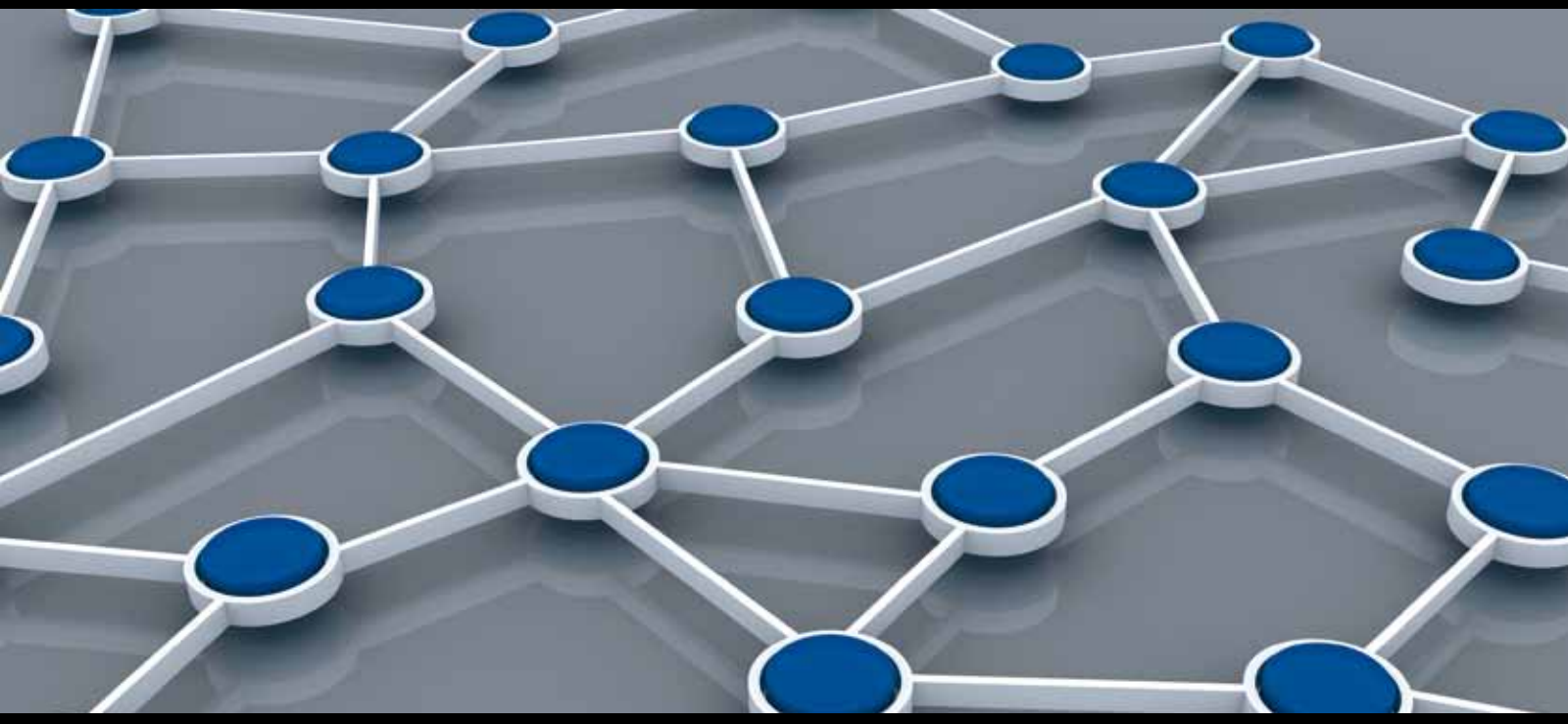
International Journal of Distributed Sensor Networks

Substitution and Wireless Sensor and Actuator/Robot Networks

Guest Editors: Jiming Chen, Hannes Frey, Pedro Ruiz, and David Simplot-Ryl



Substitution and Wireless Sensor and Actuator/Robot Networks

International Journal of Distributed Sensor Networks

Substitution and Wireless Sensor and Actuator/Robot Networks

Guest Editors: Jiming Chen, Hannes Frey, Pedro Ruiz, and David Simplot-Ryl

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 871268, 2 pages
doi:10.1155/2012/871268

Editorial

Substitution and Wireless Sensor and Actuator/Robot Networks

Jiming Chen,[1] Hannes Frey,[2] Pedro Ruiz,[3] and David Simplot-Ryl[4]

[1] *Department of Control Science and Engineering, Zhejiang University, Hangzhou, China*
[2] *University of Paderborn, Paderborn, Germany*
[3] *University of Murcia, Murcia, Spain*
[4] *INRIA Lille-Nord Europe, Villeneuve d'Ascq, France*

Correspondence should be addressed to David Simplot-Ryl, david.simplot@lifl.fr

Received 14 October 2012; Accepted 14 October 2012

Wireless sensor and robot/actuator networks (SANETs) are an integration of wireless sensor networks and multirobot systems, bringing many new applications including the critical real-time monitoring of critical infrastructures that might be subjected to natural and/or human-induced hazards. In SANET, sensors often have small size and low capacity, while robots are resource rich that allow controlled mobility and actions on environment or networks with failures. Such heterogeneity, unreliability, and many other associated features together introduce open technical, social, and economic problems that are calling for more research attentions.

The aim of this special issue is to present to both academy and industry communities the most recent researches and developments investigating these open problems. Out of 16 submissions, 8 exceptional contributions were finally selected after several rounds of review by the invited reviewers and the guest editors.

The paper by Ch. Paschalidis and R. Wu addresses the problem of energy optimized (network lifetime maximized) routing with and without energy allocation. The authors formulate a worst case problem and propose robust solutions. Unlike existing routing methods, which are shown to be optimistic, the proposed robust solutions are practical and able to balance between performance and robustness. Energy efficiency is critical for the performances of resource-constrained SANET. In the paper "*remaining energy-level-based transmission power control for energy-harvesting WSNs*," Dai et al. propose a transmission power control scheme to coordinate the communication ranges of sensor nodes, by which to balance the energy consumption and harvesting speed in energy harvesting sensor networks.

Mobility management is a substantial new problem of wireless sensor and robot/actuator networks, and how to enhance the network performances taking advantage of the mobile nodes is still a challenging issue. The paper by I. Amundson et al. proposes a waypoint navigation system, integrated with a lightweight localization algorithm, based on resource-constrained mobile sensor networks. By simulations, the authors show that the navigation performance is robust against location errors, and that navigation can run without the use of digital compass. The paper by C. Q. Nguyen et al. demonstrates the ability of using mobile robots to establish wireless mesh networks and increase the network throughput by redistributing the nodes of the existing network. In the paper "*Distributed control of mobile sensor networks under RF connectivity constraints*," Y. Stergiopoulos et al. proposes a novel motion control algorithm for mobile wireless sensor networks under connectivity constraints. The algorithm is shown to be able to optimize area coverage while preserving connectivity. The paper by K. Miranda et al. proposes a localized and adaptive approach to adjust the deployment of mobile relays, which is shown to outperform static sensor networks in terms of delay, jitter, loss, and throughput.

The rest two papers are concerned with SANET hardware and applications. The paper by J. Zhang et al. identifies the challenges of the most existing wheeled robots in unfriendly environment, for example, with obstacles. Therefore, the authors design a jumping robot that can jump up to or over obstacles to recover connections in wireless sensor networks. The paper by J. V. Marti et al. proposes a fingerprinting localization method for mobile sensor and actuator networks deployed in smoke-filled indoor areas.

The eight papers included in this special issue reflect the research diversity and richness in substitution and wireless sensor and actuator/robot networks. We hope that this special issue will provide readers with the most recent insights to, and also stimulate researchers' interests in, this field. Finally, we would like to extend our gratitude to all authors for considering this special issue as a publishing avenue of their research results, and for their valuable contributions.

Acknowledgment

We thank all the reviewers for their helpful feedback and suggestions to improve the quality of each paper.

Jiming Chen
Hannes Frey
Pedro Ruiz
David Simplot-Ryl

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 128904, 9 pages
doi:10.1155/2012/128904

Research Article

Adaptive Deployment Scheme for Mobile Relays in Substitution Networks

Karen Miranda, Enrico Natalizio, and Tahiry Razafindralambo

Inria Lille-Nord Europe, FUN Research Team, 40 avenue Halley, 59650 Villeneuve d'Ascq, France

Correspondence should be addressed to Karen Miranda, karen.miranda@inria.fr

Received 13 December 2011; Accepted 22 May 2012

Academic Editor: Hannes Frey

We present how the mobility of routers impacts the performance of a wireless substitution network. To that end, we simulate a scenario where a wireless router moves between three static nodes, a source and two destinations of UDP traffic. Specifically, our goal is to *deploy* or *redeploy* the mobile relays so that application-level requirements, such as data delivery or latency, are met. Our proposal for a mobile relay achieves these goals by using an adaptive approach to self-adjust their position based on local information. We obtain results on the performance of end-to-end delay, jitter, loss percentage, and throughput under such mobility pattern for the mobile relay. We show how the proposed solution is able to adapt to topology changes and to the evolution of the network characteristics through the usage of limited neighborhood knowledge.

1. Introduction

Problem Description. It is critical to design efficient algorithms to support ubiquitous services in networked environments. This is due to the fact that wireless technologies are evolving into the next generation, so an increasing number of users will enjoy ubiquitous access. Some of the main challenges include a fairly complex node placement/deployment problem without prior knowledge of the optimal network topology or optimal mobile routers locations.

Previous work has focused on deployment and placement of mobile devices (e.g., robots) for area coverage [1–5]. In our work, the goal is to deploy a set of wireless mobile devices between classical network routers to restore the connectivity without prior knowledge of the optimal placement of the devices. Additionally, most efforts to date use local area networks, ad hoc networks, wireless sensor networks, and mesh networks for several different purposes, such as community and neighborhood networking, transportation systems, networking for developing countries, connection of isolated locations, spontaneous networking, and disaster recovery [6]. In these papers, the spontaneous networking approach is used for specific cases, such as hurricanes, earthquakes, fiber optic cable cuts, flash crowds, or in presence of weak connectivity. Instead, in this paper, we consider the use of controlled mobility provided to wireless routers to restore or improve network connectivity. We assume that mobile relays have *self-organization*, *self-optimization*, and *self-healing* capabilities in order to allow a flexible, scalable, and resilient deployment. Hence, the main issue in this context is how to *deploy* or *redeploy* the wireless mobile relays in order to keep the network services running.

Contributions. (1) We introduce different adaptive strategies for the deployment/redeployment of wireless mobile relays. (2) Our solution is localized, scalable, and adaptive. (3) We show that our scheme outperforms the static approach.

Paper Organization. The remainder of this paper is structured as follows. In Section 2, we state the problem, introduce some motivating applications, and state our assumptions, followed in Section 3 by the description of our solution, the simulation model, and the performance metrics. In Section 4, we evaluate the proposed scheme through extensive simulations and discuss the experimental results. Finally, Section 5 concludes this work and presents future developments.

2. Some Background

Spontaneous networking, or public service networking, is used in specific cases such as hurricanes, earthquakes, fiber optic cable cuts, flash crowds, or in presence of weak connectivity, the network not only must be deployed in a short period of time, but also must have capabilities as self-organization, self-optimization, and self-healing [7]. This type of networks is called rapidly deployable network (RDN).

We consider a substitution network (SN) as a kind of RDN. An SN is a temporary wireless network created to help the base network to keep providing services. This substitution network must be rapidly deployed to quickly adapt to network topology changes and to ensure the network connectivity [8].

A specific example of an application of an SN is the contractor's mistake in the Sydney's Business District [9]. In 2009, some contractors cut through 10,000 of Telstra company copper wires and 8 fiber optic cables by mistake. This caused over 12,000 business and residential customers without phone, mobile, or Internet services for several days. The cost to Telstra of this mistake was AU$1 million just to repair the wires, plus the compensation cost for the affected customers and a demand by the Australian government. Finally, it took Telstra about a week to replace the cables and restore the service.

Another example is after a natural or man-made disaster, such as earthquakes or terrorist attacks, when the communications networks are not destroyed but congested. In September 11, 2001, the radio network used by the Emergency Medical Service was saturated by panicked operator transmitting unnecessary information [10].

2.1. Related Work. In the last years, many schemes and solutions have been proposed to improve network performance by placing wireless relays in specific positions [11, 12]. The most common objectives are energy consumption and coverage as presented in [4, 13–17]. However, these solutions are not suitable to substitution networks because they depend on a preplaned deployment.

Evans et al. introduced in 1999 the concept of a rapidly deployable network [18]. The main idea is to deploy a network infrastructure *in promptu* to provide communication services for military applications. After this work, several deployment schemes have been proposed in the literature not only for military communications but also for emergency communications.

In order to address the deployment problem, a relay-based approach is presented in [19–21]. In most of these proposals, the first responders, for example, firemen or policemen, carry a personal mobile radio and small relays. Then, the first responders must drop these devices while exploring the emergency zone in order to maintain the connectivity with the central command thus creating a multihop network. Each mobile radio exchanges control information with the closest relay to decide when to drop a new small relay. So, the main focus is to propose a deployment decision process that maximizes the network performance.

Bao and Lee present a method to rapid deploy an ad hoc backbone for spontaneous networks with no preplanning [19]. The authors present a collaborative deployment algorithm, which takes into account physical or link quality measurements such as signal-noise ratio and packet loss rate. The algorithm measures the link quality through control messages added to the control packet header of the ad hoc routing protocol. They assume that each device can notice the different type of its neighbors, that is, if the neighbor is a mobile device or a relay, and also they keep track of each relay deployed.

Later, Souryal et al. present an algorithm for NIST real-time deployment of mesh networks project [21]. The authors propose an algorithm based on a quick evaluation of the physical layer performed by the mobile radio. In a nutshell, the mobile radio establishes one-hop communication by constantly broadcasting probe packets to previous relays, when some relays in the range respond with a probe ACK packet, the mobile radio measures the RSS through ACK reception, if the RSS value falls below a given threshold level, then a new relay must be dropped.

