Engineering challenges in building sensor networks for real-world applications

Kaustubh Kulkarni, Sameer Tilak, Kenneth Chiu and Tony Fountain

kaustubh@me.umn.edu, sameer@sdsc.edu, kchiu@cs.binghamton.edu, fountain@sdsc.edu

Abstract

Wireless sensor networks can be embedded within complex environments for a wide range of monitoring and control applications including study of fire spread in woods, contamination spread in water bodies, or diffusion of toxic gases through air. In these applications, the collected data is used to drive a forecasting model, typically consisting of a CFD simulation. Traditionally, the data collection process is fixed. Active coupling between the sensor and the model, however, can significantly improve the accuracy, timeliness, and efficiency of forecasting. While there has been significant work in this area, there has not yet been a systematic analysis of how to represent the complex environments, how to represent the model, and how to architect the system. In this paper, we present a generalization of data-coupling applications, and describe a framework for decomposing models, which we apply to a simple example of water contamination.

1. Introduction

By embedding thousands of sensors within a physical environment, sensor networks promise to revolutionize a wide range of monitoring applications for detecting chemical, biological, and physical threats. For example, sensors can be deployed in the wild to monitor ecological events in migration patterns [1], or to track a smoldering forest fire for conditions that might lead to an outbreak. In-situ and fine-grained measurements collected by sensors can provide an unprecedented insight into the physical environment. These measurements can revolutionize our understanding of the ecosystem and improve our ability to predict climate variations.

Scientists typically use data collected by the sensor network, in conjunction with a model, to produce a forecast. The model is a representation of the environment within which the monitored threat develops.

The data collection and model can interact in different ways, each with different accuracy and efficiency implications. For example, the data collection can be performed completely independent of any model, ignoring any environmental complexities – sensors would be located and operated uniformly without regard for the environment. This greatly increases power consumption, since some sensors may be redundant, and some sensors may be powered on when not necessary.

This suggests that coupling the modeling to the data collection can improve the energy efficiency of the forecasts produced by such a sensor network application. While there exists a significant amount of research in various aspects of sensor network design, from hardware components [2], [3], [4], [5], [6], operating system [7], networking protocols [8], [9], [10], [11], [12] to middleware [13], [14]; there has been less attention focused on coupling the modeling of the phenomenon being sensed to the actual sensing procedure and architecture. Also, currently during design and evaluation of existing sensor network protocols various network related aspects including network topology, congestion, and radio model are considered in great details [10], whereas the details of the domain are mainly abstracted away. We argue that this overlooks a signficant source of efficiency gains.

Recently, there has been some interest in integrating engineering models for structural monitoring [15], [16]. Recently Terzis et al. [16] proposed using a Finite Element Model that predicts whether and when a landslide will occur. Sheth et al. [15] studied the problem of landslide detection. One of the techniques they mention needs the base station to collect strain readings from all the sensor nodes. Once it has these readings, then it can quite easily predict if a landslide is likely to happen. The other techniques they propose predicts a landslide by using the well- known statistical inference technique of Bayesian hypothesis testing.

The main difference between our work and these works is the fact that the focus of these studies is just to model the phenomenon (e.g., landslide), whereas we focus on modeling both the phenomenon and the physical environment/domain in which the sensors are embedded and the phenomenon occurs. More specifically, we argue that a sensor network protocol should take into account both the model of the phenomenon as well as the model of the physical environment.

The contributions of this paper are as follows. (1) We analyze several different techniques for combining data with model simulations, thus providing a general framework for their research. (2) We identify several key challenges associated with the complex physical domains that can have significant impact on the feasibility and performance of sensor network protocols; as a vehicle, we use domains which can be modeled using Navier-Stokes equations. (3) The methodology used in this paper is based on solid engineering foundations and it can be used to generate synthetic datasets (evaluation benchmarks) for evaluating performance protocols under realistic scenarios.

2. Model Representation - Physical Domain

In this section we describe how our ideas may be applied to a real problem: that of monitoring a chemical spill in a waterway. We believe that this example incorporates aspects common to many situations such as spreading of fire, diffusion of toxic gases through air etc. where sensor networks are used. We first provide some background, and then describe the system architecture.

