

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* 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 10 billion CPUs produced in year 2008 went to traditional PC platforms (desktops, laptops, clusters, etc.) whereas the remaining 9.8 billion found their 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 application 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 the 0.01% message loss rate 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. More recently, I have also started working on smartphone applications for citywide sensing and collaboration to complement and extend my work on WSNs. My current projects are as follows.

1. An in-network collaboration and coordination framework for wireless sensor/actor networks (funded by an NSF CAREER award 2008-2013)
2. Efficient and resilient querying and tracking services for wireless sensor networks (funded by Office of Naval Research 2009-2012)

3. Tool support for producing high-assurance and reliable software for wireless sensor/actor networks (funded by NSF Computer Systems Research Award 2009-2012)
4. Crowdsourced sensing (funded by Google Research Award 2010)

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

1 An in-network collaboration and coordination framework for wireless sensor/actor networks (funded by NSF-CAREER)

The goal of this project is to design and implement a framework that provides simple programming abstractions to cope with the consistent coordination challenges of wireless sensor-actor networks (WSANs) while retaining efficiency of execution. Our key insight in this project is to observe that singlehop wireless broadcast has many useful features for facilitating collaboration and coordination. Firstly, broadcasting is atomic (i.e., for all the recipients of a broadcast, the reception occurs simultaneously), which is useful for synchronizing the nodes in singlehop for building a structured operation. Secondly, broadcast messages share the same medium enabling collision detection and snooping, which are useful for implementing collaboration and coordination in a decentralized manner.

We next discuss the two components of our framework; a singlehop communication primitive for fast robust feedback collection and a transactional abstraction for robust computing in WSANs.

1.1 Singlehop robust communication in WSNs

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.

Contributions: In [1,2] 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.

We developed reliable implementations and quantitative evaluations of RCD on TinyOS and mote platforms in [3]. Using RCD, we built a primitive “pollcast” for quick and robust singlehop feedback collection from a singlehop neighborhood [3]. We have analyzed the theoretical lowerbounds and upperbounds associated with querying with pollcast in [4]. We are working on implementing applications of RCD for singlehop collaboration protocols in WSNs, including a quick and robust threshold querying primitive *tcast* [5].

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 [6] and RobCast [7].

1.2 Programming abstractions for robust computing in WSNs

Problem: Effectively managing concurrent execution is one of the biggest challenges for future WSNs: 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 WSNs 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: Our framework, TRANSACT [8,9], provides an efficient and lightweight implementation of transaction primitive in a distributed manner. In contrast to database systems, in distributed WSNs 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 WSNs, 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 WSN critically, TRANSACT implements an optimistic concurrency control solution.

We have implemented TRANSACT on TinyOS and mote platforms [9]. A major contribution of TRANSACT is to simplify the reasoning and verification of a distributed WSNs 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 atomic manner. This paradigm facilitates achieving consistency and coordination and may enable development of more efficient control and coordination programs than possible using traditional models. 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.

By restricting the state updates to be only local at the node, rather than allowing a direct write to the states of neighboring nodes, we obtain a useful subset of TRANSACT, which we dub as Read-All-Write-Self (RAWS) transactions [10]. RAWS transactions have negligible overhead¹, yet still achieve conflict-serializability, and hence a correct transformation from the shared-memory model to the wireless networks model.

2 Efficient and resilient querying and tracking services for WSNs

(funded by Office of Naval Research 2009-2012)

Problem: Querying, and tracking services require continuous maintenance of distributed data structures (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 [11], 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 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* [12] 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

¹By no overhead we mean that these transactions do not need validation timers or extra messages for aborting.

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* [13]. That is, each cluster in the hierarchical partitioning can tolerate expansion up to two-folds 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* [14] 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* [15], 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* [19], 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.

Continuing in this vein, in *Trail* [20], 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 tolerates holes in the network as long as the network is not partitioned.

We are currently working on applying our geometric ideas and techniques for devising distributed network algorithms in the context of mobile WSNs. Recently, in [21, 22], we investigated energy-efficient and low-latency data collection mechanisms 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 investigated efficient, local, and distributed strategies that ensure that the network always stays connected even though the neighbors may change for each node [23].

3 Tool support for producing high-assurance and reliable software for WSANs (funded by NSF CSR)

Problem: The ability to write programs in an abstract model and then translate them into a concrete model is indispensable for software development, as witnessed by the prevalence of compilers today. This transformation is especially valuable for distributed systems and decentralized programs (as in WSNs and WSANs). By writing abstract programs at a high atomicity model, the developer can ignore several low level issues such as communication and race conditions among different processes, and focus on the functionality to get it right. Also, since the abstract program omits these details, it is possible to thoroughly verify them by using techniques such as model checking and/or theorem proving, as well as synthesizing and adding desired level of fault-tolerance to them [24–26]. While the problem of transformation from abstract programs to concrete programs has been extensively considered in the distributed systems literature, all of these traditional transformation algorithms employ traditional network architectures at the concrete (e.g., point to point communication channels), hence, are not applicable for WSNs and WSANs.