Nevertheless, the concept of static relays has changed, for example, the LANdroids project launched by the Defense Advanced Research Projects Agency, DARPA. The goal is to propose an RDN for battlefield based on mesh networks composed of small mobile relays. Based on this call, a spreadable connected autonomic network (SCAN) is presented in [22]. SCAN is a mobile network that automatically maintains its own connectivity by moving constantly its nodes. The authors present the SCAN algorithm capable to deal with environments where the predeployment mapping is expensive or infeasible without any previous information of the environment. This protocol proposes an online distributed process where each node uses two-hop radius knowledge of the network topology and each of them determines when to stop its motion if the decision criterion indicates risk of dividing or disconnecting the network.

2.2. Key Points. We propose to deploy a network composed of a fleet of dirigible wireless mobile routers for public service. In order to fully adapt to the current conditions, the mobile routers should move or redeploy on demand. This means that, not only the edges of the net may move but also the core or part of the core. One of the deployment issues is in which direction move the router to avoid disconnection or degradation of the quality of service (QoS).

The deployment of a network composed of a fleet of dirigible wireless mobile routers (named from now on as *substitution network*) can be useful in case of multiple link failures as in natural disasters, weak connectivity, fiber optic cable cuts, or flash crowds.

In this work, we focus on a typical use case of substitution networks as presented in [8]. In this scenario, the substitution network aims at helping a base network to restore and maintain some of the basic services available before the failure. Thus, a fleet of mobile relays is self-deployed to compose a substitution network together with the base network. Thereby, we evaluate an adaptive positioning

scheme to increase the network depending on the driving applications.

Our basic idea is that, during the network lifetime, each wireless mobile device of the substitution network determines a new position by using the feedback on the link quality coming from its neighbors.

We assume that two nodes are "neighbors" when they are within the communication range of each other. Likewise, we assume that some of the devices are fixed, that traffic needs to be transferred between two fixed devices, and that wireless devices dynamically move in the scenario and act as relays, regardless the routing protocol. Besides, we assume that each device is aware of its own position by using GPS or any other localization system, so as to allow nodes to use controlled mobility. More ever, as with many link layer protocols, we assume that each node is equipped with a timer and an 802.11 wireless card, and it has an identifier that is unique in the network (MAC address).

In this paper, we use the term "broadcast" for message propagation in a device's neighborhood. As well as, we call "link parameter" a measure of link quality between a mobile device and each of its neighbors, for example, signal-noise ratio (SNR), received signal strength (RSS), round-trip-time (RTT), and transmission rate (TR).

Based on the assumptions above, we propose a solution to deploy/redeploy intermediate mobile relays, that is,

(1) localized: every decision taken by the mobile relay is based only on close neighbors (i.e., one-hop neighbors) and local link information. The mobile routers take advantage of probe packets to exchange information about their surrounding links status, and drive their positioning;

(2) scalable: as a consequence of the previous property, our solution is scalable on the network size and the mobility strategy of the surrounding wireless mobile routers;

(3) adaptive: the algorithm ensures that the connectivity quality is permanently monitored based on close neighbors and local link information. As a consequence, the proposed placement scheme is adaptive to topology changes and to the evolution of the network characteristics.

3. Proposed Scheme

Briefly, the major steps of the algorithm that runs in each node independently are the following: (1) measurement of the "link parameter" (2) computation of the new position and (3) movement towards the computed position. Each of these steps is described.

No prior knowledge of the optimal mobile device locations is assumed to be available at nodes. Our algorithm uses close neighbors and local links information to allow nodes to position themselves. Each relay runs the algorithm regularly and measures the link parameters.

3.1. Detailed Operation

3.1.1. Measure Link Parameters. In order to measure link parameters, we use an intrusive method. The wireless mobile device regularly (every t seconds) broadcasts *probe request* packets containing a sequence number and the *id* or MAC address of the wireless mobile device. Each node receiving *probe request* replies with a *probe reply message* by using unicast transmission and including information such as its id, its position, and any local information regarding the link parameters. We use an intrusive method to get up-to-date information regarding link parameters but also to get a consistent and fair view of each link in the surroundings of a mobile device. An additional advantage of using broadcasting of *probe request* packets is that we can avoid the clock synchronization problem between devices.

It is important to notice that the probe packets, request and reply, have a higher access priority than other packets. Specifically, when a probe packet is generated, it will be put at the head of line in the link layer queue. However, these packets cannot preempt a scheduled transmission at the MAC layer. Note also that since the mobile routers are only used as relays, they are able only to exchange protocol stack information up to network layer. Thus, they cannot use application or transport layer measures directly or indirectly related with the "link parameter" currently measured.

3.1.2. Compute New Position. Each node computes its new position based on the surrounding link parameters every $k \times t$ seconds, where k is the number of probe packets, to ensure that enough measures are used to get consistent statistics on the link parameter. The wireless mobile device stores the received value and the measurements obtained through the probe reply. A sliding window is used to compute the statistics, and a FIFO policy is used to remove older values of the link parameters.

The wireless mobile device compares the values of the link parameter received from the next and the previous hop, X_{next} and X_{prev}. For example, when the considered parameter is the round-trip time ($X = \text{RTT}$), if $X_{\text{next}} > X_{\text{prev}}$, then the wireless device will move toward the next node. The degree of the inequality changes according to the link parameter considered. In this case, we assume that RTT is somehow related to the distance between nodes. In case of multiple flows passing through the same device, the wireless mobile device will move towards the node i with the maximum RTT. The link parameter measurements are averaged and used to compute the new position. The mobile device can use measurements from different layers. We consider RTT as a network layer metric, TR as the rate at which a packet is sent, as a link layer metric, and the SNR and the RSS as physical layer metrics.

3.1.3. Move to New Position. In this step, each wireless device moves forward on the computed direction for a distance d. This stepwise choice is arbitrary and it would be easier to relate the traveled distance d to the link parameter value. However, we chose this stepwise movement to be

```
(i) Message formats:
   ProbeRequest messages:    Identifier src;
   ProbeReply messages:      Identifiersrc, dst;
(ii) Parameter:
   double ProbePeriod, SendTime, RTT;
   int    k, Move;
PartI—Link parameters n:
 (1) set TIMER to expire in time ProbePeriod;
 (2) while (1) do
 (3)   if ( TIMER ≤ 0 ) then
 (4)     Send ProbeRequest Message;
 (5)     SendTime = NOW;
 (6)     set TIMER to expire in time ProbePeriod;
 (7)   end if
 (8) end while
 (9) while (1) do
 (10)   Upon reception of a ProbeReply
 (11)     RTT = NOW − SendTime;
 (12)     Store RTT in a table with the ProbeReply sender;
 (13) end while
Part II—Compute new position and move:
 (1) set TIMER to expire in time k × ProbePeriod;
 (2) while (1) do
 (3)   if ( TIMER ≤ 0 ) then
 (4)     Compute link parameter for Next and Prev hops;
 (5)     if (RTT_next > RTT_prev) then
 (6)       Move towards the Next hop;
 (7)     else if (RTT_next < RTT_prev) then
 (8)       Move towards the Prev hop;
 (9)     else
 (10)        Do not move;
 (11)    end if
 (12)     set TIMER to expire in time k × ProbePeriod;
 (13)  end if
 (14) end while
```

Algorithm 1: APA (adaptive positioning algorithm).

more realistic since in real environments, some geographical positions cannot be considered as a suitable position due to potential obstacles, for example, a wireless mobile device cannot cross a vehicles road.

Based on the link parameter measurements, the mobile device tries to equalize the metrics for both the previous node and the next node. The study of this tradeoff is left as a future work. It is important to notice here that we assume a correlation between link parameters and position due to wireless channel impairments or fading effects, for example.

The protocol version of the proposed scheme, named APA for (adaptive positioning algorithm), is given in Algorithm 1.

3.2. Topology and Simulation Description. We implement APA by using the NS 2.29 [23] network simulator with patches that reflect real wireless propagation, real wireless physical layer, and the adaptive autorate fallback (AARF) mechanism for 802.11b [24]. AARF adapts the transmission rates depending on the network conditions, in order to increase link reliability. Rather than using a fixed threshold, AARF adapts such threshold following binary exponential backoff. We also extend the simulator by adding a realistic channel propagation and error model, as proposed in [25], by adding the effect of interference and different thermal noises to compute the signal to noise plus interference ratio (SINR) and accounting for different bit error rate (BER) to SINR curves for the various codings employed. We use the DSR protocol for our simulations in order to account with an initial routing solution. As we mentioned before, APA is not tied to any routing protocol in particular, so it is designed to work with any routing protocol. Table 1 shows all the parameters used in our simulations.

Below, we present an experimental performance evaluation of APA under different network metrics such as *throughput*, *delay*, and *jitter*. They are defined as follows:

(i) average throughput (TH). The average throughput of a data transfer is: F/T *bits/sec*, where F is the number of bits transferred every second to the final destination;

Table 1: Simulation parameters.

Physical	Propagation	Two ray ground
	Error model	Real
	Antennas gain	$G_t = G_r = 1$
	Antennas height	$h_t = h_r = 1$ m
	Min received power	$P_{r-\text{thresh}} = 6.3$ nW
	Mobile router energy	50 J
	Communication range	240 m
MAC	802.11b	Standard compliant
	Basic rate	2 Mbps
	Auto rate fallback	1, 2, 5.5, 11 Mbps
LL	Queue size	50 pkts
	Policy	Drop tail
Routing	Static	Dijkstra
	Routing traffic	None
Transport and application	Flow	CBR/UDP
	Packet size	1052B
Statistics	Number of simples	$k = 10$
	Broadcast period	$t = U(0.1)$
Mobility	Movement step	$d = 2$ m

Figure 1: Simple evaluation scenario.

(ii) average end-to-end delay (D). This is the total average time for a packet to travel from source to destination;

(iii) average jitter (J). We compute the jitter as the measure of the variability over time of the packet latency across a network; as known, jitter is a function of the delay;

(iv) loss percentage (L). The loss percentage is equal to $((l-T)/l)\times 100$, where l is the total number of packets arriving at the receiver during the simulation time, and T is the total number of packets.

At the transport layer, we transmit UDP traffic with a packet size of 1000 B. We also vary the average transmission rate with steps of 10, 50, 200, 300, 600, 1000 kbps for each set of simulations. Each simulation runs for a period of 2000 seconds.

4. Results

We start by simulating a simple scenario with a source and a destination node that communicate through one wireless mobile relay (Figure 1). In this topology, the destination node is placed 250 meters far from the source node. At the beginning of the simulation, the relay node is placed 10 meters far from the source node. Thus, the relay starts moving by using our APA algorithm.

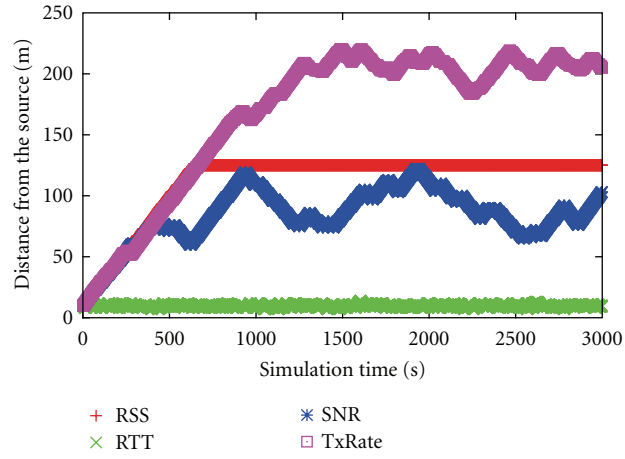


Figure 2: Placement evolution with APA comparison (RSS, RTT, SNR, and TxRate).

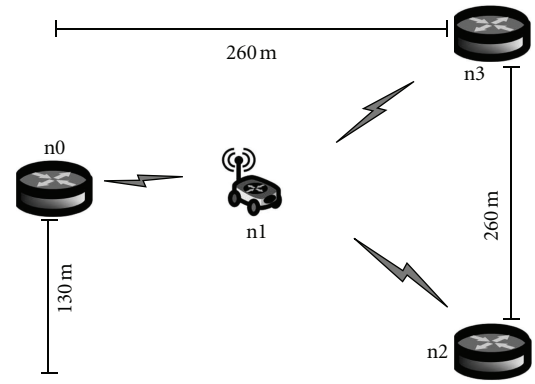
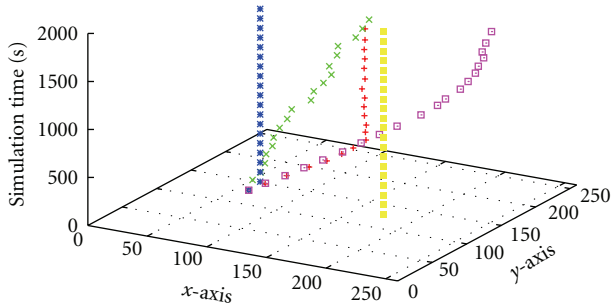


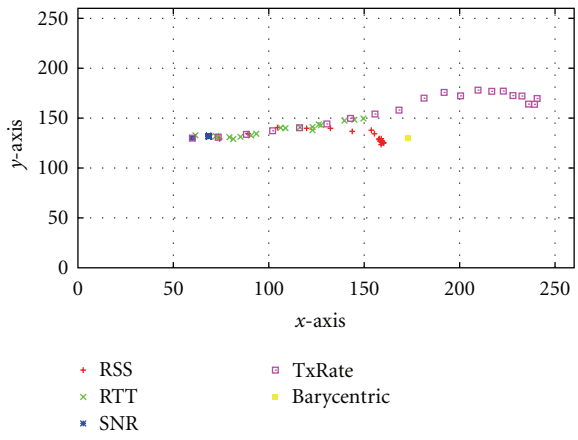
Figure 3: Evaluation scenario with one source and two different destinations.

Thus, we evaluate the convergence of our proposal with each link parameter, RSS, RTT, SNR, and TxRate. We use the topology illustrated in Figure 1, with UDP traffic with a packet size of 512 B during 3,000 seconds. The resulting movements are depicted in Figure 2, the relay moves between the source and the destination trying to position itself by equalizing each of the mentioned parameters. We observe that, by using RSS as input for our scheme, the relay reaches exactly the middle position (i.e., 125 meters from the source) after less than one third of the simulation, and it remains in that position for the rest of the simulation time. Besides, when the relay uses the RTT, SNR, and TxRate as input, it keeps moving without reaching a fixed position.

