

RESEARCH STATEMENT

MURAT DEMIRBAS

The traditional computer-centric view of computing, that forces us to interact with the computers in their terms and in their virtual world, is rapidly losing ground to a human-centric view of computing, that enables computers to interact with us in our terms and in our physical world. The push for this paradigm shift to *ubiquitous computing* (also known as *cyber-physical systems*) is fueled by the maturity of integrated circuits miniaturization technology, progress in MEMS-based sensor and actuation technology, and availability of inexpensive low-power radio technology. One example that emphasizes the extent of this shift is that only a nominal 2% of the microprocessors produced in year 2000 went to traditional PC platforms whereas the other 98% found its way in to various embedded devices, such as those that control the engines, ABS, and stability & traction of cars, those that empower consumer electronics (microwaves, washing machines, air conditioners, printers, and cell phones), as well as those employed in industrial feedback control systems.

My research interests lie in the area of wireless sensor networks (WSNs), a successful realization of the ubiquitous computing paradigm. Although the technology is quite new, WSNs received an enthusiastic reception in the science community as they enable precise and fine-grain monitoring of a large region in real-time. Successful large-scale deployments of WSNs have been achieved in the domains of ecology monitoring, habitat monitoring, and military surveillance.

This new regime, however, comes with its own set of challenges. A major challenge is the scalability of WSN programs and services. Centralized programs, such as those in the client-server paradigm, do not scale for large WSNs. It is inefficient and unscalable to force each decision of a WSN program to be taken by a centralized basestation, as this would compel the nodes to communicate their data and queries all the way to the basestation. Using long routes for forwarding data not only increases the latency but also depletes the battery power of the relaying nodes quickly. Using the basestation for every decision also leads to a communication bottleneck. Therefore, distributed and local programs are needed to achieve scalability. However, developing such distributed and local programs is notoriously difficult. Another major challenge is the unreliability of communication and that of WSN nodes. Message losses upto 30-50% are not uncommon in WSNs (contrast this with 0.01% message loss rates on the Internet). Similarly due to inexpensive hardware components the nodes are prone to false-positives, incomplete detection, and failure. It is challenging to build consistent and reliable programs on top of these networks.

Motivated by these challenges, my research is focused on developing distributed robust and resilient WSN services and applications. This goal requires work on several layers of the WSN protocol stack as well as development of cross-layer techniques. My research efforts span the following topics:

1. MAC layer protocols for robust single-hop communication
2. Geometric infrastructures for resilient large-scale WSN services
3. Programming abstractions for robust computing
4. Real-world deployments of robust WSN applications
5. Theory of self-healing

Next, I briefly summarize the contributions of my research in these directions.

1 MAC layer protocols for robust single-hop communication

(Partly funded by NSF-CAREER)

Problem: As WSNs get increasingly more integrated with actuation capabilities, consistency and timeliness guarantees become significant issues. However, in the presence of unreliable communication channels—as is the case in WSNs— even the most basic consensus problem of getting nodes agree on a binary decision is unsolvable (due to the well-known Coordinated Attack problem).

Approach: In order to circumvent this impossibility result and provide some consistency properties to WSNs, we propose a receiver-side collision detection (RCD) technique. The intuition here is to provide a way for the system to detect message losses and exploit this information in avoiding inconsistent decisions. While the transmitter cannot detect collisions in WSNs, there is no barrier against the RCD: by periodically performing a “clear channel assessment” in the idle state, a node can detect collisions in the channel successfully.

Contributions: In [1] we have shown, for the first time, that it is possible to solve consensus efficiently in WSNs. To this end, we employed RCD and synchronized rounds across the nodes. The idea behind synchronized rounds is to provide a dedicated round for communicating negative feedback, and hence conveying information even when the negative feedback messages collide. In our algorithm, a collision detected in the veto round indicates the existence of at least one veto message and that the consensus should be deferred for a later round. We have also given a classification of RCD with respect to its completeness (ability to detect collisions) and accuracy (ability to avoid false positives) and identified the lower-bounds for solving consensus for each class.