A. Background

Conventionally, fluid flow is modeled using Navier-Stokes equations, which describe the net force balance on a differential control volume of fluid element (Eq. 1). A control volume is any imaginary three dimensional region in space over which conservation of mass, momemtum and energy are applied to calculate fluid properties such as density, velocity, temperature etc. The boundary of control volume is called control surface.

$$\frac{\partial}{\partial t}(\rho\mathbf{V}) + \nabla\cdot\rho\mathbf{V}\mathbf{V} = \rho\mathbf{f} + \nabla\cdot\mathbf{\Pi}_{ij} \tag{1}$$

Where, ρ is fluid density and $\mathbf{V}$ is fluid velocity in this vector equation. The first term represents the time (t) rate of momentum increase per unit volume inside the control volume. The second term is the rate of momentum lost by convection per unit volume through control surface. This net change in momentum is equal to the sum of body forces ($\mathbf{f}$;e.g. gravity) acting on control volume and stresses (normal and shear) ($\mathbf{\Pi}_{ij}$) acting on the control surface. A constitutive relationship exists between the shear stress and shear strain for all fluids. For a Newtonian fluid, the shear stress is proportional to strain rate and can be expressed in terms of the spatial velocity gradients. This factor of proportionality is called coefficient of viscosity (μ), which is constant for liquids at constant temperature and for gases at moderately varying temperature. Liquids are incompressible and though gases are compressible, gaseous flows at low speeds can be reasonably approximated as incompressible flows ($\rho = const.$). These simplifications result in the incompressible unsteady Navier-Stokes equation (Eq. (2)). The stress term ($\mathbf{\Pi}_{ij}$) is represented as summation of normal stress due to pressure (p) and shear stress due to fluid viscosity (μ).

$$\rho\left(\frac{\partial\mathbf{V}}{\partial t} + \mathbf{V}\cdot\nabla\mathbf{V}\right) = \rho\mathbf{f} - \nabla p + \mu\nabla^2\mathbf{V} \tag{2}$$

Equation (2) accurately models a wide range of physical phenomena such as air flow around aircrafts, river flows, oceanic and atmospheric currents etc. Passive contamination such as petroleum spill in water can be incorporated in this model using the flow field solution and diffusive properties of the contamination material. Reactive flows such as fire in woods incorporate additional dynamics (chemical kinetics), but still require modeling of the fluid behavior.

It is important to understand the mathematical properties of Navier-Stokes equation since (i) they determine the constraints to be fulfilled in order to solve the equation and (ii) the properties of the solution. For example, the unsteady Navier-Stokes equation (Eq. (2)) is mathematically a parabolic partial differential equation (PDE) [17] which has a unique solution in time $t \geq t_0$ for a given physical domain, when an initial condition at time $t = t_0$ over that domain and boundary conditions at $t \geq t_0$ are specified. Initial condition is the flow field description (ρ, $\mathbf{V}$ etc) of the domain at $t = t_0$, whereas boundary conditions are the flow field descriptions at the geometrical boundaries of the domain specified at all times $t \geq t_0$. Time evolution of oil tanker spill in a river and toxic gas diffusion inside a building belong to such unsteady process and are dependent on the initial and boundary conditions. Steady flows can be represented by Eq. (2) when time derivatives of velocity are considered zero. Mathematically this equation can be closely approximated as an elliptic equation and solution to such equation depend only on the boundary conditions of the domain. A river flow in an urban region where many industrial and residential sewer waste channels join the main stream can be modeled as a steady state phenomenon on an average basis.

Fluid flow can be classified according to the flow characteristics. For steady flow, the change in fluid momentum (inertia force) is a resultant of body forces acting on control volume and pressure and shear forces acting on the control surface. A convenient non-dimensional parameter called Reynolds number (Re) defined as the ratio of inertia to shear forces is used to define flow characteristics namely, laminar and turbulent flows. A laminar flow is a smooth and deterministic flow whereas a turbulent flow is characterized by random fluctuations in velocity, pressure etc. within the flow. A low Reynolds number ($Re \leq$ 2300 for channel) typically implies laminar flow such as a calm river flow whereas a high speed air flow around civil structures in a city is highly turbulent. The unsteadiness in turbulent flow enhances mixing of contamination within the flow in comparison to that in laminar flow which is due only to molecular diffusion. Since turbulent flows show nearly random variations in flow field quantities, typically statistical methods are employed to calculate the average quantities.