Accordingly, we evaluate the network performance changing the number of flows and destinations. We use the topology depicted in Figure 3, where we present a source



(a) Cartesian plane in time 3D



(b) Cartesian plane perspective

Figure 4: Deployment evolution in time (RSS, RTT, SNR, TxRate, and Barycentric), contrasting the relay placement as function of time.

(n0) and two destinations (n2, n3) out of range. So, we use a relay (n1) to connect the source and the destinations. At the beginning of the simulation, the relay (n1) is placed 60 meters far from the source node (n0) on the straight line that connects the source node from the middle position between the receiver nodes (n2 and n3). For all the simulations, we consider transmitting UDP packets with a size of 1000 B, and we vary the transmission rate as we described in the previous section.

We compare the performance of each link parameter versus the performance of a fixed node. The fixed node is positioned on the barycenter of the given topology, that is, 173 meters far from the source node on the straight line that connects the source node from the middle position between the receiver nodes.

We present in Figure 4, the positioning evolution of the relay by using APA. This figure presents two views of the evolution, Figure 4(a) is a 3D view showing the movement on the Cartesian plane with the time on the z-axis. We observe that, when the relay uses SNR as the equalizing parameter, it stops moving after 200 seconds, by using RSS, it stops moving after 1500 seconds, whereas by using RTT and TxRate, it continues moving until the end of the simulation. The movement trace is depicted in Figure 4(b). Here, we observe that by using TxRate, the relay goes close to n3. We also observe that only the RTT parameter reaches the point that is closer to the barycenter, and the RTT-based scheme improves its performance in this scenario compared with the simple scenario presented before.

In the following simulation campaigns, we transmit two UDP flows starting at the same time. The source node (n0) transmits Flow 1 to destination node 1 (n2) and Flow 2 to the destination node 2 (n3). In Figure 5, we show the average end-to-end delay and the average jitter comparison for each flow. The performance for these two parameters, when the mobile relay is positioned on the barycenter, is constant. We see also that for a transmission rate under 600 kbps, the mobile relay obtains low values for both delay and jitter.

Figure 6 shows the results for throughput and the packet loss. We can see that when the relay is positioned on the barycenter, as the transmission rate increases, the results are constant and outperform those obtained by using the mobile relay. Besides, as the transmission rate increases, the packet loss grows. For both metrics, the RTT parameter performs better than the RTT, SNR, and TxRate parameters.

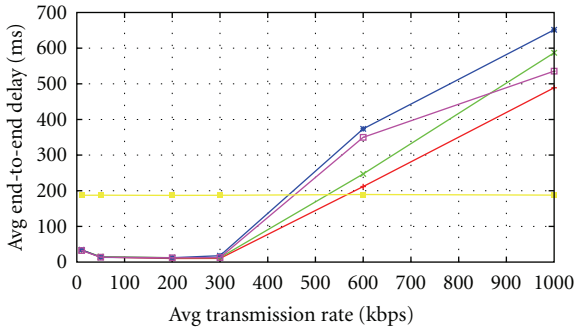
However, these results do not reflect the performance during the simulation time. This is important because, as we can see in Figure 7, the mobile relay in some moments improves the throughput values obtained with the static relay. We have to recall that the mobile relay starts moving from a position 60 meters far from the source node, which is worse than the barycenter in terms of network performance; but, when using our algorithm, the mobile relay improves its position and its performance on the fly.

5. Conclusion

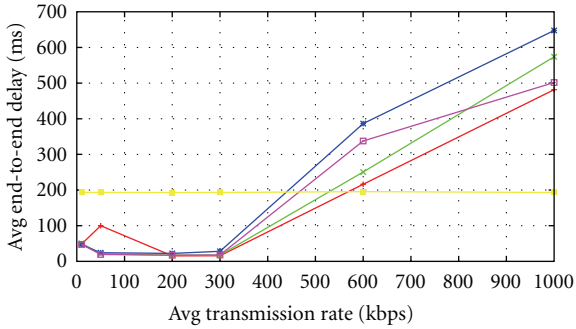
In this paper, we have presented a scheme based on different link parameters for substitution networks that operates in environments where connectivity guarantee is an issue.

We have introduced a suite of algorithm strategies to control the placement of wireless mobile devices. In particular, we have focused on networks where the source and the destination nodes of UDP traffic are connected through multihop communications performed by wireless mobile devices that act as relays. Specifically, our goal has been to *deploy* or *redeploy* the wireless mobile devices so that application-level requirements, such as data delivery or latency, are met. The APA algorithm we have proposed achieves these goals by using a localized and adaptive approach that determines the optimal positions of mobile relays in terms of delay, jitter, loss percentage, and throughput. Our simulation results show the importance of the placement of wireless mobile relay nodes to increase the performance at the application level. Finally, we compared our solution with the optimal theoretical placement, which is the barycenter.

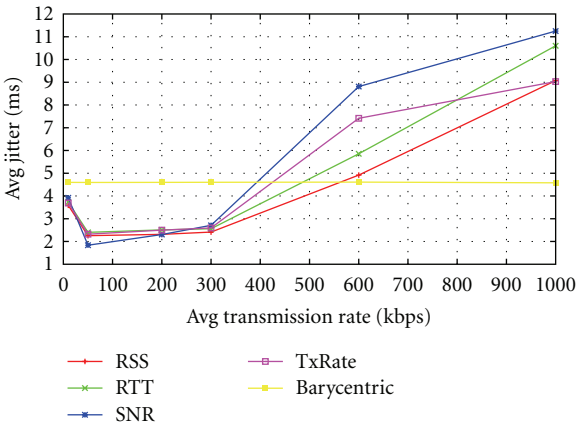
Our future work will focus on determining theoretically the optimal placement of the relay nodes in order to increase quality of service and quality of experience.



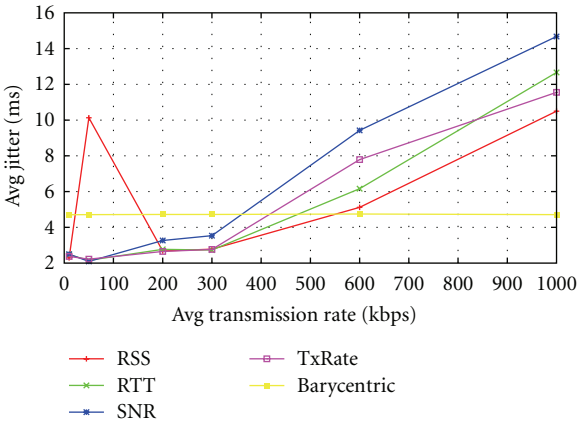
(a) End-to-end delay Flow 1



(b) End-to-end delay Flow 2

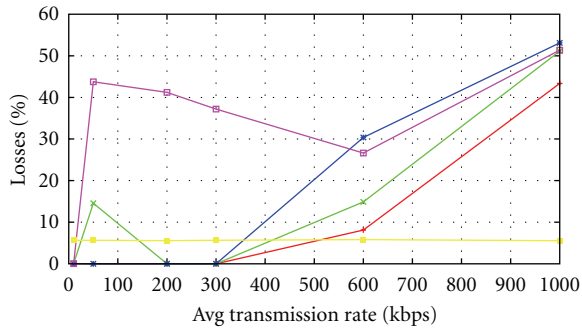


(c) Jitter Flow 1

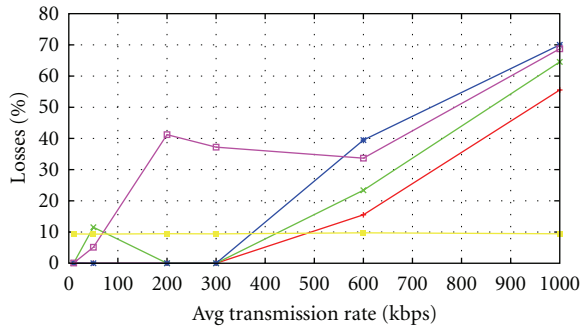


(d) Jitter Flow 2

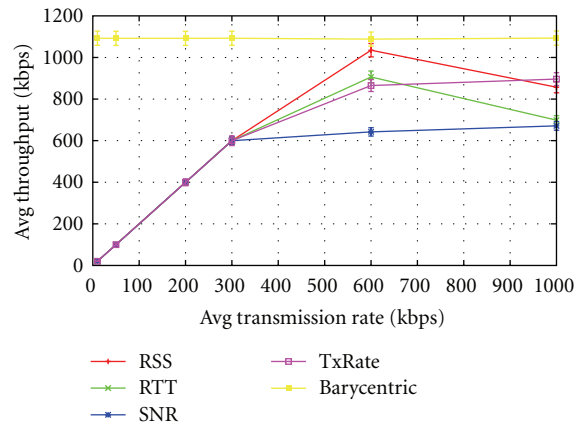
Figure 5: End-to-end delay and jitter comparison.



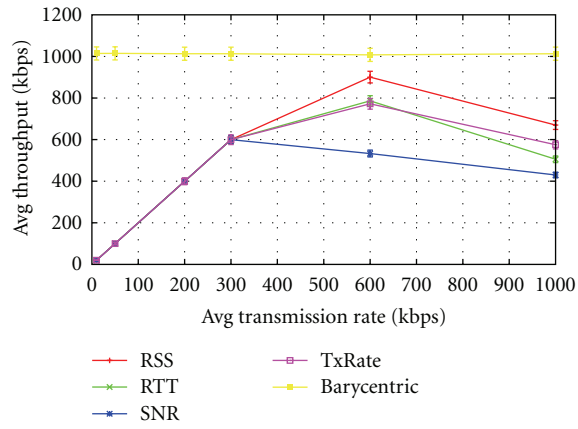
(a) Loss percentage Flow 1



(b) Loss percentage Flow 2

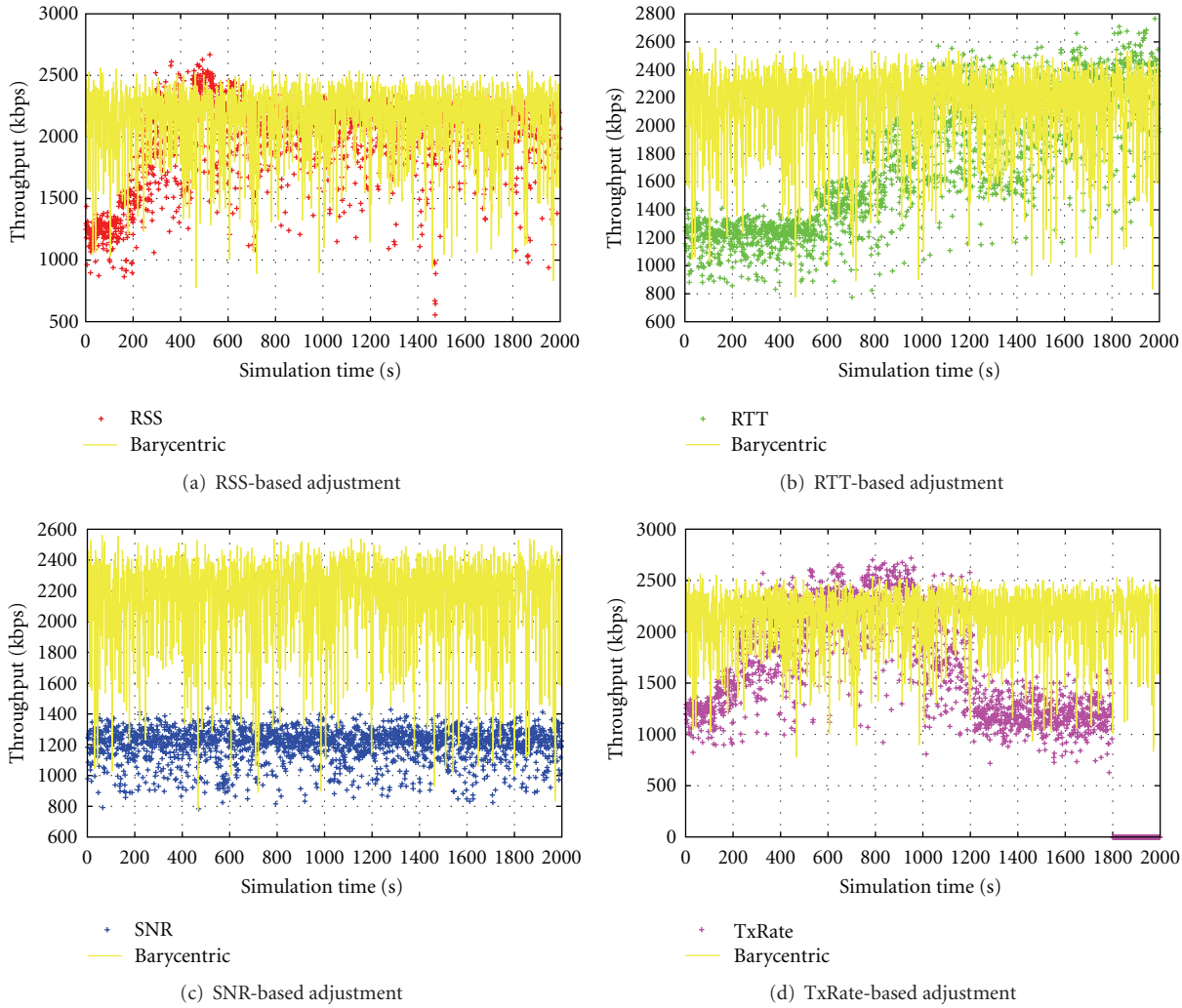


(c) Throughput Flow 1



(d) Throughput Flow 2

Figure 6: Throughput and packet loss percentage comparison.



(a) RSS-based adjustment

(b) RTT-based adjustment

(c) SNR-based adjustment

(d) TxRate-based adjustment

Figure 7: Instant throughput comparison between each link parameter and barycentric.

Acknowledgments

This work is partially supported by the French National Research Agency (ANR) under the VERSO RESCUE project (ANR-10-VERS-003) and by the Inria ARC MISSION grant.

