Design and Implementation of a Reference Model for Context Management in Mobile Ad-Hoc Networks

Dominique Dudkowski, Harald Weinschrott
University of Stuttgart
Institute of Parallel and Distributed Systems
Universitätsstraße 38
D-70569 Stuttgart, Germany
{dudkowski|weinschrott}@ipvs.uni-stuttgart.de

Pedro José Marrón
University of Bonn
Institute of Computer Science IV
Römerstrasse 164
D-53117 Bonn, Germany
pjmarron@cs.uni-bonn.de

Abstract

The fundamental paradigm of context-aware computing has lead to a diversity of conceivable context-based applications. In order to support these applications, efficient and scalable context management platforms have become utterly important. In this paper we propose a layered reference model that encapsulates suitable abstractions to tackle the complexity of context management in mobile ad-hoc networks. For each layer, we outline strategies and mechanisms we have developed to manage context information efficiently so that it can readily be used by context-based services and applications in these networks. In addition, we discuss refinements of the current model that we currently investigate, which are related to data semantics such as context quality and expressive location models, performance enhancement by adaptation, and aspects regarding the integration of ad-hoc context management with infrastructure-based systems leading to a hybrid system structure.

1 Introduction

The paradigm of context-aware computing refers to the ability of applications and services to carry out their tasks depending on context information. Such information may comprise any aspect considered relevant to the respective entity, such as the status of an application's graphical user interface or a tourist's whereabouts in the city over time. Due to the variety of conceivable context-based applications and their potential use in large-scale scenarios, support for these applications by means of scalable services that carry out frequent tasks based on context information is needed. Some of these tasks comprise the processing of spatial queries, observation of real-world events, and computation of routes for user navigation.

In the past years a number of platforms have been under development that target at the efficient and scalable management of context information primarily in fixed computer networks [1, 8, 11]. The pervasiveness of mobile devices and their capability to form mobile ad-hoc networks (MANETs) suggests autonomous context management also in these networks. Until now, an integrated approach for scalable context management addressing algorithmic aspects from the routing layer up to the service layer in mobile ad-hoc networks has not been sufficiently addressed.

The contribution of this paper is, therefore, a reference model for the implementation of context management in MANETs. The proposed model comprises three major layers, each of which addresses a specific part of the overall complexity of context management: core storage, update processing, and context services. The functionality of each layer is published via interfaces that can be exploited by context-based services and applications. For each layer, we highlight parts of our own implementation in order to show how the overall reference model can be implemented to provide efficient context-management in MANETs. We further state current research directions that we address within the Nexus project [8], whose goal is to implement and provide context models on a global scale. Specifically, these issues comprise data semantics, including context quality and expressive location models, performance enhancements through adaptation, and ultimately, concepts for integrating the ad-hoc context management system into a hybrid system structure.

The rest of this paper is structured as follows. In the next section we discuss related work, followed by a summary of key requirements related to context management in MANETs in Sec. 3. Our reference model is introduced in Sec. 4 together with selected implementation aspects of each of the model's layers. Our current research on model and implementation refinements is described in Sec. 5, before we conclude the paper in Sec. 6.

Published in Proc. of 22nd International Conference on Advanced Information Networking and Applications - Workshops (WAINA'08), pp. 832 - 837. Okinawa, Japan. March 2008

http://www.computer.org/portal/web/csdl/doi/10.1109/WAINA.2008.72

2 Related Work

Let us first consult two recent surveys on context-aware systems [1, 11] to discuss the spectrum of existing approaches with respect to context management in MANETs. Specifically, the authors of [1] have devised a layered conceptual architecture that captures the design space from sensors and raw data retrieval over preprocessing and storage/management up to the layer of context-based applications. The large part of considered systems in line with this abstract model, e.g., MobiPADS [2], Mobile Gaia [4], ACAN [10] or CORTEX [13], capture a wide range of context-related objectives. These systems, however, are context-dependent rather than context management systems for a broad spectrum of different context-based services.

Systems supporting more generic context management for diverse context-based services and applications are the framework of mobile context management in [3] and Nexus [8]. While [3] provides context dissemination for more specialized applications, it does not provide an integrated reference model to tackle the complexity of context management in MANETs more generally. Nexus, on the other side, is the only system that provides a general approach to context management for a broad range of services. However, it is infrastructure-based and does not handle the special characteristics of wireless multihop networks.