Another relevant and significant problem for single-hop communication is the reliable broadcast problem. Even though RTS/CTS handshake based solutions exist for eliminating the hidden terminal problems for reliable unicast, these solutions do not directly generalize to solving reliable broadcast problem as they result in a CTS implosion problem at the initiator node. Using the RCD and the composition of negative feedback principle, we have designed two MAC layers for reliable and priority-based broadcast, BEMA [2] and RobCast [3].

Ongoing and future work: We are currently developing reliable implementations and quantitative evaluations of RCD on TinyOS and mote platforms [4]. Our future work will include application of RCDs for singlehop collaboration protocols in WSNs.

2 Geometric infrastructures for resilient large-scale WSN services

Problem: Several WSN services, such as routing, querying, and tracking, require continuous maintenance of distributed data structures, such as trees, paths, and clusters, over a large number of nodes. In order to be scalable these services need to achieve minimal energy consumption and response time regardless of the size of the network. That is, these services should implement local operations over these global structures. For example, the goal of an in-network querying service is to answer spatial queries such as: “What is the location of the nearest enemy tank to my coordinates?” To achieve scalability, only the relevant nodes for the correct execution of a query should be involved in the execution. The goal of tracking service is to enable a pursuer to catch an evader by means of information gathered via the WSN, and similarly, queries by the pursuer and

updates of the structure due to relocations of the evader should be handled as local as possible.

In addition, locality is also needed in handling of the faults. Message losses and corruptions (due to fading, collisions, and hidden node effects) and node failures (due to software/hardware crashes or energy exhaustion) can drive portions of these large-scale structures to be arbitrarily corrupted and to become inconsistent with the rest of the structure. In the absence of a local healing mechanism, faults in one part of the system may contaminate the entire system and hence may result in a high-cost, system-wide correction.

Approach: Our approach in achieving locality and scalability of WSN services is to exploit geometric ideas and techniques while devising distributed network algorithms. In contrast to Internet, where the topology is logical and arbitrary graph models are used, WSNs are deployed in physical spaces and the use of geometric networks are warranted for modeling WSNs. We find that when the problem domain is constrained to geometric networks it is possible to devise simpler and more efficient algorithms than those designed for arbitrary graph topologies. Especially for reasoning about locality of solutions in WSN (where communication cost is the biggest constraint on design) geometric methods are a good fit. In addition, towards enabling locality in handling of faults we propose efficient fault-containment techniques by exploiting the geometry of the network.

Contributions: For in-network querying, we have designed a *distributed indexing structure* [5] that constructs a hierarchical partitioning of a WSN and overlays an approximately balanced search tree over the network. By exploiting the spatial properties of the hierarchical clustering, our protocol achieves $O(d)$ communication cost and time for querying, where d is the distance to the nearest answer. For achieving local healing of our distributed indexing structure, we use the notion of a *tolerance factor* [6]. That is, each cluster in the hierarchical partitioning can tolerate expansion up to two-fold of its ideal size. This way, the changes and failures hitting a region are subsumed locally within that region, and cascading effects, that may require a re-clustering of the entire network, are avoided.

Again, for in-network querying, we have presented a *distributed quad-tree (DQT) structure* [7] that exploits localization information in WSNs. DQT construction is local and does not require any communication. DQT achieves a querying cost of $2\sqrt{2} * d$, and, due to its minimalist infrastructure and stateless nature, DQT shows graceful resilience to the face of failures. In *Glance* [8], we have improved on our earlier results. Our main insight was to use the basestation node in an opportunistic manner for answering of some in-network queries. The knowledge that all queries target the basestation by default, combined with the geometry of the network, was useful in determining the minimum area required for in-network advertisements to satisfy a given distance-sensitivity requirement. We observed that in-network advertisements can safely ignore a majority of directions/regions and focus their advertisement to a small cone to be able to satisfy a given distance-sensitivity requirement. As a result, Glance ensures that a query invoked within d distance of an event intercepts the event's advertisement within $d*s$ distance, where s is a "stretch-factor" tunable by the user. The user may define different stretch-factor requirements (which lead to varying angles for cone advertisement) with respect to the type (i.e., severity) of events. By selecting appropriate values for s it is possible to achieve trade-offs between query execution cost and advertisement cost. Glance is also robust with respect to node failures and holes in the network.