References

[1] M. A. Batalin and G. S. Sukhatme, "Coverage, exploration and deployment by a mobile robot and communication network," *Telecommunication Systems*, vol. 26, no. 2–4, pp. 181–196, 2004.

[2] I. Chatzigiannakis, A. Kinalis, and S. Nikoletseas, "Sink mobility protocols for data collection in wireless sensor networks," in *Proceedings of the ACM International Workshop on Mobility Management and Wireless Access (MobiWAC '06)*, pp. 52–59, Terromolinos, Malaga, Spain, October 2006.

[3] W. Wang, V. Srinivasan, and K.-C. Chua, "Trade-offs between mobility and density for coverage in wireless sensor networks," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom' 07)*, pp. 39–50, Montréal, Québec, Canada, 2007.

[4] E. Natalizio, V. Loscri, and E. Viterbo, "Optimal placement of wireless nodes for maximizing path lifetime," *IEEE Communications Letters*, vol. 12, no. 5, pp. 362–364, 2008.

[5] T. Razafindralambo and D. Simplot-Ryl, "Connectivity preservation and coverage schemes for wireless sensor networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, Article ID 5977008, pp. 2418–2428, 2011.

[6] W. Allen, A. Martin, and A. Rangarajan, "Designing and deploying a rural ad-hoc community mesh network testbed," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN '05)*, pp. 740–743, Sydney, Australia, November 2005.

[7] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: commodity multihop ad hoc networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123–131, 2005.

[8] T. Razafindralambo, T. Begin, M. Dias De Amorim, I. G. Lassous, N. Mitton, and D. Simplot-Ryl, "Promoting quality of service in substitution networks with controlled mobility," in *Proceedings of the 10th International Conference on Ad-Hoc, Mobile, and Wireless Networks (ADHOC-NOW '11)*, pp. 248–261, Paderborn, Germany, 2011.

[9] R. Charette, "Cable cut in Sydney's central business district," in *Risk Factor blog IEEE Spectrum*, 2009, http://spectrum.ieee.org/riskfactor/computing/it/cable-cut-in-sydneys-central-business-district.

[10] A. M. Townsend and M. L. Moss, "Telecommunications infrastructure in disasters: preparing cities for crisis communications," Tech. Rep., Center for Catastrophe Preparedness and Response, 2005.

[11] S. Commuri and M. K. Watfa, "Coverage strategies in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2, no. 4, pp. 333–353, 2006.

[12] R.-H. Zhang, Z.-P. Jia, and X.-S. Xu, "Nodes deployment mechanism based on energy efficiency in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 1, p. 99, 2009.

[13] J. Carle and D. Simplot-Ryl, "Energy-efficient area monitoring for sensor networks," *Computer*, vol. 37, no. 2, pp. 40–46, 2004.

[14] D. Simplot-Ryl, I. Stojmenovic, and J. Wu, "Energy-efficient backbone construction, broadcasting, and area coverage in sensor networks," in *Handbook of Sensor Networks: Algorithms and Architecture*, John Wiley & Sons, 2005.

[15] K. T. Phan, R. Fan, H. Jiang, S. A. Vorobyov, and C. Tellambura, "Network lifetime maximization with node admission in wireless multimedia sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3640–3646, 2009.

[16] J. Zhang and T. M. Lok, "Cooperative protocols for multiple-source multiple-relay wireless networks," *International Journal of Sensor Networks*, vol. 4, no. 4, pp. 209–219, 2008.

[17] A. Bari, D. Teng, R. Ahmed, and A. Jaekel, "Relay node placement with energy and buffer constraints in wireless sensor networks using mobile data collector," *International Journal of Sensor Networks*, vol. 8, no. 3-4, pp. 147–159, 2010.

[18] J. B. Evans, G. J. Minden, K. S. Shanmugan et al., "Rapidly deployable radio network," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 689–703, 1999.

[19] J. Q. Bao and W. C. Lee, "Rapid deployment of wireless ad hoc backbone networks for public safety incident management," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 1217–1221, November 2007.

[20] A. Wolff, S. Subik, and C. Wietfeld, "Performance analysis of highly available ad hoc surveillance networks based on Dropped Units," in *Proceedings of the IEEE International Conference on Technologies for Homeland Security (HST '08)*, pp. 123–128, Boston, Mass, USA, May 2008.

[21] M. R. Souryal, A. Wapf, and N. Moayeri, "Rapidly-deployable mesh network testbed," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, pp. 1–6, December 2009.

[22] J. Reich, V. Misra, D. Rubenstein, and G. Zussman, "Connectivity maintenance in mobile wireless networks via constrained mobility," in *Proceedings of the IEEE INFOCOM*, pp. 927–935, April 2011.

[23] Network simulator v.2, http://isi.edu/nsnam/ns/.

[24] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *Proceedings of the 7th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 126–134, October 2004.

[25] J. Del Prado Pavon and S. Choi, "Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement," in *Proceedings of the International Conference on Communications (ICC '03)*, pp. 1108–1113, Anchorage, Alaska, USA, May 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 614532, 13 pages
doi:10.1155/2012/614532

Research Article

Using Mobile Robots to Establish Mobile Wireless Mesh Networks and Increase Network Throughput

Cory Q. Nguyen,[1] Byung-Cheol Min,[1] Eric T. Matson,[1,2] Anthony H. Smith,[1] J. Eric Dietz,[1,3] and Donghan Kim[2]

[1] *College of Technology, Purdue University, West Lafayette, IN 47907, USA*
[2] *Department of Electronics and Radio Engineering, Kyung Hee University, Yongin 446-701, Republic of Korea*
[3] *Purdue Homeland Security Institute, Purdue University, West Lafayette, IN 47907, USA*

Correspondence should be addressed to Byung-Cheol Min, minb@purdue.edu

Received 20 January 2012; Revised 5 June 2012; Accepted 20 June 2012

Academic Editor: David Simplot-Ryl

We discuss the proof of concept that gives mobile robotic units the ability to provide a mobile wireless mesh network providing wireless service to end-clients and also demonstrate the ability to increase the throughput of this mobile wireless mesh system by autonomously reducing the hop count required for network traffic to transit through. In doing so, this proof-of-concept contributes to future development of a robust system which can be deployed and utilized in different situations and industry.

1. Introduction

In the 21st century, WIFI and wireless hotspots are familiar services provided to consumers—specifically referring to the 802.11a/b/g/n IEEE standard. In many nations, consumers access data and web content using mobile devices such as tablets, phones, and laptops via WIFI. It has integrated so effectively into our lives and social fabric that we easily take it for granted. What would we do without broadband wireless?

The motivation of the paper is to demonstrate the ability of using mobile robotic units to establish a mobile mesh network. The concept of mobile networks using robotics is nothing of a novel idea as our research group already introduced in [1, 2]; however, the use of robots to form a wireless mesh broadband network is a novel idea and approach to providing broadband wireless service almost anywhere in the world and in any situation. We believe robotics autonomously forming self-healing broadband wireless networks will be the future of wireless services and data communication. Realisticlly, we look to apply this technology first to search-and-rescue situations. However, we introduce this concept with the intent for it to be applicable to many different industries and situations.

The purpose of this paper is to demonstrate a proof-of-concept robotic system that utilizes wireless mesh technology to form autonomous wireless broadband networks. Furthermore, the system will also demonstrate the ability to increase the throughput of the existing system by redistributing the nodes of the existing network given that a precondition is met. Through this proof-of-concept, we hope to show the potential of future networks to be self-forming, adaptable, and self-healing whenever node failures exist in the network. We further emphasize that the use of mesh technology is a differentiating factor from other existing systems with a similar purpose.

In this paper, we will discuss related works and how our system and concept differ from other systems. We further explain and demonstrate how it is more scalable and adaptable. We reference our previous work regarding radio frequency (RF) signal sensing and the use of relative signal strength indicator (RSSI) [3] to control the actions of our robotic units. Thereafter, we apply this basic concept to multiple robotic units and test the system in an open outdoor environment. We will describe the layout and results of our simulation and experiment. Additionally, we will explain the algorithm of our system, demonstrating the proof of concept

of which we hope future research could use as a basis to build upon. Finally, we will discuss the assumptions and limitations that enable this system and concept to work and the future work that is required.

2. Background

Conventional mesh technology is known to be extensible, resilient, self-healing, and provide coverage in hard-to-wire areas [4, 5]. Conventional wireless implementations rely on a wired wireless distribution system (WDS) to communicate between APs for roaming and management; this communication link is known as the backhaul [6]. Mesh technology eliminates the need of a wired WDS. This enables mesh technology to be rapidly deployed with a lower-cost backhaul and allows more flexibility in the configuration of the network based on demand. Additionally, another benefit of wireless mesh technology is the high data rate of 54 Mbps [7]. With these benefits of wireless mesh access point (AP) compared to conventional wireless AP implementation and configuration, wireless mesh APs were used in our system to establish an autonomous wireless broadband networks [8].

We understand the importance of wireless communication; we utilize its services every time we check our email at the airport or coffee shop on our laptops, whenever we turn our phones to WIFI mode when we enter a building to check the news or the weather. Without a doubt, WIFI services are important. The degree of importance of communication and wireless communication is further elevated during instances of natural disasters. According to a World Health Organization (WHO) report after the earthquake of Haiti:

> *Information may be the most important commodity during emergencies. Information may also be the most rapid public health response ahead of the delivery of aid. In addition, the dissemination of information in a timely and transparent manner also helps generate trust and credibility in response activities and agencies providing relief [9].*

During severe natural disasters, like the earthquake of Haiti [10] or the tsunami that hit Japan in 2011 [11], communication infrastructures are either obliterated or so severely damaged that they are rendered inoperable. Hence, rescue teams and responders resort to two-way radios [12] or flying in circles broadcasting emergency messages to victims [13].

Our system or proof-of-concept's design goal is to provide broadband wireless communication. We find that if we can prove the ability to establish a mobile wireless mesh network, we can provide a basis for which this concept can be integrated onto different robotic platforms to be implemented during severe natural disasters. Because mesh network technology is known to be resilient, self-healing, and scalable, we find these qualities appropriate to address the need to fulfill a communication gap during severe natural disasters. Additionally, the ability of a mobile platform is desirable to accommodate for topographical challenges and operational movement of personnel.

There are similar systems and research ongoing with the same purpose and goal as our system; however, none have implemented mesh technology; hence, they do not benefit from the attributes of a mesh network. The US Navy uses robots to deploy network "relay bricks" to extend a single communication link to a robot from a far distance, allowing the operator to control the unit from many miles away [14]. The limitations of this approach are the ability to communicate with one end-device and the immovable nature of the "relay bricks." If the bricks were laid out in a northward direction, but later in the operation, the robot on the far end needed to move west, east, or south, the system would not be able to accommodate due to the immovable bricks.

In London, researchers use mobile robots equipped with ad hoc radios to help officials coordinate search and rescue operations. The use of ad hoc radios, however, limits the system from establishing communication with other nodes through a third-party node; all communication established must be peer to peer using ad hoc radios [15]. Another system known as Autonomous Wireless Aerial Vehicles (AWARE) uses aerial vehicles and personnel to establish a communication medium for emergency personnel. The system relies on aerial vehicles to place static wireless sensors in different locations to provide communication coverage for personnel. The system's static placement of wireless nodes does not scale well to changing environments and conditions or operational needs [16].

Our research concept began using small robotic units, iRobot Create [17], integrated with mesh AP, Proxim-4000 [18]; we successfully demonstrated the ability to have robots perform specified actions based on the RF signal received. We further demonstrated that based on simple RF sensing, our robotic unit was able to complete and optimize a communication link of which two stationary APs were initially placed at a distance and initially unreachable [1]. From this basic concept, we derived a more robust system, of which outdoor robots, P3-AT [19], were equipped with mesh APs with the goal to establish autonomously a linear wireless broadband mesh network [20].

The motivation for this paper is to demonstrate that it is possible to establish a mobile broadband wireless network using mobile robots and wireless mesh technology. The proof-of-concept is evaluated through a two-stage experiment of which the first stage, a wireless mesh network, is established in a linear topology. In the second stage of the experiment, we demonstrate the ability for a robot to redistribute autonomously the network to reduce the hop count network traffic transit, which results in an increase in throughput. The increase in throughput allows for more devices and units to exist on the network and to communicate and transmit data in a timely manner.

3. Concept of RF Signal Sensing

In our previous work, we experimented with the basic concept of RF signal sensing, and based on the RSSI the robotic unit performed a certain action. We denote the RSSI which a robotic unit responds to as the RSSI Threshold.

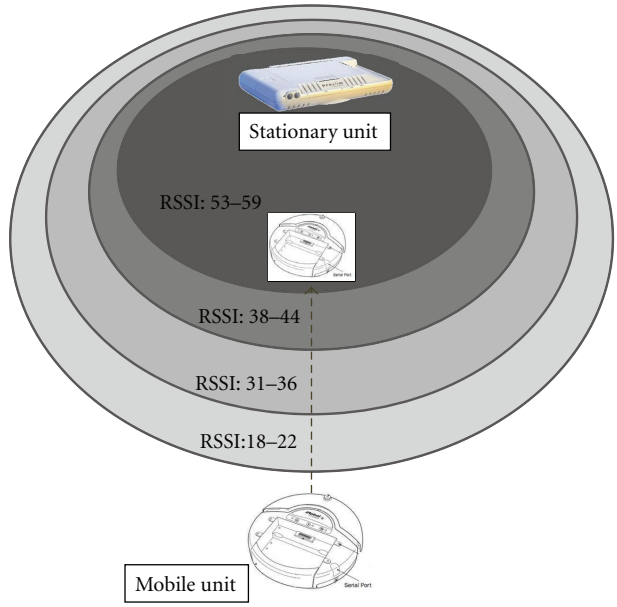


Figure 1: Basic RF sensing using iRobot Create and simple-reflex agent model [1].