As mentioned earlier, an unsteady Navier-Stokes equation (NS) belongs to parabolic PDE and would require an initial condition (IC) and boundary condition (BC). A laminar flow solution of an unsteady flow is highly dependent on both of these conditions; whereas in case of fully developed turbulent flow, due to its statistically random fluctuations, it is dependent strongly on the local BC and weakly on the IC. This allows some tolerance on the accuracy of IC in case of turbulent flow and hence, IC can be considered a flexible constraint. A steady NS equation, belongs to

TABLE 1: MODELS AND SOLUTION CONSTRAINTS REPRESENTING FLUID FLOWS

Physical Example	Mathematical Model	PDE	Constraints
Time Evolution of oil spill in river	unsteady NS eqn	Parabolic	BC,IC
Industrial drain outlet in river	steady NS eqn	elliptic	BC

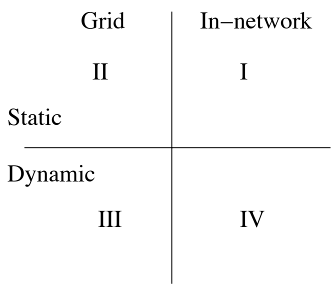


Fig. 1: Data collection techniques and models

elliptic PDE and requires only BC for both laminar and turbulent flows. Again, laminar flow solution is critically dependent on the BC at all locations upstream and downstream of the calculation point inside the domain; whereas a turbulent flow solution is strongly dependent on the local BC and weakly dependent on the region far upstream or downstream. This weak dependence exhibited by turbulence flow towards ICs and BCs far away upstream is typically referred to as ”low memory” of turbulence flow. Table (1) summarizes the mathematical models and their constraints based on the flow type.

3. MODEL REPRESENTATION - SENSOR NETWORK

There are a number of different ways that modeling information can be used to direct data collection, as shown in Figure 1. We divide these along two axes. The X axis denotes whether the model is computed statically, prior to sensor deployment, or whether it is computed dynamically, driving the data collection in real-time. The Y axis denotes where the model is being computed. At one extreme, the model is simulated on high-end computational resources, i.e. Grid in fig. 1 external to the sensor network. At the opposite end, the model is simulated completely in the network itself with limited processing capabilities. We now briefly describe the energy and accuracy tradeoffs that arise in these quadrants.

The solution of Navier-Stokes equations depends on the initial and boundary conditions as well as the flow type - laminar or turbulent flow. The knowledge of the solution characteristics is critical for effective and efficient sensor networks. For example, the contamination threat in case of turbulent flow will be experienced over a larger area than in case of laminar flow for a given domain. Similarly, the low memory of turbulent flows can be used as an advantage towards modeling smaller domains for flow simulations resulting in higher efficiency and less computational time. Intelligent coupling between physical model and sensor network model will depend on the situation and four scenarios according to Fig. 1 are discussed here.