3 Requirements

An adequate reference model for context management in mobile ad-hoc networks requires the consideration of a number of key requirements. We have identified the following three groups of requirements, related to system performance, context information, and functional properties.

Requirements related to **system performance** relate to more general aspects of the overall system and in particular to the inherent characteristics of mobile ad-hoc networks:

(1) Communication efficiency of the overall system as key to the feasibility of a complex management system and the maximization of overall quality of service
(2) Resilience to dynamic and potentially weak topological structures, as well as independence from strict assumptions that are based on topological connectivity and low node population (network density)
(3) Resilience against significant node mobility on various mobility levels, such as pedestrian and vehicular traffic in typical urban context-aware scenarios
(4) Enforcement of *spatial coherence* [7], that is, retaining context information at dedicated locations in the network to enable efficient processing of functional features in the first place, in particular, of location-based services (e.g. spatial queries, events)

Requirements related to **context information** address the specific characteristics of context information itself:

(5) Ability to manage significant amounts of context information as required by the diverse context services
(6) Ability to efficiently process dynamic context information in order to provide sufficient data quality over time for optimum user experience
(7) Support for rich data semantics by considering context quality such as the accuracy of location information, as well as expressive hybrid location models combining geometric and symbolic location information

Functional requirements are derived from the view of context-based applications and services:

(8) Functionality of service, that is, provide the most relevant services within the reference model to in turn support context-based applications to a large extent
(9) Generic interfaces for interaction between context-based services and applications on one side and more fundamental context management facilities on the other side to allow cross-layer optimizations
(10) Adequate abstractions within the interfaces for integrability purposes, in particular, with hybrid system structures to form hybrid context management systems

4 Reference Model

Our reference model is targeted at MANETs that are formed by spontaneous communication between large numbers of autonomous wireless devices (nodes). Nodes in the network may issue data updates and queries (e.g. spatial range queries), and take the role of a data server to store a portion of the overall context information. We assume that data objects (e.g. modelled vehicles) each possess a number of attributes (e.g. location) and meta data (location accuracy).

Fig. 1 depicts the reference model that we have derived from the requirements. In the following, we will describe in more detail each layer together with selected implementation details to show the suitability of the overall management approach. Integrative aspects with respect to hybrid system structures, as indicated on the left-hand of the figure, is under current research and will be discussed in Sec. 5.

4.1 Core Storage

The *core storage* layer performs fundamental tasks related to data storage in MANETs. At the base is the abstraction of data servers each storing a subset of the context information. Storage follows the location-centric storage (LCS) paradigm [7], where a data subset is stored at servers located

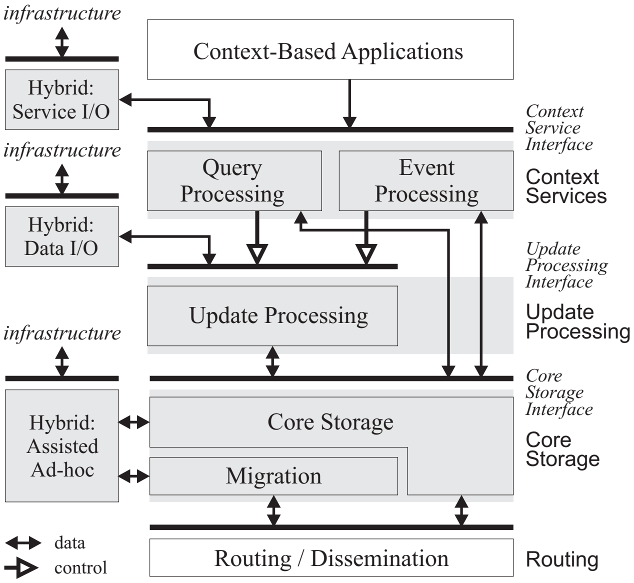

Figure 1. Reference Model

near a geometric location in the network, which enables efficient location-based services in the first place. Tasks performed at the core storage layer include forwarding updates and queries from clients to servers for processing. This layer is also responsible for maintaining spatial coherence by means of data migration, that is, keeping data in close proximity to its geometric location over time.