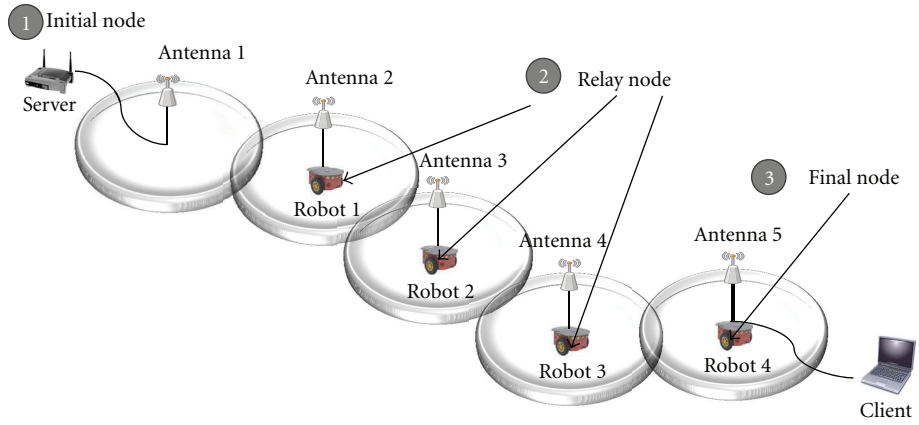


Figure 2: A configuration of self-configurable wireless networks using multiple mobile robots carrying antennas.

Figure 1 demonstrates the robotic unit's ability to stop when it detects an RSSI level greater than 53. This condition could have been applied to any RSSI level, but the condition **If** *RSSI Threshold* ≥ 53 **then** *stop* was used as an example—it is utilizing the simple-reflex agent-based model.

Based on this simple concept of RF sensing and utilizing simple-reflex agent model, we further applied it to our proof-of-concept using mesh technology with more robust robotic units.

4. Algorithm

The algorithm of this proof-of-concept consists of two stages: *linear expansion* (LE) and the *backbone infrastructure route optimization* (BIRO). Essentially, not only is it the experiment demonstrating that robotic units can autonomously create a mobile broadband mesh network but also it is capable of increasing the throughput of the network by autonomously reducing the hop count that network traffic requires to transit through. The basic concept of using multiple robots in the proposed application is shown in Figure 2.

As depicted in Figure 2, the use of multiple mobile robots carrying APs will allow a wireless signal to be relayed from the server to the client. Each robot carries one AP with an internal antenna to form a linear network for long distance coverage. Also, additional antennas and radios can be added to create additional multipoint connections. The number of robots is determined by the requirements of the system, that is, $N_n = N_r$, where N_n is the number of nodes to form a linear link and N_r is the number of needed mobile robot. The set of robotic units in the algorithm can be then represented by $\mathbf{S} = \{L, F, T, R\}$, where L is a leader, F is a set of followers, T is a tail robotic unit, and R is a BIRO robotic unit. Then,

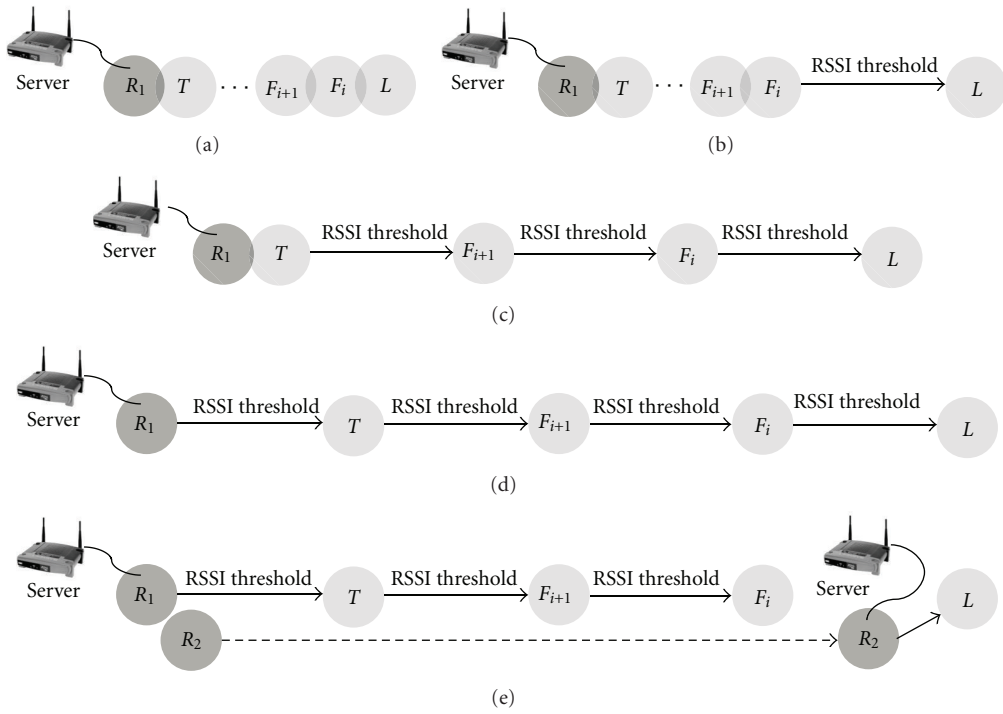


Figure 3: Visual description of LE algorithm: (a) initial state (b) for leader, L (c) for followers, F_i (d) for tail, T (e) visual description of BIRO algorithm.

the number of robotic units in the set **S** is determined by N_n. Their relations can be expressed as follows:

$$N_L = N_T = 1,$$
$$N_F = N_n - 2, \tag{1}$$

where N_L and N_T are the number of the leader and the tail, respectively, and N_F is the number of followers. Note that $N_n = N_r = (N_L + N_F + N_T)$. Thus, in order to form a linear link, N_r should be at least three (one is for the leader, second is for the follower, and last is for the tail robot).

In the first stage of the algorithm, LE, the purpose is to establish a wireless mesh link using multiple mobile mesh APs and a root AP. Additionally, the purpose of the LE stage is to stretch the coverage of the mobile network as far as possible without losing the established connection with the previous node.

LE stage begins with all multiple mobile units associated to the root node, which remains stationary and positioned in a straight convoy formation, as depicted in Figure 3(a). Each unit is assigned a role based on its position in the convoy. The first node is designated as L (leader). The nodes from the second unit to the unit preceding the last in line are designated as F_i (follower), where $i \in \{1 \ldots N_n - 2\}$. The last unit in line is designated as T (tail node).

The algorithm is set up so that the robots are always sensing RF signal to determine their action. In this algorithm, each robot is responsible for sensing the RF of the robot preceding it, and the tail node robot is responsible for sensing the RF of the stationary *root* or *gateway* meshed enabled AP. With this algorithm, if the "*root*" AP is to move closer to the T node unit, then the entire system will move forward to account for the change in *RSSI Threshold* in effect, it would create a ripple effect that is reflected throughout the system.

Each robotic unit looks up their assigned priority and the assigned RSSI threshold limit, and sonar range limit. If robot is assigned "L," the robot will drive straight until the RSSI connection with "F_1" has reached the assigned RSSI threshold; that is, the condition **If** *RSSI Threshold* $\geq$ *certain value* **then** *stop* is activated. This role of the leader is algorithmically summarized in Algorithm 1 and is depicted in Figure 3(b) for a visual explanation.

If robot is assigned priority "F_i," object detection capabilities have been enabled, given the initial close proximity of the robotic units. The use of object detection prevents the robots from running into each other as they are sensing RF levels of their peers at different interval times. Then, the robot will drive straight until the RSSI connection with "F_{i+1}" or with "T" (if the robot is the last follower, i.e., when $i = N_n - 2$) has reached the assigned RSSI threshold. If there is an obstacle within the sonar range, the robot will stop. If obstacle is no longer in range, robot continues to drive if RSSI threshold has not been reached. This role of the follower is algorithmically summarized in Algorithm 2 and is depicted in Figure 3(c) for a visual explanation.

If robot is assigned T, object detection capabilities have been enabled as well, and the robot will drive straight until the RSSI connection with *root*1, R_1, has reached the assigned RSSI threshold. Then, most of its actions are similar with those of the follower. This role of the tail node is

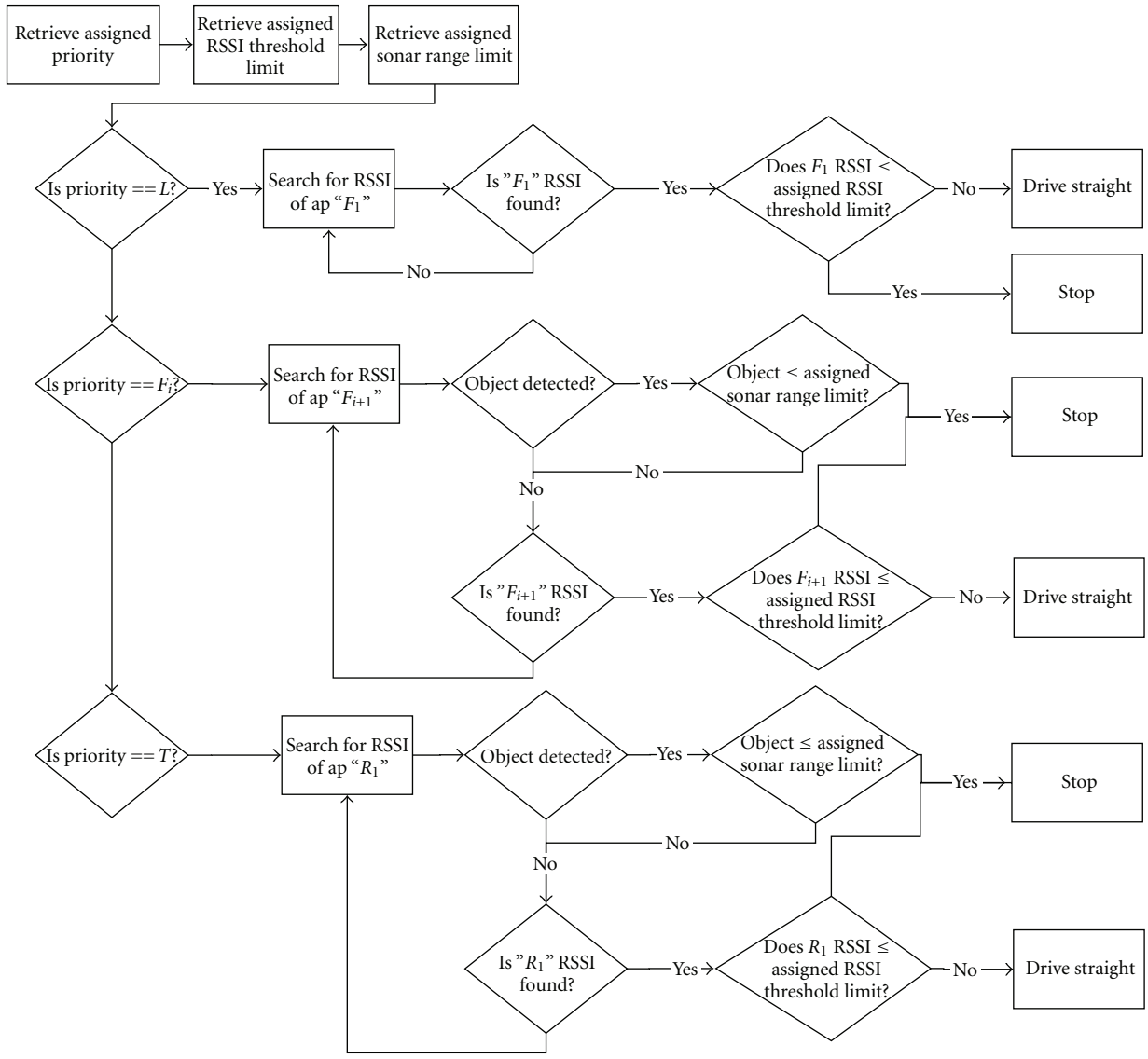


Figure 4: LE Algorithm formalized.

algorithmically summarized in Algorithm 3 and is depicted in Figure 3(d) for a visual explanation. Then, Figure 4 shows a formalization of the LE algorithm.

It is important to mention that the *root* or *gateway* mesh AP, denoted as R_1 in Figure 3, has the ability to redistribute nodes of an existing network. Using indicators such as *RSSI factor, Hop Factor*, and *Roaming Threshold* [3], the AP using its internal designed algorithm makes a decision of how to redistribute the network to reduce the hop count of network traffic. Based on *root* mesh AP ability to redistribute the network, a robotic unit is equipped with a *root* mesh AP, denoted as R_2. In the second stage of the experiment upon the completion of the first stage, the BIRO algorithm initiates.

In the BIRO algorithm, the robotic unit is designated as R_2 (*root*). This unit drives straight scanning its network topology tree until it detected "L," which is the end node of the network. Once the end node of the system is detected, then stop as shown in the algorithm in Figure 3(e). The BIRO algorithm is summarized in Algorithm 4. Then, this action changes the existing network topology tree and reduces the hop count that network traffic form "L" has to travel. Note that the BIRO algorithm is dependent upon the *root* mesh AP's ability to redistribute the network.

5. Simulation

To validate our approach, especially for the LE algorithm, we have built a simulation environment in Simulink of Matlab. We have then run the program in a second time scale to obtain the simulation results that would be the same as those from the real world. Figure 5 represents the entire simulation environment, mainly composed of mobile robots motion block and main block with the LE algorithm (original simulation blocks were from [21] and slightly modified for this research). We have then established several assumptions as follows.

```
/* Stage 1. Linear Expansion Algorithm (LE)                    */
/* For a leader                                                */
if Priority == "L" then
   repeat
      Search for RSSI of AP "F1";
   until "F1" RSSI found
   if "F1" RSSI ≤ assigned RSSI threshold limit then
      Stop driving;
   else
      Drive straight;
```

Algorithm 1: The LE algorithm for a leader.

```
/* Stage 1. Linear Expansion Algorithm (LE)                    */
/* For a follower                                              */
if Priority == "Fi" then /* i ∈ {1 ... Nn − 2}                 */
   if Object detected ∧ Object ≤ assigned sonar range limit then
      Stop driving;
   else
      repeat
         Search for RSSI of AP "Fi+1" or "T" when i = Nn − 2;
      until "Fi+1" or "T" RSSI found;
      if "Fi+1" or "T" RSSI ≤ assigned RSSI threshold limit then
         Stop driving;
      else
         Drive straight;
```

Algorithm 2: The LE algorithm for a follower.

```
/* Stage 1. Linear Expansion Algorithm (LE)                    */
/* For a tail                                                  */
If Priority == "T" then /* Last robot in the line              */
   if Object detected ∧ Object ≤ assigned sonar range limit then
      Stop driving;
   else
      repeat
         Search for RSSI of AP root 1 "R1";
      until "R1" RSSI found;
      if "R1" RSSI ≤ assigned RSSI threshold limit then
         Stop driving;
      else
         Drive straight;
```

Algorithm 3: The LE algorithm for a tail.

```
/* Stage 2. Backbone Infrastructure Route Optimization (BIRO)  */
begin
   View topology tree;
   if "L" is in topology tree then
      Stop driving;
   else
      Drive straight;
```

Algorithm 4: The BIRO algorithm.