For tracking in WSNs, in *Stalk* [9], we have employed hierarchical partitioning to maintain a tracking structure over a small number of nodes and with accuracy proportional to the distance from the

evader. The idea of maintaining information at farther away nodes with lesser accuracy is achieved by maintaining the tracking path at increasingly higher level clusterheads as the distance from the evader increases. Each node in the tracking path points to a node that is closer to the evader, and hence that has more recent and refined information about the location of the evader. In this structure: 1) Operations to find the mobile object distance d away take $O(d)$ time and communication to complete, 2) Updates to the tracking structure after the object has moved a total of m distance take $O(m * \log m)$ amortized time and communication to complete, 3) The tracked object may relocate without waiting for completion of the updates resulting from prior moves, and 4) The mobile object can move while a find is in progress.

For achieving local healing of the tracking structure in Stalk, we have used *containment waves*. The key idea is to wait for a longer time before updating a wider region's view. This way, more recent (more accurate) updates coming from lower levels can catch-up to and contain misinformed updates at higher levels. As a result, contamination due to faults is restricted to an area proportional to the perturbation size, and our tracking path stabilizes within work proportional to the perturbation size instead of the network size. Furthermore, our solution is such that the latency imposed by waiting for larger timeouts at higher levels of the hierarchical partitioning does not affect the availability of our tracking path; it is still possible to seamlessly track continuously moving objects.

Recently in *Trail* [10], we have presented a tracking protocol that achieves the same linear costs for find and update in Stalk without requiring a hierarchical partitioning of the network. Trail is robust to failures in WSNs and tolerates holes in the network as long as the network is not partitioned.

Ongoing and future work: We are currently working on applying our geometric ideas and techniques for devising distributed network algorithms in the context of mobile WSNs. Recently, in [12], we investigate an energy-efficient and low-latency data collection mechanism for WSNs using a network-controlled mobile basestation (MB). Our scheme progressively relocates the MB closer to the regions that produce high data rates and reduces the average weighted multihop traffic enabling savings in energy. In contrast to the existing solutions where WSN nodes buffer data passively until visited by a MB, our scheme maintains an always-on multihop connectivity to the MB by means of an efficient distributed tracking mechanism. This allows the nodes to forward their data in a timely fashion avoiding latencies due to long-term buffering.

For the deployment and relocation of mobile WSNs, where the goal is to relocate nodes to provide dynamic coverage by following the interest gradient in an area, we investigate efficient, local, and distributed strategies that ensure that the network always stays connected even though the neighbors may change for each node [13].

3 Programming abstractions for robust computing

(Partly funded by NSF-CAREER)

Problem: Effectively managing concurrent execution is one of the biggest challenges for future wireless sensor/actor networks (WSANs): For safety reasons concurrency needs to be tamed to prevent unintentional nondeterministic executions, on the other hand, for real-time guarantees concurrency needs to be boosted to achieve timeliness.

Approach: We propose a transactional abstraction, and an associated optimistic concurrency control

framework for WSANs that enables understanding of a system execution as a single thread of control, while permitting the deployment of actual execution over multiple threads distributed on several nodes. In order to provide a distributed and local conflict detection and serializability, we exploit the properties of wireless broadcast communication.

Contributions: A major contribution of our framework TRANSACT [14, 15] is to simplify the reasoning and verification of a distributed WSANs program. Building blocks for process control and coordination programs (such as, leader election, mutual exclusion, cluster construction, neighborhood discovery, recovery actions, and consensus) are easy to denote using TRANSACT. Also, TRANSACT introduces a novel *consistent write-all* paradigm that enables a node to update the state of its neighbors in a *consistent* and *simultaneous* manner. This paradigm facilitates achieving consistency and coordination and may enable development of more efficient control and coordination programs than possible using traditional models.