The functionality of the core storage layer is published to the upper adjacent layer via the core storage interface. By this interface, updates and queries may be passed down to the core storage layer, which will in turn propagate each update and query to the correct server. At the target server, control is handed back to the upper layer, in order to correctly execute updates and evaluate queries based on the specific semantics of the context information (cf. Sec. 4.2 and 4.3). The core storage layer also connects to the routing layer, as it naturally relies heavily on routing and dissemination primitives for model management (e.g. data forwarding) as well as system management (e.g. data migration).

Core storage addresses system performance requirements (1) to (4) on the basic level. That is, it alleviates the impact of node mobility and dynamic network topologies on storage performance while at the same time ensuring efficient data forwarding by maintaining spatial coherence. Since the core storage layer has no knowledge about the context data's semantics, it may perform only generic optimizations based on the interplay between routing and dissemination strategies. However, it provides location-centric storage, which is already tailored to the functional requirements of context-based services and applications.

Implementation: In [6] we present algorithms for efficient forwarding of updates and queries to data servers. For that purpose, we have introduced the concept of bidirectional perimeter routing (BPR [6]). In contrast to unidirectional perimeter routing as used, e.g., in greedy perimeter stateless routing (GPSR [9]), a perimeter is traversed in both directions. At the same time, we introduce an indirection scheme that decouples the storage of context information on a dedicated server from the actual reference geometric location. This is in contrast to data-centric storage over GPSR [12], where data is stored directly at a home node determined based on full perimeter traversals. Complementary, we provide migration strategies in [7] and show that spatial coherence can be effectively maintained under the typical characteristics of MANETs.

Due to the core storage layer's key role in efficiency and appropriateness of context management, we have developed an analytical model that allows us to assess the performance spectrum within which our core storage approach performs well in comparison to alternative conceivable approaches. For that, we consider a square area $A = 1200\text{ m} \times 1200\text{ m}$ with 600 nodes moving randomly at a speed of 1.5 m/s. Each node located in the square $S = 400\text{ m} \times 400\text{ m}$ centered in A updates its location at a rate of 0.3 updates per second according to each considered approach. We assume that every node in A queries for all updated location objects every 60 seconds, totalling 10 queries per second.

Fig. 2 shows the communication costs required for a number of approaches: global and cell-based replication where all data is stored on nodes in A and S, respectively; global and cell-based partitioning, where each data object is stored on the originating node only, and retained within S by migration in the case of cell-based replication; DCS over GPSR according to [12], and our own approach, LCS over BPR. In Fig. 2.a, with increasing update frequency, replication is eventually outperformed due to the fact that each update has to be replicated at flooding costs either in the whole region, or inside the cell S. In Fig. 2.b, a variation in the query frequency shows the same effect with respect to data partitioning whose performance degrades rapidly, because each query must be processed by referring to a large number of nodes. The core storage's optimum performance range highlighted in both figures corresponds well to the requirements of typical context services that we consider.

For additional performance measurements based on simulative methods of the core storage approach we would like to refer to [6] and [7].

4.2 Update Processing

The layer of *update processing* contains the necessary functions to propagate updates from clients to the relevant data servers. Updates may, for example, contain the information captured by sensors from the environment by a client, or a