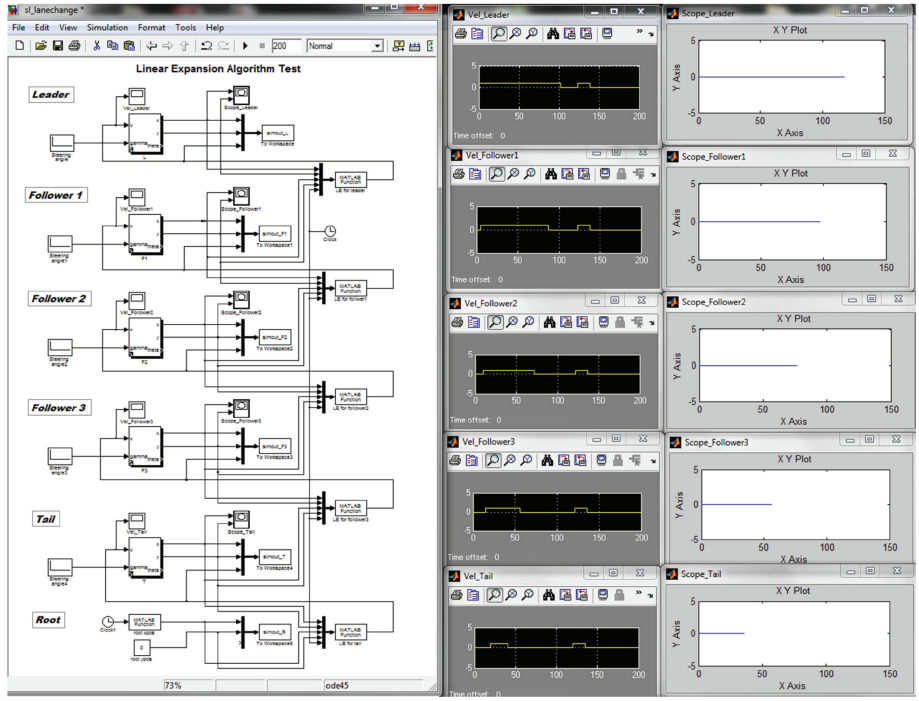


Figure 5: LE Algorithm test environment built in Simulink.

(i) Slew rates of the mobile robots' velocity are instant; that is, acceleration and deceleration of the robot's motion are ignored.

(ii) RSSI values for RF signal sensing are proportional to the distance between neighboring two nodes.

(iii) All robots are located in a same point at the initial simulation run.

The second assumption may differ from the real world application, because the RSSI values are not often proportional to the distance between the two nodes due to the ever-changing RSSI patterns [2]. However, in the *Friis transmission formular* (2), if we assume antenna gains G_R and G_T are equal to 1 and the loss factor also equals 1, then the power level of received signal P_R can be approximated as a function of distance r between the two nodes as a domination factor affecting its value $P_R/P_T \approx 1/4\pi r^2$:

$$P_R = G_R\left(\frac{\lambda^2}{4\pi}\right)\frac{G_T}{4\pi r^2}P_T, \tag{2}$$

where P_R and P_T are the power level of received and transmitted signal, G_R and G_T are the gains of the receiving and transmitting antennas, r is the distance between the antennas, and $(\lambda^2/4\pi)$ is called the *free-space loss factor*.

For simulation, we have used a simple model of the mobile robot having a motion equation as [21]

$$\begin{aligned}\dot{x} &= V\cos\theta,\\ \dot{y} &= V\sin\theta,\\ \dot{\theta} &= \frac{V}{L}\tan\gamma,\end{aligned} \tag{3}$$

where x, y, θ, V are the robot's x position, y position, heading angle, velocity in the world frame. L is a length between two wheels, and γ is the angle of the steered wheel. Note that the position of the robot having this motion equation is controlled by the robot's velocity manipulation in general. We have used five mobile robots ($N_n = N_r = 5$) in this simulation and have denoted them as L, F_1, F_2, F_3, and T, respectively. All robots and the root node are initially located at the same origin $(0, 0)$, and the robots are set to move only to the x direction with the same velocity, $V = 1$ m/sec. The robots heading angles θ are all set to 0°. To avoid running into each other, all robots start moving with 5-second time interval. We have set the value of the condition, **If** *RSSI Threshold* (here, distance-based) $\geq$ *certain value* **then** *stop*, as 20 meters. With all these presettings, if the LE algorithm works properly, the leader mobile robot L will move and locate at 100 meters away from the origin because there will be 5 nodes established, and each node has a capacity of 20 meters connectivity. In addition, since the robots position is controlled by its velocity manipulation, the displacement D_L can be calculated by $D_L = \int_0^t V_L dt$, where V_L is the leader robots' velocity and t is the total simulation time.

Figure 6 shows the linear formation was generated over about 110 seconds of the simulation time. In Figure 6, time transitions during the simulation are depicted on top right figures. At the beginning of simulation, all robots and the root node were located in the origin. Then, the first mobile robot, denoted as L, started moving forward. This robot kept moving until the stop condition was activated at 105 sec (look at the top graph in Figure 8). The second mobile robot then, denoted as F_1, followed the first robot until the stop condition, based on the distance between its position and

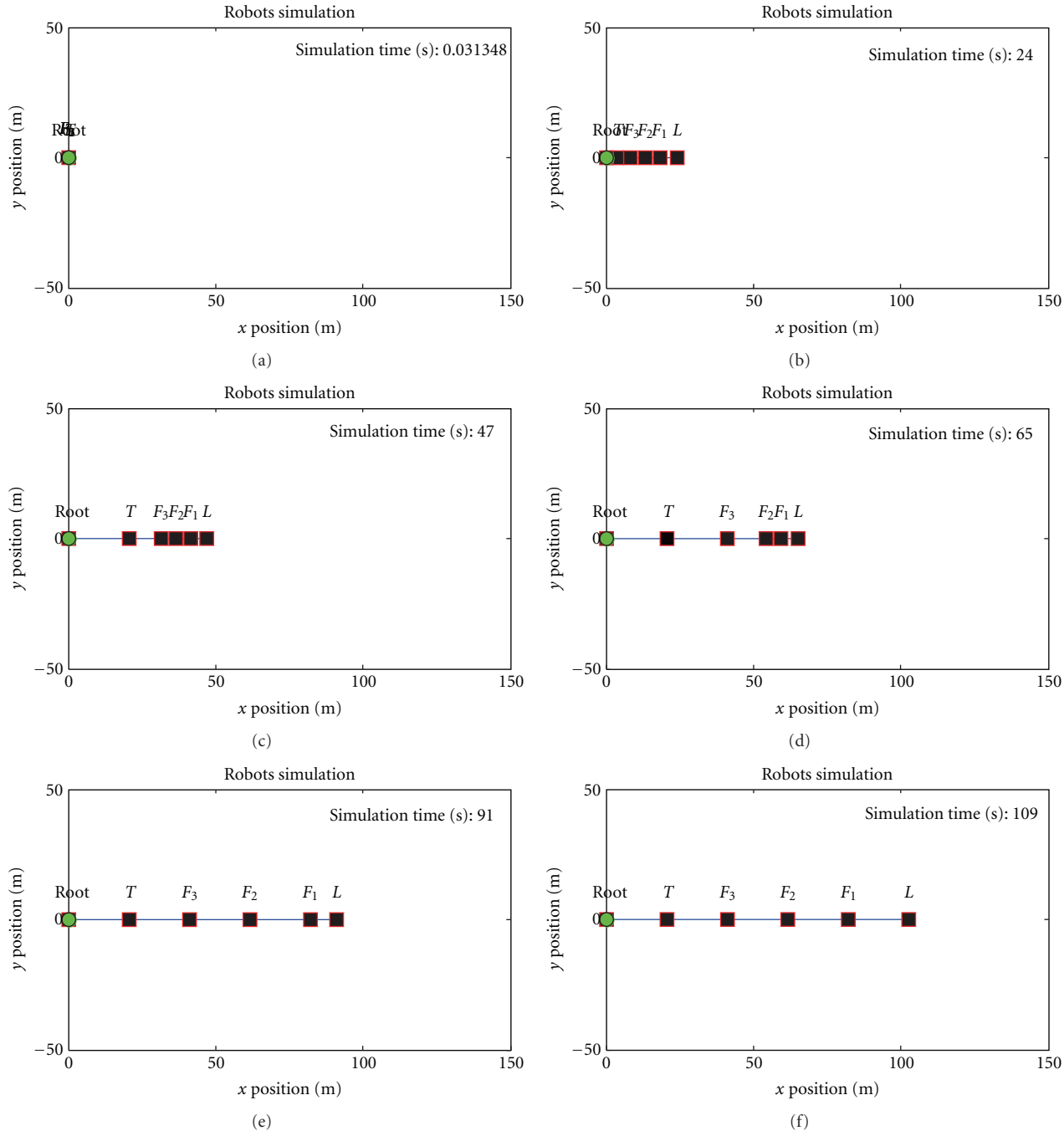


Figure 6: Linear formation results using LE Algorithm.

the third robots position, was activated at 90 sec. The third robot, the fourth robot, and the fifth and the last robot, denoted as F_2, F_3, and T, started moving at 10, 15, and 20 sec, respectively. And then they also kept moving until their stop conditions are activated. The all robots stopped eventually at 105 sec in the simulation, and the final formation is depicted in the bottom right figure in Figure 6. As shown in the final figure, all robots were correctly located with 20-meter distance interval each other, and the leader robot L is located at (100 m, 0 m) as expected. You can also verify this linear formation result from graphs (from 0 sec to 110 sec) in Figure 8 showing all robots' velocity history along with time. Consequently, from these two figures (Figures 6 and 8), we could confirm that the LE algorithm successfully forms a linear formation with multiple robots satisfying the given condition.

After the complete of the first linear formation (around 110 sec), we have investigated the effect to the entire system when the *root* AP moves closer to the T node unit. In this simulation (from 120 sec to 200 sec), we have intentionally

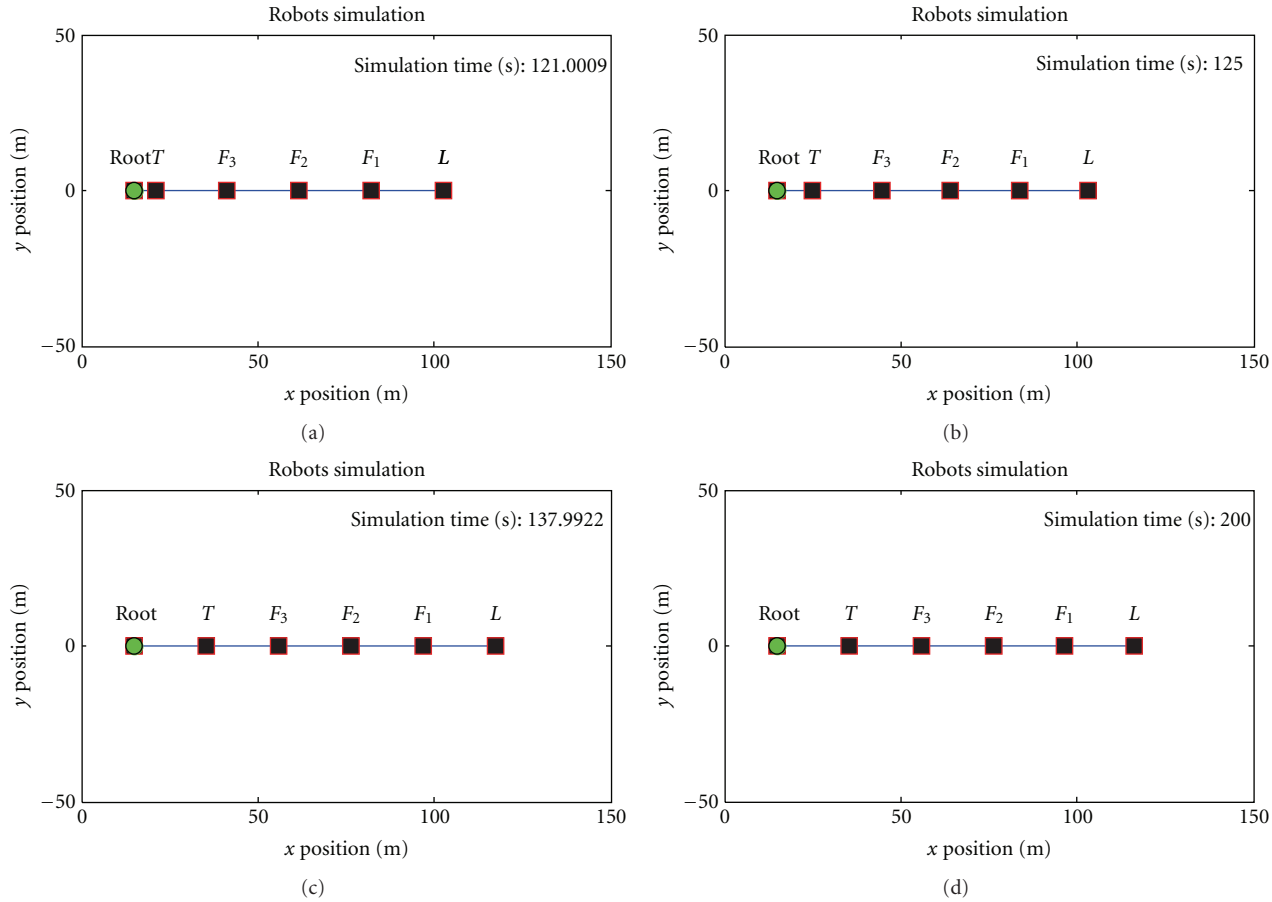


Figure 7: Effects of relocating of *root* node.

moved the *root* AP to 15 meters to the right direction from the beginning of simulation (120 sec) so that it could be closer to the tail node T. The results of this additional simulation are depicted in Figure 7. As shown in Figures 7 and 8 around at 120 sec, as soon as *root* was closer to the T, the distance-based condition of the most left node was deactivated, and then the T resumed moving forward. The movement of the T resulted in having F_3 move forward as well, and finally the shift of the root initial position moved the entire system to the right direction sequentially, from the left to the right. Eventually, the leader robot L ended up locating at (115 m, 0 m) as expected. Consequently, from this additional simulation, we could also confirm that all nodes already established with the LE algorithm could move again in a convoy formation by means of the relocating of "*root*" position.