TRANSACT is novel in that it provides an efficient and lightweight implementation of a transactional framework in a distributed manner. Implementing transactions in distributed WSANs domain diverges from that in the database context significantly, and introduces new challenges to address. In contrast to database systems, in distributed WSANs there is no central database repository or an arbiter; the control and sensor variables, on which the transactions operate, are maintained distributedly over several nodes. As such, it is infeasible to impose control over scheduling of transactions at different nodes, and also challenging to evaluate whether distributed transactions are conflicting. However, by exploiting the properties of broadcast communication inherent in WSANs, TRANSACT overcomes this challenge and provides a lightweight implementation of transaction processing. Since imposing locks on variables and nodes may impede the performance of the distributed WSAN critically, TRANSACT implements an optimistic concurrency control solution.

Ongoing and future work: We have implemented TRANSACT on TinyOS and mote platforms [15]. As a demonstration of our framework, we will implement a distributed multiple-pursuer/multiple-evader tracking application, employing a 200 node WSN and several iRobot Roomba-Create robots acting as pursuers and evaders. Using TRANSACT, we will implement the consistency critical components of the in-network tracking service, such as evader association and handoff, updating of the distributed tracking directory/structure, and maintenance and recovery of the tracking structure in the face of node failures and displacements. In addition, the pursuer robots will utilize TRANSACT to implement collaborative stalking and cornering of an evader, as well as group membership and intruder assignment among the pursuers.

We plan to integrate verification support to TRANSACT in order to enable the application developer to check safety and progress properties about her program. Since TRANSACT already provides conflict serializability, the burden on the verifier is significantly reduced. Another advantage TRANSACT provides is the simplistic format of the methods, which facilitates translation between TRANSACT methods and existing verification toolkits and model checkers.

4 Real-world deployments of robust WSN applications

4.1 Surveillance applications

Problem: A significant application of WSNs is in the area of intrusion detection and the related

problems of classifying and tracking targets. However, unreliability of sensing and communication are major challenges towards deploying such applications.

Approach: Our approach is based on a dense, distributed, wireless network of multi-modal resource-poor sensor nodes combined into loosely coherent sensor arrays that perform in situ detection and classification using the notion of influence fields.

Contributions: We have successfully developed a “*A Line in the Sand*” surveillance network [16] that achieved detection, classification, and tracking of various types of intruders (such as persons and cars) as they moved through the network. A major contribution of our work is that we do not assume a reliable network; on the contrary, we quantitatively analyze the effects of network unreliability on application performance. Our work includes multiple experimental deployments of over 90 sensors nodes at Mac Dill Air Force Base in Tampa, Florida, as well as other field experiments of comparable scale. Based on these experiences, we identified a set of key lessons and articulated the challenges facing extreme scaling to tens or hundreds of thousands of sensor nodes.

4.2 In-building monitoring applications

Problem: WSNs are instrumental for monitoring temperature, humidity, and illumination of a controlled environment, such as in-building environments. The design goals for such a system are extended deployment lifetime, remote querying and configuration, ease of deployment, and reliability.

Approach: To enable energy efficiency and ease of deployment and querying we propose a single-hop WSN architecture tightly integrated with Web services.