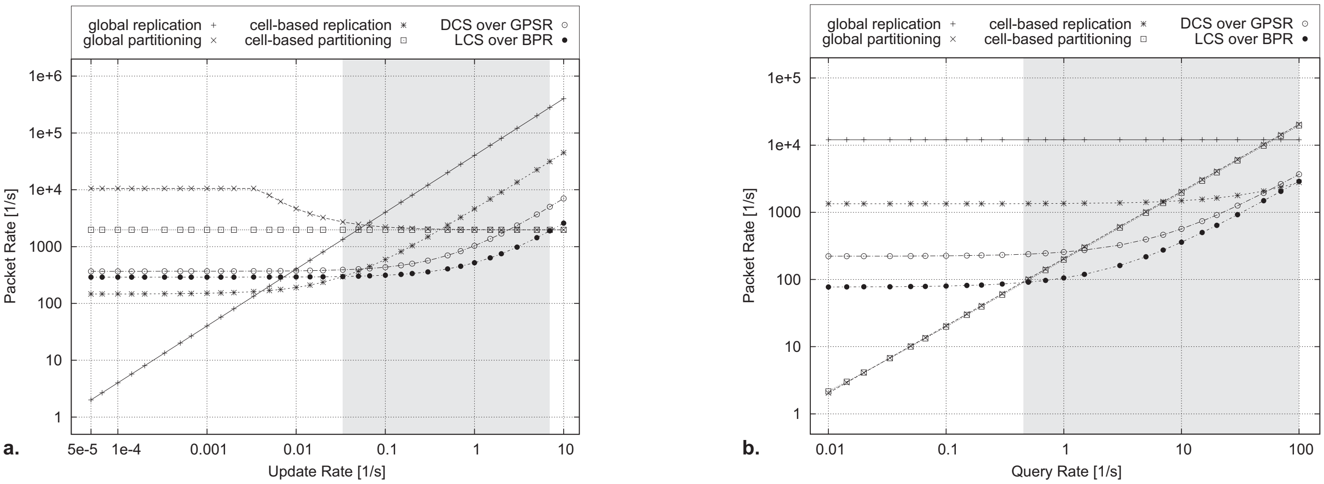

Figure 2. Performance spectrum. a. update frequency. — b. query frequency.

client's own position information acquired by a locally installed GPS device. At this layer, the semantics of the context information are known, thus updates can be executed on the local database. For example, an existing copy of a data object can be overwritten by a new update if the incoming information is more current.

Knowing the data semantics allows to define additional optimizations that were not possible on the storage layer. For instance, the update frequency can be reduced to a level where the location accuracy requirements of applications are still satisfied. Such quality requirements can be specified by context-based services and applications via the update processing interface, which are then taken into account by the update processing component. In addition, this interface allows services to subscribe to particular context information (context subscription) for their specific needs that is not available through existing updates already. For example, a meeting event may require the location information of two distant objects, which would normally not be stored on a single data server. These kinds of interactions are special parameterizations in contrast to data exchange, which we have indicated with special arrows in Fig. 1.

Implementation: Currently, we have implemented the processing of updates for location information [5]. This information is particularly useful as a use case for other types of dynamic context data, which frequently involve similar dynamics and quality models, e.g., in the form of two-dimensional probability distributions.

In addition to the update processing mechanism in [5], we illustrate the concept of context subscriptions in Fig. 3. Data server DS_1 takes the role of a subscriber and issues two subscriptions to DS_2 to request the location information of particular objects: one subscription for data object o_2's location whenever o_2 is located inside of region R_1, and one for objects of type *pedestrian* that are located inside of region R_2. When DS_2 stores object information matching any of these filters, DS_2 will forward the object's position to DS_1, which will in turn be able to process, e.g., a meeting event involving this object and some other object already stored at DS_1. Also in this forwarding process, we fully consider optimizations based on dead reckoning to achieve a more efficient processing of these additional subscribed updates. This approach is applicable to many other kinds of context data supported in a specific context model.

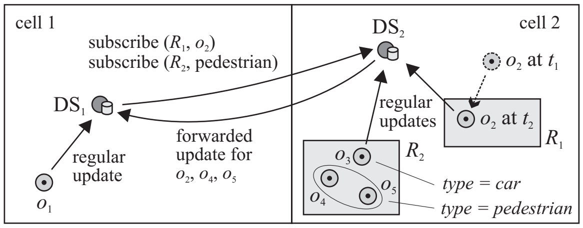

Figure 3. Context Subscriptions

4.3 Context Services

Context services refers to the third level of our context management reference model. In Fig. 1, we have exemplarily included services for query processing and event observation, which are two of the most frequently demanded services in context-based applications.

By accessing the context management system through the update processing and core storage interface, these services are decoupled from the specific characteristics of the mobile ad-hoc network. Through the former interface, services are able to parameterize update processing for optimizations based on context quality and additional required updates, as discussed in Sec. 4.2.

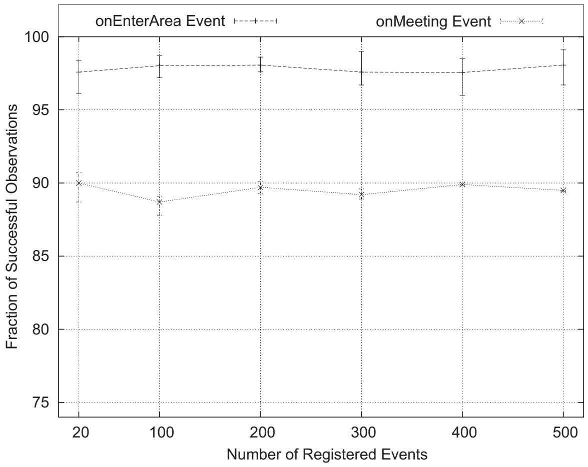

Figure 4. Event Observation

Implementation: We have devised algorithms for processing spatial queries, namely, probabilistic range and nearest neighbor queries [5]. In the proposed algorithms, we are able to process queries based on extended location semantics, which model inaccurate position information as obtained from typical GPS devices.