6. Experiment Setup

Before the experiment could be executed, 4 mobile P3-AT (three are for the LE, and one is for the BIRO, so the number of nodes and robots in the algorithm becomes $N_n = N_r = 3$ in the LE), 5 Proxim-4000, and 6 laptops were required with crossover cables. Three P3-AT each equipped with a Proxim-4000 and the LE algorithm, installed on each laptop, were positioned in a convoy formation. Each AP on each P3-AT was set to *Mobile AP* mode [3]. The R_1 mesh AP was placed stationary at the end of the convoy, and the AP was set to *Mesh Portal* mode [3].

On the left of the tail node T, a P3-AT equipped with a Proxim-4000 is set to *Mesh Portal* mode and remains until the first stage of the experiment is completed. Two laptops are equipped with Iperf [22]; one is set as the server, and the other is set as the client. Additionally, crossover Ethernet cables are used to ensure that the laptops are associated to their assigned units and do not associate wirelessly to other units; this ensures validity in the experiment.

The experiment is executed in two stages: LE and BIRO. Procedures for the first stage of the experiment, LE, are depicted in Figure 9(a) and as follows.

(1) Initiate and start all robotic units, execute LE algorithm.

(2) Wait for all units to stabilize (stop) for good.

(3) Start Iperf server connected via crossover to R_1.

(4) Connect Iperf client connected via crossover to L.

(5) Collect throughput measurement three times, each measurement collected in 10 seconds interval.

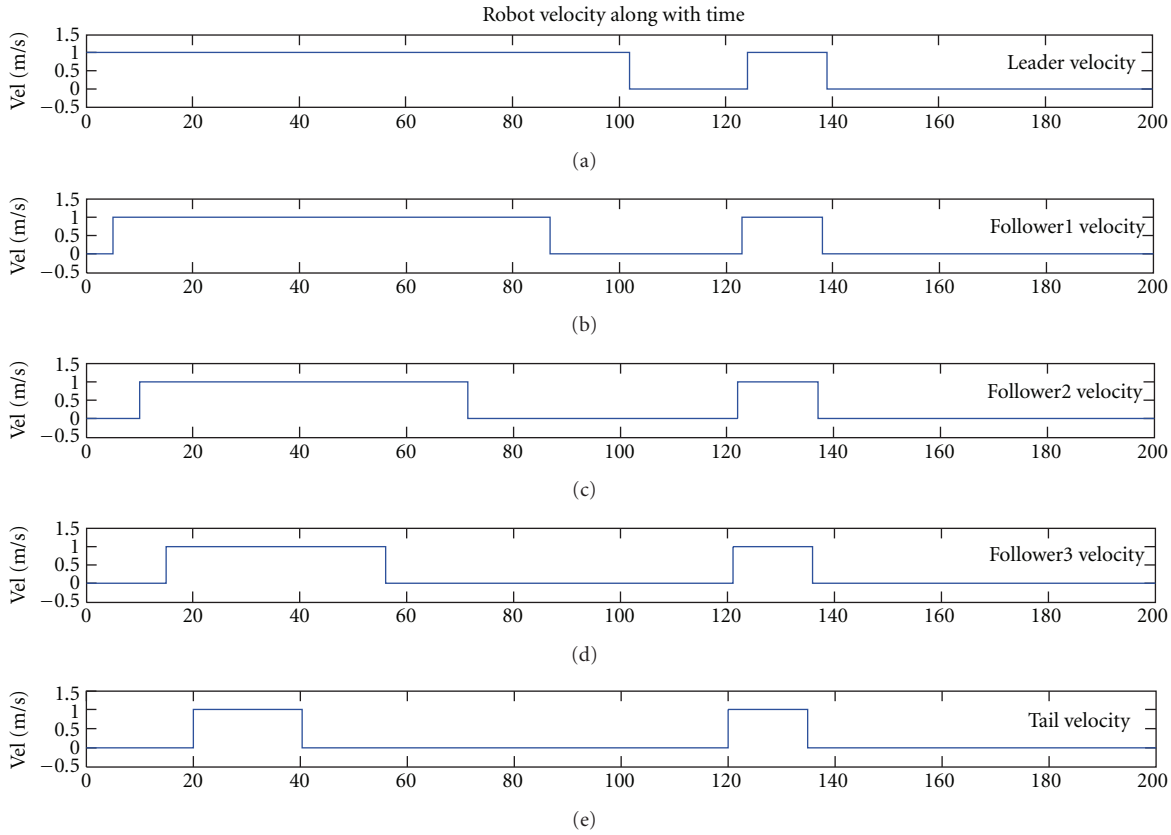
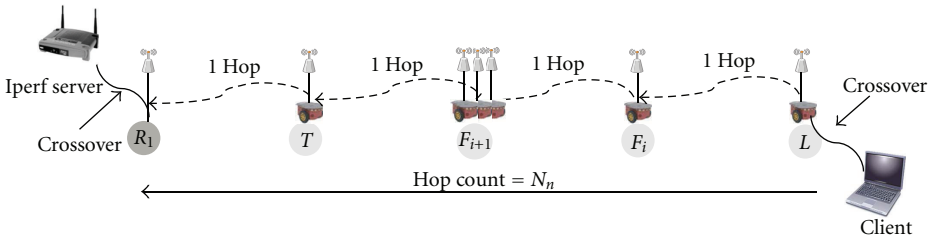
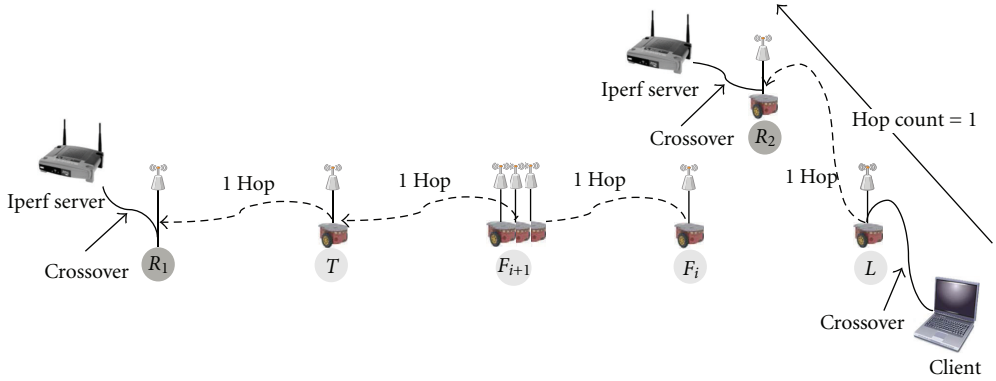


Figure 8: Mobile robots' velocities from the LE simulation.



(a) Data collection of LE stage of algorithm



(b) Data collection of BIRO stage of algorithm

Figure 9: Stage transition from LE to BIRO.

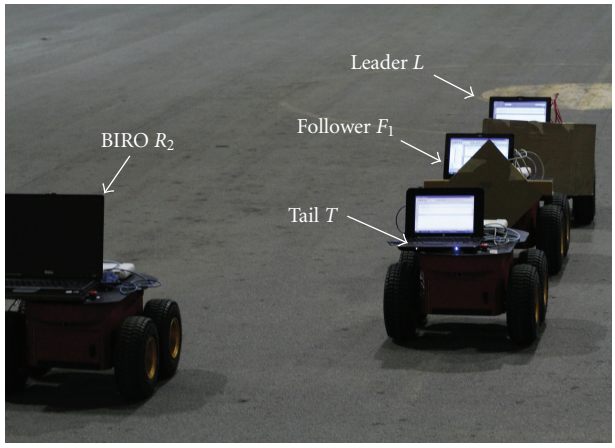


Figure 10: Initial state of system, robots in a convoy formation.

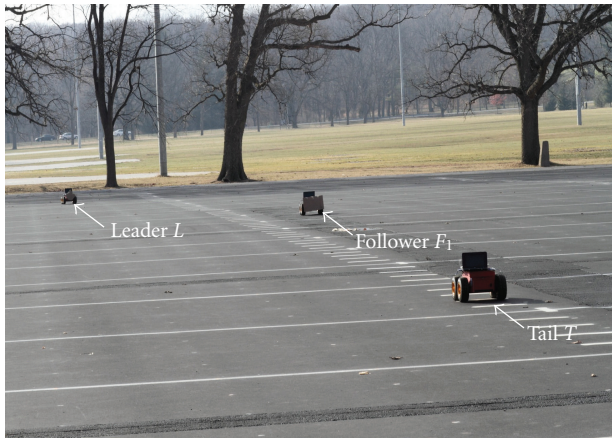


Figure 11: Outdoor field experiment of LE stage of experiment.

After the completion of the LE stage, the BIRO stage executes as depicted in Figure 9(b) and as follows.

(1) R_2 initiates and executes BIRO algorithm.

(2) Wait for R_2 to stabilize (stop).

(3) Start Iperf server connected via crossover to R_1.

(4) Connect Iperf client connected via crossover to N_1.

(5) Collect throughput measurement three times, each measurement collected in 10 seconds interval.

The entire 2-staged experiment was repeated 10 times. Over 300 data points were collected as a result. Additionally, the *RSSI Threshold* set for this experiment was 30.

7. Results and Analysis

The results of the experiment were a linear formation of robotic units connected via wireless mesh APs using internal antennae. The backhaul was using 802.11a, and 802.11g was used to provide wireless data service to end-users and end-devices. Figure 10 shows the robots in formation and ready for deployment, and Figure 11 shows the three robotic units stretched over a football field.

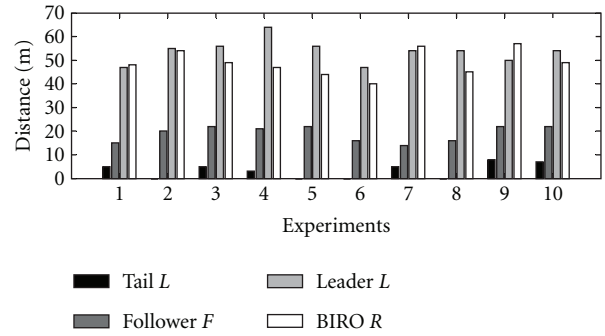
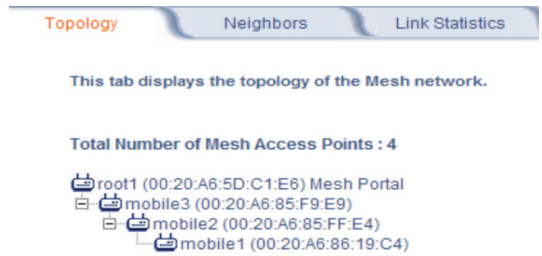
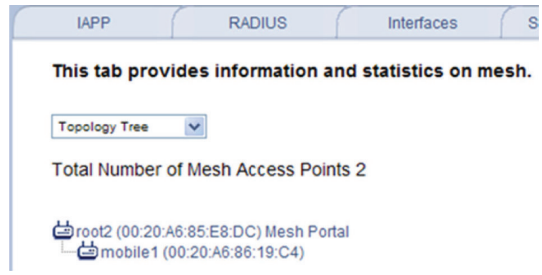


Figure 12: Distances from outdoor field experiment of LE stage and BIRO stage.



(a) LE stage of experiment



(b) BIRO stage of experiment of R_2

Figure 13: Network Topology.

Figure 12 shows the three nodes stretched over a distance up to 70 meters. This figure proves that a short communication link with one node could be extended to a far distance using multiple robots in a linear link. Also, it proves that the BIRO node successfully detected the end node "L" and stopped around it.

In the LE stage of the experiment, when all three mobile units were stretched out in a linear formation, the R_1 AP established the following network topology in Figure 13(a). As shown in Figure 13(a), mobile1 L associates with mobile2 F_1 and mobile2 associates with mobile3 T, which finally associates with the *root* AP R_1. With this topology, we could guarantee that network traffic would transit three hops before reaching the *root* AP.