Contributions: We have developed and deployed an in-building monitoring system INSIGHT (INternet-Sensor InteGration for HabitaT monitoring) [17]. Due to our single-hop network decision, the deployment of the network is as easy as turning on a node and dropping it some place for monitoring. Also, without the need for forwarding messages from other nodes, the nodes do not need to stay awake or coordinate to wake up, hence INSIGHT can achieve about 6 months lifetime (using standard AA batteries) by sampling sensors every minute. To enable remote querying, we maintain a webserver and an SQL database at the base station. Users simply type in the base station’s web address into their internet browser to query for data. Data is available for extraction through an XML front-end on the webserver or in the form of a TinyML query. Using the website, sensor data from motes can be visualized, plotted as a graph and compared with each other. Moreover, users can subscribe to get email alerts when a sensor reading has exceeded a threshold configured via the web interface. The client can also reconfigure the monitoring parameters, such as the sampling frequency and the thresholds, through the web interface. INSIGHT was deployed in a greenhouse and the website is accessible at <http://INSIGHT.podzone.net>

Ongoing and future work: We are currently working on deploying an in-building personnel tracking system *Elvis* that builds on the INSIGHT architecture. We are planning to use *Elvis* as a building block for smart-building/smart-office applications.

5 Theory of self-healing

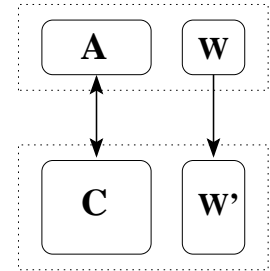
Problem: Since the complexity of software grows drastically with respect to its size, large-scale software systems are extremely error-prone and fail frequently. Again due to their overwhelming complexity, design of self-healing for large-scale software systems remains as a challenging task. Blackbox solutions, such as reset and restart approaches, may be adequate for centralized software systems, but they are inapplicable for distributed software since a reset would mean a global reset of the entire network, and would incur a lot of downtime. Thus, a more efficient and informed approach is needed for achieving scalable design of self-healing with respect to software size.

Approach: Towards addressing the scalability of the design of self-healing with respect to the implementation size, we propose that **scalable design of self-healing can be achieved without knowledge of system implementation but with knowledge only of system specification**. That is, for the design of self-healing we eschew knowledge of system implementation in favor of knowledge of system specification. Since specifications are more succinct than implementations, our **specification-based design of self-healing** approach offers the promise of scalability. Also, since specifications admit multiple implementations and since system components are often reused, a specification-based approach offers reusability. Finally, in contrast to a blackbox approach, a specification-based approach allows the design of low-cost self-healing by virtue of exploiting more information about the system.

Contributions: In order to demonstrate that scalable design of self-healing is achievable via a specification-based approach, we have developed a novel method that enables such a design. Loosely speaking, our method is to first design a *wrapper* component to add self-healing at the system specification level and then to transform this wrapper to the system implementation level by means of a *fault-tolerance preserving and compositional refinement* [18, 19]. Even though the wrapper is designed solely by studying the system specification and not the system implementation, the nature of our transformation enables us to conclude that the transformed wrapper provides self-healing to the system implementation. For example :

Given an abstract system specification \mathcal{A} , we first design a fault-tolerance wrapper \mathcal{W} such that adding \mathcal{W} to \mathcal{A} yields a fault-tolerant system. Our transformations ensure that for any concrete implementation \mathcal{C} of \mathcal{A} , adding a concrete implementation \mathcal{W}' of \mathcal{W} would also yield a fault-tolerant system.

Note that since the refinements from \mathcal{A} to \mathcal{C} and \mathcal{W} to \mathcal{W}' can be done independently, specification-based design enables *a posteriori* or dynamic addition of self-healing. That is, given a concrete implementation \mathcal{C} , it is possible to add self-healing to \mathcal{C} by first designing an abstract fault-tolerance wrapper \mathcal{W} using solely an abstract specification \mathcal{A} of \mathcal{C} , and then adding a concrete refinement \mathcal{W}' of \mathcal{W} to \mathcal{C} .



Ongoing and future work: Our notion of self-healing preserving refinement depends on the semantics of the input. However, we conjecture that it is possible to identify certain syntactic constructs that imply self-healing of a program. Some examples of these constructs are self-cleaning data structures (e.g., sets) and soft-state variables that are assigned fresh values periodically. We plan to develop tools (*syntax-driven self-healing preserving compilers*) that exploit such syntactic constructs in order to check whether a given program is fault-tolerant or not, and to add self-healing to a rich class of programs.