In this paper, we present selected results for the observation of spatial events in MANETs. Fig. 4 shows the success rate in observing the onEnterArea event (object enters geometric region) and the onMeeting event (two objects meet at a given distance) as a function of the number of registered events. In light of the quality constraints that are inherent to any method of position acquisition, observation accuracy is sufficiently high to be used in typical civilian scenarios, even for a large number of registered events.

5 Current Research

Based on experience with our reference model in Fig. 1 and its implementation we have identified a set of issues we are currently investigating in order to enhance the performance and value of our context management system.

5.1 Data Semantics

Up to this point only geometric location information and its inaccuracies were modelled. Currently two main extensions, with respect to data semantics, are investigated.

First, the context management system needs to support a variety of different location models in order to reflect different location systems and to increase the expressiveness of the modelled data. This implies support for symbolic as well as complex hybrid location models. Particularly, this requires new routing mechanisms as well as management of location model data in the ad-hoc network. Our reference model provides the appropriate structuring that allows to break up the complexity of this task.

Second, the system needs to provide context data as well as meta data to describe the quality (e.g. accuracy) of the data provided by the system. This allows to estimate the reliability of the system's returned results. To realize this concept, we develop proper quality metrics and mechanisms to adapt update processing in order to provide the data with defined quality. The extension of the data semantics supported by the system is highly challenging, with impact on all layers of the reference model.

5.2 Adaptation

For a more complex set of concurrently running context services, efficiency of the overall system is the most critical issue. Therefore, despite the already realized adaptation mechanisms (data migration, context subscription), adaptivity needs to be exploited on a broader scale.

With the support of rich data semantics as described in Sec. 5.1, the system can adapt its update processing mechanism to run at reduced update frequencies also for other types of context data besides location information (Sec. 4.2). This will in turn lead to an overall increase of system efficiency. In addition, the concept of context subscriptions can be similarly generalized to adapt to more individual functional requirements of certain context services over time, while the core functionality of the data management reference model is retained.

5.3 Hybrid System

The current state of our context management system supports pure ad-hoc system structures, which we plan to bring together with infrastructure-based networks to form a hybrid context management system.

Assisted ad-hoc refers to a system where the context data is retained only on the nodes in the ad-hoc network. However, infrastructure can be used to improve performance and quality of service. The motivation to support this concept is due to restrictions of MANETs which can lead to situations that simply cannot be resolved. For example consider a node trying to access data stored in a different network partition than its own. Such disconnections between partitions can be overcome by relying on the infrastructure for a specific action, which would lead to inaccessible data in a pure ad-hoc network otherwise.

In the case of *full hybrid model management* both, an infrastructure-less and infrastructure-based system are integrated in order to achieve an efficient and robust overall system. It is expected to combine the advantages of the isolated approaches allowing for the infrastructure-based por-

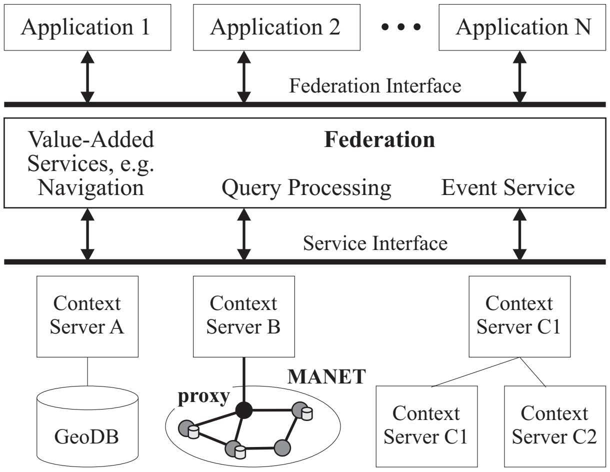

Figure 5. Nexus Hybrid System Structure

tion of the system to access context data that is stored in the infrastructure-less part of the system and vice versa.

