich application on low wireless bandwidth, as is the case in wireless networks. Formal modelling context dependency will also be aimed in near future.

VIII. CONCLUSION

This paper presents a context management framework that is based on the idea of leveraging overlay network to efficiently manage profile context making it available at all times in the system, and in turn address the issue of context dependency. Our system utilises overlay network paradigm in an innovative and unique way to optimize the distribution of users' context information. Until now context management systems have presented various solutions that are mostly based on query-based centralised architecture, but in our approach the user context is stored and distributed using a P2P overlay network. Availability of context due to overlay network reduces the risk of performance bottlenecks introduced due to context dependency. The initial simulation results have shown that our proposed context management system performs better in terms of network traffic generated as a result of queries from various applications requiring context information. Our proposed context management framework is particularly aimed at devices and applications in a ubiquitous environment where lack of resources restricts optimal performance.

REFERENCES

- [1] Dey, A. & Abowd, G. (1999) Towards a Better Understanding of Context and Context-Awareness. In: *CHI '00: Workshop on the What, Who, Where, When, and How of Context-Awareness*, Georgia Institute of Technology, Atlanta, GA, USA, 12 pages.
- [2] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network", *Proc. ACM SIGCOMM 2001 Conference*, pp. 161-172, 2001.
- [3] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", *Proc. ACM SIGCOMM 2001 Conference*, pp. 149-160, 2001.
- [4] B.Y. Zhao, J.D. Kubiatowicz, and A.D. Joseph, "Tapestry: A Fault-tolerant Wide-area Application Infrastructure", *ACM SIGCOMM Computer Communication Review*, Vol. 32, No. 1, p. 81, 2002.
- [5] M. Bawa, G. Manku, and P. Raghavan, "SETS: Search Enhanced by Topic Segmentation", *Proc. 26th Annual ACM Conference on Research and Development in Information Retrieval*, pp. 306-313, 2003.
- [6] M. Kwon and S. Farny, "Topology-Aware Overlay Networks for Group Communication", *Proc. 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 127-136, 2002.
- [7] Mani M, NGYUEN A.M., Crespi N., "What's Up: P2P Spontaneous Social Networking", *Proceeding of PERCOM 2009, IEEE International Conference on Pervasive Computing and Communications*, March 2009, Galveston Tx, USA.
- [8] Bamboo-DHT <http://bamboo-dht.org/>
- [9] M. Mani, W. Seah, N. Crespi, "Efficient P2P service control overlay construction to support IP Telephony services over adhoc networks," *In Proceeding of IEEE MASS*, October 2007.
- [10] OpenDHT <http://www.opendht.org/>
- [11] Planetlab <http://planet-lab.org>
- [12] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. Handling Churn in a DHT. In *Proceeding of USENIX*, June 2004.
- [13] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu. "OpenDHT: A public DHT service and its uses," *In Proceeding of SIGCOMM*, Aug. 2005.
- [14] S. Rhea. "OpenDHT: A public DHT service," PhD thesis, U.C. Berkeley, Aug. 2005.
- [15] A. K. Dey and G. D. Abowd, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *HCI Journal*, vol. 16, no. 2-4, pp. 97-166, 2001.
- [16] M. Wojciechowski and J. Xiong, "Towards an open context infrastructure," in *Proc. of the Workshop on Context Awareness for Proactive Systems (CAPS'06)*, pp. 125-136, 2006.
- [17] R. Sandhu and J. Park, "Usage control: A vision for next generation access control," in *Proc. of the Workshop on Math. Methods, Models, and Architectures for Computer Network Security*, pp. 17-31, 2003.
- [18] Dey A. K., Providing Architectural Support for Building Context-Aware Applications, PhD thesis, College of Computing, Georgia Institute of Technology, December 2000.
- [19] Dey A. K., "Understanding and using context.," *Personal and Ubiquitous Computing*, 5(1):4-7, 2001.
- [20] Huebscher, M. C., McCann, J. A. An adaptive middleware framework for context-aware applications, *Personal and Ubiquitous Computing, Volume 10, Issue 1*, Feb 2006, Pages 12 - 20.
- [21] Lei H., Sow D. M., Davis J. S. I., Banavar G., Ebling M. R., "The design and applications of a context service", *Mobile Computing and Communications Review*, 6(4): 44-55, 2002. ACM SIGMOBILE.
- [22] Gu, T., Pung, H. K., and Zhang, D. Q. 2005. A service oriented middleware for building context-aware services. *Journal of Network and Computer Applications*. 28, 1 (Jan. 2005), 1-18.
- [23] J. Heer, A. Newberger, C. Beckmann, and J. I. Hong, "liquid: Contextaware distributed queries," in *Proc. of the Fifth International Conference on Ubiquitous Computing: Ubicomp 2003*. Seattle, WA: Springer-Verlag, 2003, pp. 140-148.
- [24] G. Judd and P. Steenkiste, "Providing contextual information to pervasive computing applications," in *Proc. of the IEEE International Conference on Pervasive Computing (PERCOM)*, Dallas, March 23-25 2003.
- [25] G. Chen and D. Kotz, "Context aggregation and dissemination in ubiquitous computing systems," in *Proc. of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.
- [26] S. S. Yau, D. Chandrasekar, and D. Huang, "An adaptive, lightweight and energy-efficient context discovery protocol for ubiquitous computing environments," in *Proc. of 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)*, 2004.
- [27] *Enterprise Privacy Authorization Language (EPAL)*, Version 1.2, 2003; the version submitted to the W3C. Available at <http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/>.
- [28] *eXtensible Access Control Markup Language (XACML)*, Version 2.0; OASIS Standard, February 1, 2005. Available at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.