MOBILITY MANAGEMENT IN SENSOR NETWORKS

Muneeb Ali  
*Computer Science Department,  
Lahore University of Management Sciences (LUMS),  
Opposite Sector U, DHA, Lahore - 54792, Pakistan  
muneeb@sics.se

Thiemo Voigt  
*Swedish Institute of Computer Science (SICS),  
Box 1263, SE-164 29 Kista, Sweden  
thiemo@sics.se

Zartash Afzal Uzmi  
*Computer Science Department,  
Lahore University of Management Sciences (LUMS),  
Opposite Sector U, DHA, Lahore - 54792, Pakistan  
zartash@lums.edu.pk

Abstract  In sensor-nets the individual sensor nodes are generally assumed to be static. However, some recent applications of sensor-nets (e.g. in medical care and disaster response) make use of mobile sensor nodes, which poses some unique challenges to sensor-net systems researchers. In this paper we address the issue of mobility management from a sensor-net architecture point of view. While in current practice individual applications and protocols at different layers obtain, store and manage mobility information we propose to reduce this redundancy by storing mobility information in a logically centralized database that is visible across all layers.

Keywords: Wireless Sensor Networks, Mobility Management, Cross-layer Service

1. Introduction  

Sensor networks have the potential to revolutionize medical care and disaster response amongst others. However, there is a significant gap
between the current sensor network technologies and the special needs of medical care - one of these is the need to handle mobility of nodes. The applications of sensor networks in medical care and disaster relief [1–2] call for re-thinking of protocol design as there is a need to better adapt the current protocols to mobility. The assumption of static sensor nodes, generally made in sensor-net research, is no longer valid in medical care (e.g. sensors attached to doctors or first responders). In this paper we address the issue of mobility management from a sensor-net architecture point of view and provide a framework for handling mobility. Our design is primarily motivated by medical care and disaster response applications of sensor-nets, but it is generic enough that it could be applied to any mobile sensor network.

The traditional (conceptual) network protocol stack for sensor networks [3] is given in Figure 1. However, unlike the Internet this protocol stack is not strictly enforced in most sensor networking architectures. There are no standardized layer boundaries and research groups have produced monolithic solutions that cut across the boundaries of different layers.

Culler et al. [4] argue that the lack of an overall sensor network architecture is the primary factor limiting research progress in sensor-nets because of reduced synergy between different research efforts. One of the early encouraging steps towards a sensor network architecture is the sensor-net protocol (SP) [5], a flexible alternative to the recent ZigBee standard. SP sits between the data-link and network layer and provides a standardized interface to medium access control (MAC), with some feedback in both directions. We consider SP as an important step in
realizing a larger sensor-net architecture. Developing a sensor-net architecture would be a growing and organic process. In this paper we discuss the open issue of addressing mobility management in sensor-nets in the general context of moving towards a sensor-net architecture. We argue that there is a need to store mobility information into a database visible across all layers since in many scenarios, in particular in medical care and disaster response, both the application, network layer services and medium access control layer require mobility information. Currently these layers obtain, store and manage this information individually. The central database proposed in this paper enables sharing of this information across all layers leading to more efficient resource usage and potentially more precise mobility information.

We discuss related work in Section 2, present our mobility management framework in Section 3, discuss simulation and implementation issues in Section 4, and present conclusions and future work in Section 5.

2. Related Work

Internet solutions generally do not apply to sensor-nets, but their underlying techniques do. Our mobility-management framework is motivated by Snoeren et al.’s concept of cross-layer visibility as a service in the Internet [6].

The research community generally ignores mobility in sensor-nets because sensor-nets were originally assumed to consist of static nodes. However, recent efforts such as RoboMote [7] and Parasitic-Mobility [8] have enabled mobility in sensor-nets. Moreover, there has been an increased interest in medical care and disaster response appli-
cations of sensor-nets, sometimes referred to as “body sensor networks”, and these environments make use of mobile sensor nodes e.g. sensors attached to doctors or first responders. AID-N [9], SMART [10], and CodeBlue [1–2] are amongst several projects targeting medical care using sensor-nets. We believe that research in the area of body sensor networks is at a relatively early stage with the main focus being on the development of hardware technologies that enable medical applications [11]. To the best of our knowledge CodeBlue [1–2] is the only initiative that has comprehensively addressed sensor-nets for medical care and proposed a software architecture. However, even the CodeBlue architecture does not explicitly address mobility and we are not aware of any work in the general sensor-net literature that presents a mobility-management architecture framework. SP’s unifying link abstraction [5] and the mobility-management cross-layer service presented in this paper could be integrated into the larger architecture of CodeBlue [2] to provide better mobility handling and to enable efforts from different research groups to inter-operate with each other.

3. Mobility-management Framework

We propose a “bow-tie” architecture (Figure 3) for providing cross-layer mobility information in the general sensor-net architecture. The left side of Figure 3 illustrates the sensor-net protocol stack with SP [5] as the “narrow waist”. SP provides a unifying link abstraction to the data
link layer and allows applications and network protocols running on top of SP to become independent of the underlying data link and physical radio technologies. The center of Figure 3 shows the cross-layer database that is populated from mobility information gathered by the left-side network protocol stack. The cross-layer database is implemented as a shared buffer and provides services to the management applications that form the right side of the bow-tie in Figure 3. The cross-layer mobility-information database is populated on-demand of the concerned architecture components. The information update depends on the policy mechanism and such information could either be updated in a “lazy” best-effort manner or the sensor node could actively request such information depending on the individual needs of the concerned applications.