Approach: The main challenge in addressing this problem is that the high-level models considered in developing and verifying distributed systems do not account for the challenges and opportunities in wireless networks. To address the impedance mismatch between the high-level models used in distributed systems literature and the wireless network model, our transformation employs a middleware that has the flexibility to choose between a push-based or a pull-based middleware depending on which one would be more efficient based on the communication characteristics of the abstract program, and produce C++ and TinyOS code to be executed at the concrete model, while preserving both the correctness and the reliability properties of the abstract system. Therefore, our transformation tool creates a bridge between high level distributed systems protocols (i.e., those for routing, leader election, group membership, virtual synchrony, consensus, mutual exclusion) and the real deployments. Moreover, our transformation also ensures fault-tolerance at the concrete system by guarding against the corruption of the auxiliary state introduced at the concrete while implementing abstract data structures and operations.

Contributions: We are designing and developing a tool (and the associated algorithms) that

allows programs for WSANs to be written in high-level models similar to those used traditionally to describe abstract distributed programs, and that automatically transforms these abstract programs, while preserving their correctness and reliability properties, into programs that can be easily deployed in WSANs. Since several problems (or variations thereof) considered in traditional distributed systems (e.g., routing, reliable broadcast, tracking, tree maintenance, consensus, mutual exclusion, group membership, leader election) need to be solved in the context of WSANs, the results from this project will enable the reuse of existing algorithms as well as speed up the development and deployment of WSANs.

In addition we are also designing and developing a tool (and associated algorithms) that manipulates the given abstract distributed programs for the automated addition of desired level of fault-tolerance. This tool enables the designer to concentrate only on the functional aspects of the protocol by automating the addition of fault-tolerance. We expect that this feature will be especially valuable for developers who are proficient about the functionality requirement of their program and eager to leave the fault-tolerance aspect to an automated approach (designed by experts in fault-tolerance). For automated addition of fault-tolerance, we leverage on our collaboration with Prof. Sandeep Kulkarni at Michigan State University. The results from our preliminary implementation of automated synthesis of fault-tolerance are very promising, and attest to the feasibility of our techniques. In [28], we have demonstrated the feasibility of synthesis of programs with state space of 2^{100} .

4 Crowdsourced sensing (funded by Google Research)

Problem: Smartphones have been hailed as the solution to the large-scale sensing problem as they provide significant advantages over traditional WSNs. Firstly, smartphones are mobile. Wherever a smartphone user goes, smartphone can take sensor readings (with built-in sensors for acoustic, image, video, accelerometer, tilt, magnetometer, and potentially with other integrated custom sensors) and the dynamic geolocation feature of smartphones enables these readings to be location and time-stamped. Secondly, smartphones are personal and administrated by their users. Therefore, energy-efficiency is not as big a concern for using smartphones for sensing, and also the human operator can intervene and help for tasks requiring human intelligence (such as taking a picture of a requested location, answering a question). Finally, smartphones have already solved the market penetration problem and are available ubiquitously with ever growing acceleration.

However, smartphones fall short of their potentials. The state-of-the-art today is for each device to connect to Internet to download/upload data, and accomplish a task that does not require collaboration and coordination. Providing an infrastructure for publish/subscribe and tasking of these devices would enable any device to task other devices in a region to acquire data, as well as search the data published by these devices.

Approach: We propose that Twitter can provide an ideal “open” publish-subscribe infrastructure for sensors and smartphones and enable deployment of crowdsourced sensing and collaboration applications. The open publish-subscribe system of Twitter implies that different actors may integrate published data differently, and third parties can use the gathered data in unanticipated ways to offer new services. The social networks angle of Twitter is also useful for crowdsourced sensing and collaboration applications. Finally, the wide popularity of Twitter (more than 30 million users

just in the US) make Twitter a good venue for crowdsourced sensing and collaboration: It is easier to give the community a tool than to give the tool a community.

Contributions: We designed and implemented a crowdsourced sensing system over Twitter, and deployed a crowdsourced weather radar using our system [31]. Central to our system is a Twitter-bot with an integrated database, *Askweet*, that accepts questions, crowdsources them, and aggregates the answers to reply back to the querier. The system also includes a smartphone client, *Sensweet*, for automatically pushing sensor readings to Twitter. Twitter acts as a middleware for publish/subscribe as well as search & discovery. Our results from this real-world Twitter experiment show promise for the feasibility of this approach. We find that although we do not offer the users any incentives to reply, our queries receive at least 15% reply ratios. Surprisingly, 50% of the total replies arrive within the first 10 minutes of our query, and 80% of the replies arrives within the first 2 hours, promising low-latency operations for crowdsourcing applications. We also find that consistently the majority of replies come from users that access Twitter from mobile phones.

In our current work, we aim to (1) solve the challenges in sensor/smartphone integration to Twitter (establishing sensor tweet standards, providing incentives, building filters, exploiting social networking features), (2) implement our crowdsourced sensing and collaboration system over Hadoop and deploy it on the cloud for elastic scalability, (3) deploy a participatory noise-mapping application on smartphones, and (4) design social collaboration applications. For our future work, we will use our work on mobile user profiling [32–35] to improve the efficiency of our crowdsourced sensing system. Another direction we are pursuing is to develop techniques on opinion and information mining over Twitter [36,37] to be used for citywide sensing applications.