References

- [1] G. Chockler, M. Demirbas, S. Gilbert, C. Newport, and T. Nolte. Consensus and collision detectors in wireless ad hoc networks. In *Principles Of Distributed Computing (PODC)*, pages 197–206, 2005.
- [2] M. Demirbas and M. Hussain. A mac layer protocol for priority-based reliable broadcast in wireless ad hoc networks. *BroadNets*, 2006.
- [3] M. Demirbas and S. Balachandran. Robcast: A singlehop reliable broadcast protocol for wireless sensor networks. *The Sixth International Workshop on Assurance in Distributed Systems and Networks (ADSN)*, 2007.
- [4] M. Demirbas, O. Soysal, and M. Hussain. Singlehop collaborative feedback primitives for wireless sensor networks. *INFOCOM miniconference*, 2008.
- [5] M. Demirbas and H. Ferhatosmanoglu. Peer-to-peer spatial queries in sensor networks. *The Third IEEE Int. Conf. on Peer-to-Peer Computing*, pages 32–39, September 2003.
- [6] M. Demirbas, A. Arora, V. Mittal, and V. Kulathumani. Design and analysis of a fast local clustering service for wireless sensor networks. *Broadband Wireless Networking Symposium, BroadNets*, pages 700–709, October 2004.
- [7] M. Demirbas and X. Lu. Distributed quad-trees for spatial querying in wireless sensor networks. *IEEE International Conference on Communication (ICC)*, 2007.
- [8] M. Demirbas, A. Arora, and V. Kulathumani. Glance: A lightweight querying service for wireless sensor networks. *OPODIS*, pages 244–259, 2006.
- [9] M. Demirbas, A. Arora, T. Nolte, and N. Lynch. A hierarchy-based fault-local stabilizing algorithm for tracking in sensor networks. *8th International Conference on Principles of Distributed Systems (OPODIS)*, pages 299–315, 2004.
- [10] V. Kulathumani, M. Demirbas, A. Arora, and M. Sridharan. Trail: A distance sensitive wireless sensor network service for distributed object tracking. *European Conference on Wireless Sensor Networks*, pages 83–100, 2007.
- [11] M. Demirbas, A. Arora, and M. Gouda. A pursuer-evader game for sensor networks. *Proceedings of the Sixth Symposium on Self-Stabilizing Systems(SSS'03)*, pages 1–16, June 2003.
- [12] M. Demirbas, O. Soysal, and A. S. Tosun. Data salmon: A greedy mobile basestation protocol for efficient data collection in wireless sensor networks. *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 267–280, 2007.
- [13] S.K. Yoon, O. Soysal, M. Demirbas, and C. Qiao. Coordinated locomotion of mobile sensor networks. Technical Report 12, University at Buffalo, SUNY, 2007. Under submission.
- [14] M. Demirbas. A transactional framework for programming wireless sensor/actor networks. *Proceedings of the 11th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, pages 123–129, 2007.
- [15] M. Demirbas, O. Soysal, and M. Hussain. Transact: A transactional programming framework for wireless sensor/actor networks. *Submitted to IPSN*, 2007.
- [16] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y-R. Choi, T. Herman, S. S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks (Elsevier)*, 46(5):605–634, 2004.

- [17] M. Demirbas, K. Y. Chow, and C. Wan. Insight: Internet-sensor integration for habitat monitoring. *Advanced EXPERIMENTAL activities ON WIRELESS networks and systems (EXPONWIRELESS) Workshop (as part of WOWMOM)*, pages 553–558, 2006.
- [18] M. Demirbas and A. Arora. Convergence refinement. *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, pages 589–597, July 2002. Best paper(1st/335).
- [19] A. Arora, M. Demirbas, and S. S. Kulkarni. Graybox stabilization. *Proceedings of the International Conference on Dependable Systems and Networks (ICDSN)*, pages 389–398, July 2001.