- Quadrant II - This is the simplest of the four models. It consists of a pre-programed single solution scenario for the physical process. The known solution guides the placement of the sensors over the entire domain. The sensor network is used mainly as a detection tool for the threat being monitored. An indication of threat level above a prescribed threshold activates the warning systems and pre-approved contingency plan is executed. This model is useful only for simple situations such as building fire alarms. The known ventilation system and air flow through the building can be used to predict the likely region exposed to fire threat apriory. This solution does not take into account the exact source of fire and hence, typically is a conservative solution. This model is known to be very inefficient and less economical.
- Quadrant III - This is the best of the four models. It consists of a resource rich computational grid connected to a sensor network by fast and reliable communication system. This model is capable of handling real time simulations of the physical process based on the data sensed and forcast the results of threat propogation. The necessary contigency actions are communicated to the sensors through fast communication network for contingency actions. This process can be iterated to optimize the sensor utilization and computing efforts. A high resolution fluid dynamics model simulation based on the real time velocity, pressure and contamination data supplied by the sensors can be used to forcast the contamination spreading path and activate the corresponding sensors only. However this model is computationally expensive and depends crucially on the communication speed between the grid and sensor network.
- Quadrant I - A model based on in-network decision capability is necessary to overcome the limitation imposed by communication delay between the sensor network and central computational grid. In this type of model, sensors are equiped with low level processing ability and stored information about solutions of more than one likely scenarios of threat propogation. A decision based algorithm is implemented locally in the network to pick the appropriate solution from a library based on the information sensed and is implemented locally. The solution library might contain information according to flow type such as laminar or turbulent, likely locations of con-

tamination sources, etc. which are specific for the physical phenomenon modeled. The sensor will sense the velocity fluctuations over a prescribed time interval to determine the turbulence intensity in the flow. Comparing the measured turbulence intensity with a prescribed threshold value would direct the sensor to select the appropriate turbulence flow model from the library. Though this model will be much economical than QIII and efficient than QII, it is capable of handling limited cases only.

- Quadrant IV - A sensor network coupled with the knowledge and capability of modeling the appropriate flow situation is necessary for economical working and possible optimization of the sensor network. In this type of model, the sensor is equipped with moderate level of processing ability like QI. However, instead of implementing a library solution, a simplified model (low resolution, simple and hence less approximate) is used to forcast the threat propogation. A good knowledge of physical phenomenon such as requirements to model the flow and properties of solution can be used intelligently to achieve this faster and economical computation. Thus, a local action based on the results can be initiated.

4. A Case Study

Considering a specific case study, we now describe various challenges that a sensor network system would need to consider. Specifically, we demonstrate the effect of flow type and boundary conditions for a two-dimensional steady channel flow depicting a straight section of river flow on the design of network protocols. These examples are used just for demonstrating our point, and not as an end in themselves.

Figure (2) shows the computational domain and boundary conditions prescribed for the different cases. Figure (2A) is a river flow bounded by straight rigid banks whereas figure (2B) shows additional flow effluxing from one wall modeling series of industrial and civil drainage outlets along the river side. A pressure difference between inlet and exit of the channel ($P1 - P2$) drives the flow in laminar mode. An initial 10% (of free stream velocity) turbulent intensity (TI) is imposed in case of turbulent flow simulation for both of these physical cases. A mercury contamination source is located as shown in Fig. (2). The effect of flow type and boundary conditions on steady state solution of mercury concentration throughout the domain is studied. We also discuss the implications of this solution on the efficiency/feasibilty of networking protocols.

We used FLUENT [18] to solve the flow field and mercury contamination concentration is calculated using discrete phase model in FLUENT.

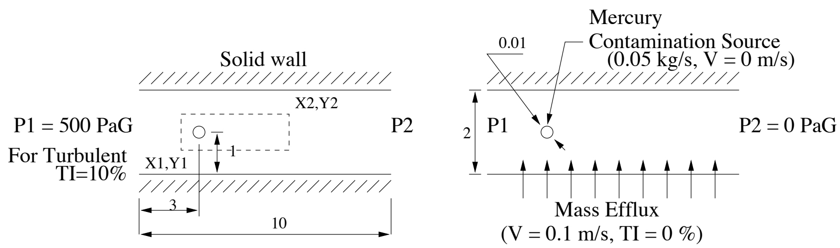
1) Effect of Flow Type: Figure pairs (3(a)),(3(b)) and (3(c)), (3(d)) show the effect of flow type on the mercury concentration distribution for identical boundary conditions. It is clear that turbulent flow enhances macroscopic mixing in comparison to laminar flow. Thus, for an equal strength of contamination source, turbulent flow spreads the contamination over a larger area reducing the concentration levels at locations affected by contamination in case of laminar flow. Clearly, in the case of turbulent flow (3(b)), sensors across a larger area need to be turned on as compared to the laminar flow case(3(a)). Without accurate knowledge of flow type (or even for systems built in Quadrants I, II, or III), localized real-time decisions made by individual sensors, regarding selection of sensor mode and reporting frequency would be either inefficient (assuming turbulent flow when it is laminar) or inaccurate (assuming laminar flow when it is turbulent). As a proof-of-concept, we now describe how a typical framework (built in Quadrant IV) that could embody the model and data collection via a novel rule based engine (shown in Algorithm 1). The strength of turbulence is characterized by turbulence intensity which is the ratio of velocity fluctuations to its mean value; greater the turbulence intensity, higher is the mixing. A sensor can sample the water velocity and use turbulence intensity as an indicator of flow type. A sensor upon detecting its flow type and physical coordinates can periodically sample water for detecting presence of contaminant. If it is situated within (X_1, Y_1, X_2, Y_2) rectangle (Fig.(2)), upon detecting contamination concentration ($C > Threshold$), takes different actions depending upon flow type. For example, if the flow is laminar, it wakes up only its immediate horizantally downstream sensor. Whereas, for turbulent flow, it wakes up all of its neighboring sensors.