The mobility information could be required by sensor-net medical care applications (e.g. monitoring physical movement of depression patients), network layer services (e.g. neighbor discovery and route maintenance), and medium access control layer (e.g. mobility-adaptive MAC [12]). Therefore, instead of exporting information between layers it is more useful to import mobility information into a separate management database visible across all layers. To allow solutions from different research groups to inter-operate we need to standardize what goes into the database. This allows network technologies and management applications to evolve independently over time while preserving the interface between them.

Our mobility-management design does not take any stance on time synchronization and can work with any underlying time synchronization mechanism e.g. [13] or [14]. Like SP [5] our mobility-management framework does not take any stance on naming, but assumes that accessible nodes have unique addresses. For details on naming issues see [15].

In the reminder of this section we first present the AR-1 mobility estimation model we intend to use in our architecture, then discuss location issues as well as the energy costs for local dissemination of mobility information.

### 3.1 Mobility Estimation

We intend to use the AR-1 model [16] for mobility estimation. The AR models are autoregressive models used for estimations. Zaidi and Mark have shown that these models can be used to estimate mobility in wireless networks [16]. In the AR-1 model the mobile node’s state, at time t, is defined by a column vector.

\[
s_t = [x_t, \dot{x}_t, \ddot{x}_t, y_t, \dot{y}_t, \ddot{y}_t]',
\]
where \( s_t \) is the mobility state, \((x_t, y_t)\) specify position, \( \dot{x}_t \) and \( \dot{y}_t \) specify velocity, notation \( t \) specifies the matrix transpose operator, and \( \ddot{x}_t \) and \( \ddot{y}_t \) specify the acceleration in the x and y directions. The AR-1 model [16] gives,

\[
s_{t+1} = As_t + \omega_t,
\]

(2)

where \( A \) is a \( 6 \times 6 \) transformation matrix, the vector \( \omega_t \) is a \( 6 \times 1 \) discrete-time zero mean, white Gaussian process with autocorrelation function \( R_\omega(k) = \delta_k Q \), where \( \delta_0 = 1 \) and \( \delta_k = 0 \) when \( k \neq 0 \). The \( Q \) is the covariance matrix of \( \omega_t \). The values for matrix \( A \) and the covariance matrix \( Q \) are estimated based on training data using the Yule-Walker equations [17]. See [16, 18] for details on mobility estimation.

The mobility state information \( \hat{s}_t \), at a given time \( t \) can be used to predict the mobility state at any time \( t+i \). The optimal predicted state \( \hat{s}_{t+i} \) of the mobile node in the minimum mean-square error (MMSE) sense is given by,

\[
\hat{s}_{t+i} = A^i \hat{s}_t
\]

(3)

More accurate mobility estimation could be obtained by using the AR-3 estimation model [19] instead of the AR-1 model, but we believe that using the computationally intensive AR-3 model on memory-constrained sensor nodes is not feasible from a practical point of view. However, the choice of the estimation model, and its effect on different performance metrics, remains an open area for future work.

### 3.2 Localization Issues

Accuracy of mobility estimation depends on the accuracy of the underlying localization mechanism. Localization is a well-studied problem in wireless sensor networks and it has been shown that many multi-hop localization algorithms provide quite accurate results in simulation [20–22]. Others are trying to bridge the gap between simulation and real world performance of localization algorithms [23]. Further, some effort is performed on localization for mobile sensor networks [24]. A detailed discussion of localization algorithms is beyond the scope of this paper.

### 3.3 Energy Costs

With advances in low-power computing chips, the wireless interface will be the primary consumer of energy in any device that combines computation and radios. Therefore, we primarily focus on communication energy costs while evaluating the energy costs of our mobility-management design. In the AR-1 model, self mobility can be estimated
without any communication. However, for the mobility information to be useful to applications (e.g. network layer topology management) any node would also need information on all neighbor nodes’ mobility estimation. Every node could perform local processing and instead of sending out raw location values each node could transmit only the final locally-calculated predicted location information, but even such predicted future location would need to be communicated at regular intervals. Hence, there is a need of a cost-benefit evaluation to determine if it is worth expending energy on such mobility information.

4. Simulation and Implementation Discussion

Assumptions made in most simulation environments for wireless networks, e.g. a radio’s transmission area is circular, all radios have equal range etc., do not necessarily reflect the real-world conditions [25]. In order to fully understand the complexity from a system perspective and to develop solutions that work in real life, it is necessary to not only model or simulate, but also to implement and test on real world systems.

We are currently implementing SP [5] and the mobility-management cross-layer service described in this paper in the Contiki [26] operating system using the Protothreads [27] library. From our prior experience we have found Protothreads to be extremely useful in reducing the complexity of event-based programming [27]. For simulations we are using the COOJA simulator [28] developed for the Contiki OS. Using the COOJA simulator and programming in standard C for simulations greatly reduces the time to map the code, written for our simulations, to real deployments running Contiki on Telos Motes [29]. Furthermore, for mobility evaluations we are implementing realistic mobility models [30] and are using comprehensive real mobility traces (collected by some research efforts [25, 31]) for our evaluations.

5. Conclusions and Future Work

The mobility-management framework presented is ongoing work and in this paper we have presented our design and provided relevant discussions for early information dissemination. We believe that SP [5] is an encouraging step towards a sensor-net architecture and that the sensor-nets community should make use of SP with mobility-management as a cross-layer service to provide a standardized yet flexible framework for future research in the area.
Acknowledgments

We thank Koen Langendoen for valuable feedback that helped to improve this paper.

References