Suitable abstractions are required based on which an integration can be achieved. Fig. 5 shows the hybrid system structure of the Nexus [8] platform. Applications access context data through the federation which abstracts from the diversity and distribution of different data providers. A MANET is abstracted as just another type of context server and accessed via a proxy adaptor which in turn connects to the data storage servers in the infrastructure-less part of the network. Thereby, the MANET can be accessed on the different interfaces indicated on the left-side of Fig. 1, transferring raw or semantically enriched data between both types of network structures.

6 Conclusions

In this paper we proposed a reference model for context management that supports the development of context-based services and applications in mobile ad-hoc networks. By introducing three layers, each of which addresses a particular portion of the overall complexity, the model allows optimizations of individual algorithm components and at the same time a view of the interrelations across the overall context management functionality. Interfaces connecting adjacent layers allow services and applications exchange data with data storage and parameterize update processing to adapt the system to individual application needs.

Current work within the Nexus project focuses on additional research topics, including the consideration of context quality and methods of adaptation to further increase overall system performance. The ultimate goal is the integration of the ad-hoc context management system into a hybrid system structure such as the Nexus architecture to provide a comprehensive context model on a global scale.

References

[1] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *J. Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[2] A. T. Chan and S.-N. Chuang. Mobipads: A reflective middleware for context-aware mobile computing. *IEEE Trans. on Software Engineering*, 29(12):1072–1085, Dec. 2003.

[3] T.-S. Chen, G.-J. Yu, and H.-J. Chen. A framework of mobile context management for supporting context-aware environments in mobile ad hoc networks. *Proc. Int'l Wireless Communications and Mobile Computing Conf. 2007 (IWCMC'07)*, pages 647–652, Honolulu, HI, USA, Aug. 2007.

[4] S. Chetan, J. Al-Muhtadi, R. Campbell, and M. D. Mickunas. Mobile Gaia: A middleware for ad-hoc pervasive computing. *Proc. 2nd IEEE Consumer Communications and Networking Conf. (CCNC'05)*, pages 223–228, Las Vegas, NV, USA, Jan. 2005.

[5] D. Dudkowski, P. J. Marrón, and K. Rothermel. Efficient algorithms for probabilistic spatial queries in mobile ad hoc networks. *Proc. 1st Int'l Conf. on Communication System Software and Middleware (COMSWARE'06)*, New Delhi, India, Jan. 2006.

[6] D. Dudkowski, P. J. Marrón, and K. Rothermel. An efficient resilience scheme for data centric storage in mobile ad hoc networks. *Proc. 7th Int'l Conf. on Mobile Data Management (MDM'06)*, Nara, Japan, May 2006.

[7] D. Dudkowski, P. J. Marrón, and K. Rothermel. Migration policies for location-centric storage in mobile ad-hoc networks. *Proc. 3rd Int'l Conf. on Mobile Ad-hoc and Sensor Networks (MSN'07*, Beijing, China, Dec. 2007.

[8] M. Grossmann, M. Bauer, N. Hönle, U.-P. Käppeler, D. Nicklas, and T. Schwarz. Efficiently managing context information for large-scale scenarios. *Proc. 3rd IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom'05)*, pages 331–340, Kauai, HI, USA, Mar. 2005.

[9] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. *Proc. 6th Ann. Int'l Conf. on Mobile Computing and Networking (MobiCom'00)*, pages 243–254, Boston, MA, USA, Aug. 2000.

[10] M. Khedr and A. Karmouch. ACAN - ad hoc context aware network. *Proc. Canadian Conf. on Electrical & Computer Engineering (CCECE'02)*, pages 1342–1346, Winnipeg, Canada, May 2002.

[11] K. E. Kjaer. A survey of context aware middleware. *Proc. IASTED Software Engineering Conf. 2007*, Aug. 2007.

[12] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-centric storage in sensornets with ght, a geographic hash table. *Mobile Networks and Applications*, 8(4):427–442, Aug. 2003.

[13] C.-F. Sørensen, M. Wu, T. Sivaharan, G. S. Blair, P. Okanda, A. Friday, and H. Duran-Limon. A context-aware middleware for applications in mobile ad hoc environments. *Proc. 2nd Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC'04)*, pages 107–110, Toronto, Canada, Oct. 2004.