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] G. Chockler, M. Demirbas, S. Gilbert, N. A. Lynch, C. C. Newport, and T. Nolte. Consensus and collision detectors in radio networks. *Distributed Computing*, 21(1):55–84, 2008.
- [3] M. Demirbas, O. Soysal, and M. Hussain. Singlehop collaborative feedback primitives for wireless sensor networks. *INFOCOM*, pages 2047–2055, 2008.
- [4] J. Aspnes, E. Blais, M. Demirbas, R. O’Donnell, A. Rudra, and S. Uurtamo. k+ decision trees. *International Workshop on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities (Algosensors)*, 2010.
- [5] M. Demirbas, H. Gunes, and S. Tasci. A singlehop collaborative feedback primitive: Threshold queries in wireless sensor networks. Technical Report no:2010-07, SUNY Buffalo, 2010.
- [6] M. Demirbas and M. Hussain. A mac layer protocol for priority-based reliable broadcast in wireless ad hoc networks. *BroadNets*, 2006.
- [7] 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.
- [8] 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.

-
- [9] M. Demirbas, O. Soysal, and M. Hussain. Transact: A transactional programming framework for wireless sensor/actor networks. *IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN)*, pages 295–306, 2008.
- [10] O. Soysal and M. Demirbas. Transactional concurrency control for multihop wireless sensor networks. Technical Report no:2009-05, SUNY Buffalo, 2009.
- [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 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.
- [13] M. Demirbas, A. Arora, V. Mittal, and V. Kulathumani. A fault-local self-stabilizing clustering service for wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems, Special Issue -Localized Communication*, 17(9):912–923, September 2006.
- [14] M. Demirbas, X. Lu, and P. Singla. An in-network querying framework for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 2009. To appear.
- [15] M. Demirbas, A. Arora, and V. Kulathumani. A lightweight querying service for wireless sensor networks. *Theoretical Computer Science Journal, Elsevier*, 2009. To appear.
- [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] X. Lu and M. Demirbas. Writing on water, a lightweight soft-state tracking framework for dense mobile ad hoc networks. *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 359–364, 2008.
- [18] X. Lu, M. Demirbas, and C. Qiao. A holistic solution to the pursuer evader tracking problem. *Proceedings of the 28th IEEE International Symposium on Reliable Distributed Systems (SRDS)*, pages 179–188, 2009.
- [19] 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.
- [20] V. Kulathumani, M. Demirbas, A. Arora, and M. Sridharan. Trail: A distance sensitive network service for distributed object tracking. *ACM Transactions on Sensor Networks*, 2009. To appear.
- [21] 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.
- [22] O. Soysal and M. Demirbas. Data spider: A resilient mobile basestation protocol for efficient data collection in wireless sensor networks. *The 6th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2010.
- [23] S.K. Yoon, O. Soysal, M. Demirbas, and C. Qiao. Coordinated locomotion of mobile sensor networks. *Fifth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2008.
- [24] 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).
- [25] 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.

-
- [26] M. Demirbas and A. Arora. An application of specification-based design of self-stabilization to tracking in wireless sensor networks. In *SSS*, pages 203–217, 2008.
 - [27] M. Arumugam, M. Demirbas, and S. Kulkarni. Slow is fast for wireless sensor networks in the presence of message losses. In *12th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, 2010.
 - [28] B. Bonakdarpour and S. S. Kulkarni. Exploiting symbolic techniques in automated synthesis of distributed programs. *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2007.
 - [29] 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.
 - [30] R. Sudhaakar, A. Sanzgiri, M. Demirbas, and C. Qiao. A plant-and-play wireless sensor network system for gate monitoring. *The 10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, 2009.
 - [31] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. Yilmaz, and H. Ferhatosmanoglu. Crowd-sourced sensing and collaboration using twitter. *11th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2010.
 - [32] M. A. Bayir, M. Demirbas, and N. Eagle. Discovering spatiotemporal mobility profiles of cellphone users. *To appear in Elsevier Pervasive and Mobile Computing Journal, Special Issue on Human Behaviour in Ubiquitous Environments*, 2009.
 - [33] M. A. Bayir, M. Demirbas, and A. Cosar. A web based personalized mobility service for smartphone applications. *To appear in The Computer Journal*, 2009.
 - [34] M. A. Bayir and M. Demirbas. Pro: A profile based routing algorithm for pocket switched networks. *IEEE Global Communications Conference (Globecom)*, 2010.
 - [35] M. Demirbas, C. Rudra, A. Rudra, and M. A. Bayir. imap: Indirect measurement of air pollution with cellphones. *PerCOM*, pages 537–542, 2009.
 - [36] C. G. Akcora, M. A. Bayir, M. Demirbas, and H. Ferhatosmanoglu. Identifying breakpoints in public opinion. *Workshop on Social Media Analytics*, 2010.
 - [37] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. *33rd ACM SIGIR Conference*, 2010.