Algorithm 1 Planner Algorithm

if $C > Threshold$ and $loc \in (X_1, Y_1, X_2, Y_2)$ **then**
  if $turbulent\_flow()$ **then**
    $wakeup\_all\_neighboring\_sensors()$
  else if $laminar\_flow()$ **then**
    $wakeup\_horizonal\_down\_stream\_sensor()$
  end if
end if

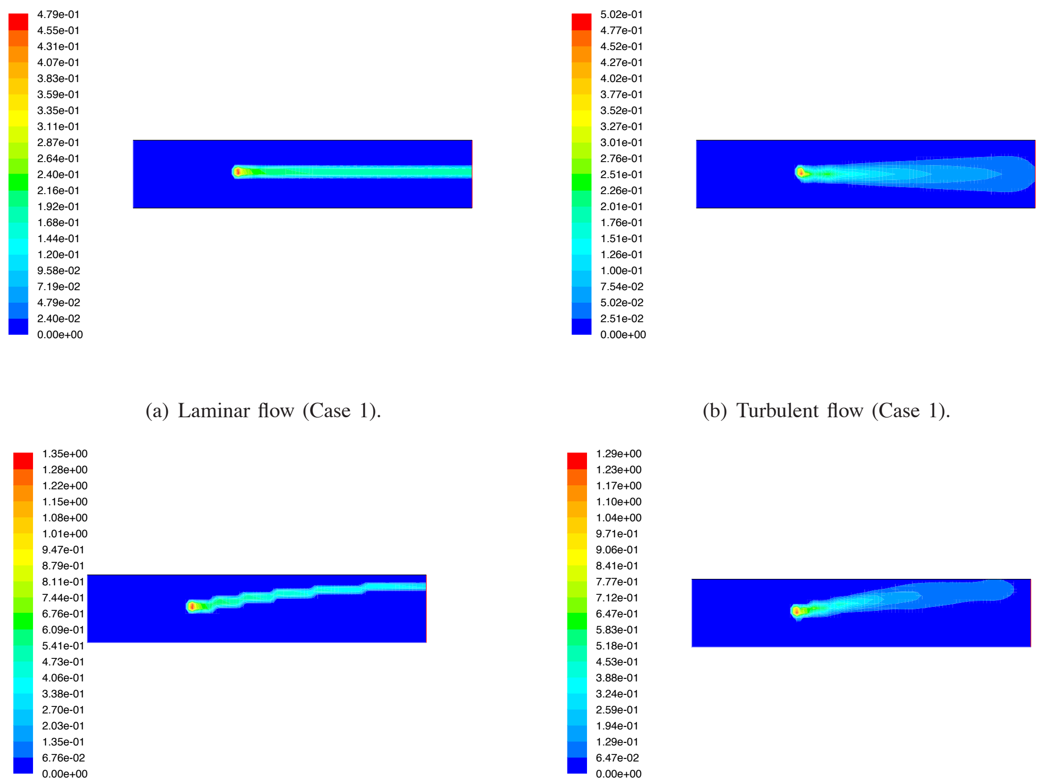
For various reasons including, accurate micromodeling, phenomenon bounbdary tracking and distributed sensor coordination, it is essential to turn on right set of sensors and at the same time to conserve energy it is important to turn off sensors whose data is not required.

2) Effect of Boundary Conditions: The solution of Navier-Stokes equation is critically dependent on the boundary conditions. It is clearly seen (Fig. pairs (3(a)), (3(c)) and (3(b)),(3(d))) that the initial flow structure of regular channel type flow changes throughout the domain due to the mass inflow from lower boundary. Note that the jaggedness in Fig. (3(c)) is due to the coarse computational grid used in the simulation. Clearly, without accurate knowledge of boundary conditions, localized real-time decisions made by individual sensors regarding selection of sensor mode, operation, reporting frequency will be either inefficient or inaccurate. It is important to know the physical



A. Test Case 1 B. Test Case 2

Fig. 2: Computational Flow Domain [dimensions in m]



(a) Laminar flow (Case 1). (b) Turbulent flow (Case 1).

(c) Laminar flow (Case 2). (d) Turbulent flow (Case 2).

Fig. 3: Mercury Concentration (kg/m^3) for various flow and boundary conditions

boundary conditions of the domain very accurately even though the area of interest (dotted rectangle in Fig (2)) might be much smaller. Building boundary condition knowledge on the fly might require a sensor to have global knowledge about physical domain (e.g. presence of a downstream drainage), which poses a significant challenge in terms of designing scalable and energy efficient protocols. Similar to flow conditions, we can adapt Algorithm 1 to incorporate boundary conditions within a rule-based engine.

5. Implications of technology choice

Despite the relatively recent emergence of sensor networks as a field of study, already a large number of sensor hardware platforms and software elements (operating systems, networking protocols, data base systems, etc.) have emerged. Sensor hardware platforms vary in capability from miniature sensors such as the Berkeley motes [19], which are equipped with 8-bit microprocessors with a few kBytes of memory and low bandwidth, low range radios, to PDA class sensors such as PASTAs [20]. Other sensor platforms include Mantis [21], [22], and WINS [23]; Table 2 presents a comparison of various sensor platforms.

Choice of technology has a clear implication on how modeling information can be used (ref. Figure 1). For example, for a sensor network composed of resource-constrained mica nodes an appropriate choice would be Quadrant I or Quadrant II. Note that Quadrant IV would not be a reasonable choice due to its higher power and computation requirements. Again, the de-

TABLE 2: SENSOR PLATFORMS.

Platform	MICA	Mantis Nymph	RockWell WINS	Stargate
Processor	ATMega103L	ATMega128L	Intel-StrongArm	Intel-Xscale
CPU speed	4 MHz	7.37 MHz	133 Mhz	400 MHz
RAM	4 kB	4 kB + 64 kB	1MB	64 MB
Max. Data Rate	40(kb/s)	76.8 (kb/s)	100 (kb/s)	11(Mb/s)
Operating system	TinyOS	MOS	Win CE	Linux

cision between use of Quadrant I and Quadrant II is application specific. For a sensor network consisting of more powerful nodes including Stargate and WINS nodes, choice of Quadrant IV is feasible along with Quadrant I and II[1]. Therefore, in this setting, the sensor node can solve the CFD.

6. CONCLUSION AND FUTURE WORK

In this paper, we show that coupling the model simulation with the data collection via in-network processing leads to significant benefits. We believe that this is an essential step for building efficient and effective sensor network protocols. In particular, in an unattended sensor network, if sensors have no knowledge of domain properties, their localized decisions can diverge significantly from optimal decisions. However, incorporating domain knowledge efficiently itself is a challeging problem. To this end, we are planning to develope a middleware that will provide necessary abstraction and shield applications from the complexity of physical domains which can be modeled using Navier-Stokes equations. We believe that the middleware will help rapid application develpoment and deployment to a great extent.

Also, if either the flow type or boundary conditions are known with uncertainity, either due to lack of detailed domain knowledge or due to highly dynamic environment, or due to measurement errors, incorporating this stochastic behavior in protocol design is an interestingly challenging problem.

7. ACKNOWLEDGMENTS

This work was partially supported by a grant from the Gordon and Betty Moore Foundation and the following NSF grants OISE 0314015, OCI 0722067, 0627026, 0446802, 0446017, 0446298, and DBI 0639229.

REFERENCES

[1] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," in *In Prof. of ASPLOS 2002*. ACM Press, 2002.

[2] G. Asada, M. Dong, T. Lin, F. Newberg, G. Pottie, and W. Kaiser, "Wireless integrated network sensors: Low power systems on a chip," in *European Solid State Circuits Conference*, Oct. 1998.

[3] A. Chandrakasan, A. Amirtharajah, S. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang, "Design considerations for distributed microsensor systems," in *Proc. of the IEEE 1999 Custom Integrated Circuits Conference (CICC'99)*, May 1999.

[1] In the current setting with a 2.8 GHz Pentium-IV machine with 512 MB memory the solution time was approximately 5-10 minutes.

[4] A. Sinha, A. Wang, and A. Chandrakasan, "Algorithmic transformations for efficient energy scalable computation," in *Proceedings of the International Symposium of Low Power Electronics and Design (ISLPED'00)*, 2000.

[5] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design and Test of Computers*, vol. 18, no. 2, Mar. 2001.

[6] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking*, July 2001.

[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for network sensors," in *Proc. of ASPLOS 2000*, Nov. 2000.

[8] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, "Protocols for self-organization of a wireless sensor architecture," *IEEE Personal Communications*, vol. 7, no. 5, Oct. 2000.

[9] S. Tilak, N. Abu-Ghazaleh, and W. Heinzleman, "Infrastructure tradeoffs in sensor networks," in *ACM Workshop on Sensor Networks (WSN'02)*, 2002, (submitted).

[10] C. Wan, S. Eisenman, and A. Campbell, "CODA: Congestion detection and avoidance in sensor networks," in *SenSys 2003*.

[11] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," in *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, Aug. 1999.

[12] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient routing protocols for wireless microsensor networks," in *Proc. 33rd Hawaii International Conference on System Sciences (HICSS '00)*, Jan. 2000.

[13] T. A. et al., "Envirotrack: Towards an environmental computing paradigm for distributed sensor networks," in *IEEE International Conference on Distributed Computing Systems*, March 2004.

[14] A. Murphy and W. Heinzelman, "Milan: Middleware linking applications and networks," November 2002. [Online]. Available: citeseer.ist.psu.edu/murphy02milan.html

[15] A. Sheth, K. Tejaswi, P. Mehta, C. Parekh, R. Bansal, S. Merchant, T. Singh, U. B. Desai, C. A. Thekkath, and K. Toyama, "Senslide: a sensor network based landslide prediction aystem," in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2005, pp. 280–281.

[16] A. Terzis, A. Anandarajah, K. Moore, and I.-J. Wang, "Slip surface localization in wireless sensor networks for landslide prediction," in *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*. New York, NY, USA: ACM Press, 2006, pp. 109–116.

[17] J. Tannehill, R. Pletcher, and D. Anderson, "Computational fluid mechanics and heat transfer," 1997.

[18] "FLUENT," http://www.fluent.com.

[19] "Crossbow – smart sensors in silicon (crossbow website)," 2003, commercial product based on Berkeley MICAs http://www.xbow.com).

[20] "Pasta microsensor user's guide," (http://pasta.east.isi.edu/documentation/users_guide/).

[21] "Mantis project," (http://mantis.cs.colorado.edu/index.php/tiki-index.php.

[22] H. Abrach, J. Carlson, H. Dai, J. Rose, A. Sheth, B. Shucker, and R. Han, "Mantis: System support for multimodal networks of in-situ sensors," in *Technical Report CU-CS-950-03, Department of Computer Science, University of Colorado*, April 2003.

[23] "Rockwell science center sensor network project," 2003, (http://www.rsc.rockwell.com/wireless_systems/sensorware).