Following the completion of the LE stage, the BIRO stage was executed. Upon the completion of the BIRO stage, the following network topology was established as shown in Figure 13(b). As shown in Figure 13(b), *root* AP R_2 was able

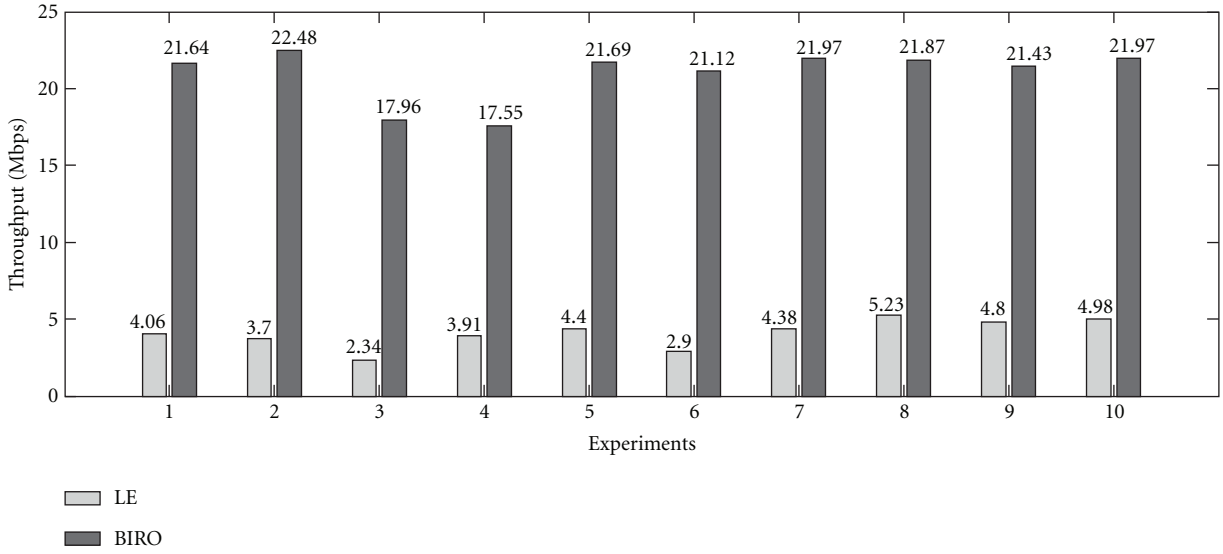


Figure 14: Network throughput performance over 10 experimental runs.

to redistribute the network so that mobile1 L associated to R_2 and mobile2 F_1 and mobile3 T are associated with R_1. By associating to the new *root* AP, clients connected to mobile1 L transit only one hop count to reach the *root* or *gateway* AP. It is an assumption of this experiment that once clients are able to reach the *gateway* in R_2, they have access to the same network resources and services as if they were connected through the *gateway* in R_1.

From the collection of network throughput from 10 experimental runs, Figure 14 shows the throughput performance of the experiment for the LE and BIRO stage of the experiment. From the chart in Figure 14, it is observed that by reducing the hop count during the BIRO stage of the experiment, the throughput was significantly greater than the throughput performance during the LE stage of the experiment. During the LE stage, network traffic transits through 3 hops, compared to 1 hop during the BIRO stage. The chart also shows the numerical data of the throughput collected for each experimental run. During each experiment of each stage, the collection of throughput data was measured 3 times for 10 seconds. The average was calculated from the 3 measurement instances taken from each experiment of each stage.

As a result, Figure 14 shows that robotic units have the ability to optimize a mobile wireless mesh network by reducing the amount of hop counts a network much traverses. Finally, this experiment shows that it is not only possible to establish a mobile wireless mesh network but it is also possible to improve the existing mobile wireless mesh network with additional robotic units equipped with *gateway* AP that have the ability to redistribute the wireless mesh network.

8. Assumption and Limitations

The success of the experiment and the future applicability of this research concept run under the assumption that the *root* APs, R_1 and R_2, have a way of connecting to each other, either through the use of external omni antennae or directional antennae or have the ability to connect to a stationary point-to-multipoint WIFI system. A successful connection between R_1 and R_2 would allow clients connected to nodes associated to R_1 can connect and communicate with clients connected to nodes associated to R_2.

Additionally, the three mobile AP units were limited to the use of internal antennae, and the units were only mounted 0.5 meter above the ground. This low-level mount of the AP significantly reduces the distance covered by each mobile AP unit and the performance of the overall system. Additionally, the units were limited to a small power source lasting only 4 hours.

9. Future Work and Conclusion

Future work will include the implementation of a system that utilizes external antennae, both omnidirectional and directional. Additionally, a new mounting system will be designed to increase the height of the antennae to increase coverage and network performance for the system.

With the given experiment, we have successfully proved the proof-of-concept of establishing mobile broadband mesh networks using mobile robots. Additionally, we have shown that we can further improve network performance by reducing the hop count of the network traffic from the furthest network node, all of this operation done robotically and autonomously. If R_1 was to move closer to tail node T, the entire system would autonomously shift to accommodate the change. We further emphasize the successful redistribution of the network through topology trees of the first and second stage of the experiment.

We hope that through the use of the simple concept of RF sensing, the concept of mobile broadband mesh network can be further developed and matured to be implemented on different robotic platforms paving way to a new era of

wireless communication and services by providing wireless communication almost anywhere and at anytime in the world.

References

[1] C. Nguyen, B. Leong, E. Matson, A. Smith, and J. Wachs, "AWARE: autonomous wireless agent robotic exchange," in *Proceedings of the International Conference on Intelligent Robotics and Applications (ICIRA '10)*, Shanghai, China, Noember 2010.

[2] B.-C. Min, J. Lewis, D. K. Schrader, E. T. Matson, and A. Smith, "Self-orientation of directional antennas, assisted by mobile robots, for receiving the best wireless signal strength," in *Proceedings of the IEEE Sensors Applications Symposium (SAS '12)*, pp. 261–266, February 2012.

[3] Proxim Wireless, Orinoco AP-4000, AP-4000M and AP-4900M User Guide, Proxim wireless, 2006.

[4] "802. 11s IEEE standard," 2011, http://standards.ieee.org/findstds/standard/802.11s-2011.html.

[5] S. Conner, "IEEE 802. 11s tutorial oeriew of the amendment for wireless local area mesh networking," 2006, http://ieee802.org/802_tutorials/06-November/802.11s_Tuto-rial_r5.pdf.

[6] "Conguring wds, fast secure roaming, and radio management," 2012, http://www.cisco.com/en/US/docs/wireless/access_point/12.2_15_JA/configuration/guide/s15roamg.html.

[7] Proxim wireless mesh series, 2012, http://www.proxim.com/products/enterprise-wireless-lan-wi-mesh/orinocor-wi-mesh-series.

[8] A. Smith and E. T. Matson, "Use of antennas as sensors to discover signals to form mobile broadband networks," in *Proceedings of the IEEE Sensors Conference (SENSORS '09)*, pp. 581–584, October 2009.

[9] WHO, Initials, "Public health risk assessment and interentions: earthquake—Haiti," World health organization, 2010, http://www.who.int/diseasecontrol_emergencies/publications/who_hse_gar_dce_2010_1/en/index.html.

[10] "Rescue crews pull 2 more from haitian market," MSNBC, 2010, http://www.msnbc.msn.com/id/34829978/ns/world news-americas/?GT1=43001.

[11] "Japan earthquake: death toll passes 10, 000," 2011, http://www.telegraph.co.uk/news/worldnews/asia/japan/8405619/Japan-earthquake-death-toll-passes-10000.html.

[12] K. K. Hoon and C. Meyers, "Snow muffles rescue work in japan's deastated northeast," 2011, http://www.reuters.com/article/2011/03/16/us-japan-quake-snow-idUSTRE72F2WN-20110316.

[13] M. M. Ahlers and M. Mount, "Radio station in the sky warns haitians not to at-tempt boat oyage," 2010, http://articles.cnn.com/2010-01-19/world/haiti.broadcast.warning_1_haiti-migration-coast-guard?_s=PM:WORLD.

[14] N. Pezeshkian, H. G. Nguyen, and A. Burmeister, "Unmanned ground vehicle radio relay deployment system for non-line-of-sight operations," in *Proceedings of the 13th IASTED International Conference on Robotics and Applications (RA '07)*, pp. 501–506, August 2007.

[15] S. Timotheou and G. Loukas, "Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios," in *Proceedings of the 24th Annual ACM Symposium on Applied Computing (SAC '09)*, pp. 1171–1175, March 2009.

[16] A. Ollero, P. J. Marron, M. Bernard et al., "AWARE: platform for autonomous self-deploying and operation of wireless sensor-actuator networks cooperating with unmanned AeRial vehiclEs," in *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR '07)*, pp. 1–6, September 2007.

[17] irobot create., 2012, http://store.irobot.com/shop/index.jsp?categoryId=3311368.

[18] Orinoco ap-4000, 2012, http://www.proxim.com/products/enterprise-wireless-lan-wi-fi-mesh/orinocor-ap-4000.

[19] Mobile robots., 2012, http://www.mobilerobots.com/Research-Robots/P3AT.aspx.

[20] C. Q. Nguyen, *Intelligent autonomous mobile wireless broadband linear expansion & route optimization [M.S. thesis]*, Purdue Uniersity, 2011.

[21] P. I. Corke, *Robotics, Ision & Control*, Springer, 2011.

[22] J. Dugan and M. Kutzko, "Iperf," 2011, http://sourceforge.net/projects/iperf/.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 951213, 10 pages
doi:10.1155/2012/951213

Research Article

Localization of Mobile Sensors and Actuators for Intervention in Low-Visibility Conditions: The ZigBee Fingerprinting Approach

Jose V. Marti,[1] Jorge Sales,[1] Raul Marin,[1] and Ernesto Jimenez-Ruiz[2]

[1] *Computer Science and Engineering Department, Jaume I University, Avenida Vicente Sos Baynat s/n, 12071 Castelló de la Plana, Spain*
[2] *Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, UK*

Correspondence should be addressed to Jose V. Marti, vmarti@icc.uji.es

Received 23 December 2011; Revised 29 May 2012; Accepted 18 June 2012

Academic Editor: Hannes Frey

Indoor localization in smoke conditions is one of the EU GUARDIANS project goals. When smoke density grows, optical sensors such as laser range finders and cameras cease to be efficient. Zigbee sensor networks provide an interesting approach due to the fact that radiofrequency signals are propagated easily in such conditions. Moreover, they permit having an alternative communication infrastructure to the emergency brigades, allowing also the implementation of localization algorithms for the mobile sensors, actuators, and firefighters. The overall localization method (i.e., ARIEL) aims to acquire the nodes position in real time during an intervention, using different sensor inputs such as laser, sonar, Zigbee, and Wifi signals. Moreover, a fine grained localization algorithm has been implemented to localize special points of interest such as emergency doors and fire extinguishers, using a Zigbee programmable high-intensity LED panel. This paper focuses on the Zigbee fingerprinting localization method used to obtain the position of the mobile sensors and actuators by training a database of radio signals for each scenario. Once this is done the proposed recognition method runs in a quite stable and accurate manner without needing any sophisticated hardware. Results compare the procedure with others such as KNN, and neural networks, demonstrating the feasibility of the method for a real emergency intervention.

1. State of the Art

Localization of mobile sensors and actuators is an active research field that becomes even more interesting and necessary in indoor applications such as fire emergency interventions, where the GPS is either not accessible or not practical to be used [1, 2].

First of all, some works use the laser range finder as a way to obtain the position of a mobile system in indoor environments [3, 4]. This solution is quite straight forward when the geometrical map of the building is well known, including the furniture. Other works focus on using visual landmarks to localize the mobile systems through vision cameras [5, 6]. These two alternatives are very accurate in situations of good visibility (e.g., nonsmoke conditions), although they are expensive to be implemented.

Moreover, in the sensor networks community, several interesting localization methods based on radio-frequency signals can be found, which can be transmitted in smoke conditions. In fact, some techniques have recently been proposed for determining the position of mobile nodes by measuring this kind of signals, such as time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA), received signal strength (RSSI), and others [7–9]. In particular, the TDOA method can use a radio signal combined with a sonar. By measuring the difference in time of flight between the radio and the sonar signals, one can estimate the distance between the transmitter and the receiver in a very accurate manner [10, 11] although some extra work must be done to avoid the effect of reflections.

Radiofrequency allows the distance between transmitter and receiver to be calculated by measuring the RSS (Received Signal Strength) and applying to it the propagation/attenuation model represented by (1):

$$\mathrm{RSS} = A * d^{-n}, \tag{1}$$

where A is the RSS at 1 meter from the transmitter, d is the distance between transmitter and receiver, and n is the

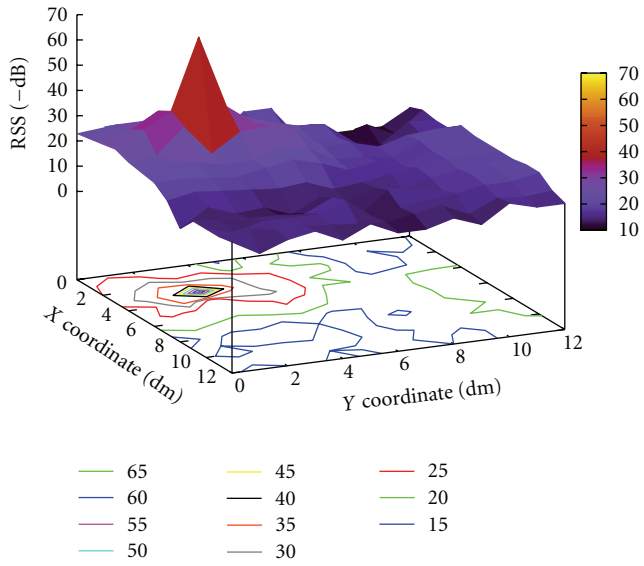


Figure 1: Signal strength distribution in an obstacle-free environment (outdoors).

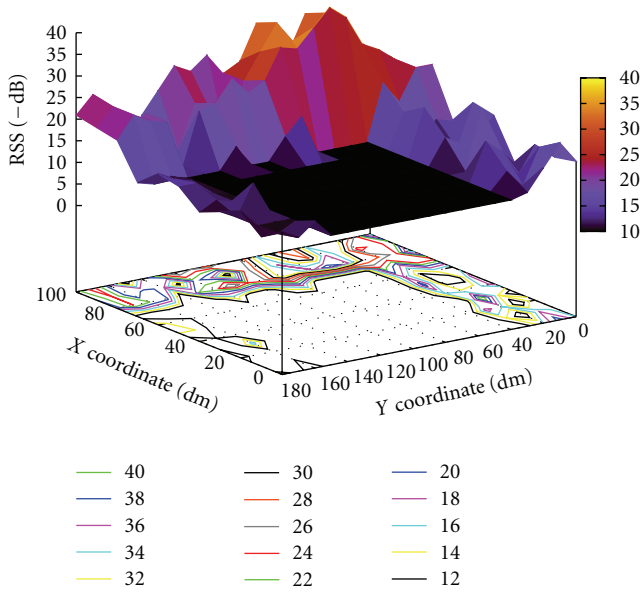


Figure 2: Signal strength distribution in an irregular environment (indoors).

propagation factor. In fairly open outdoor areas this is a suitable method to calculate distances, since there are no reflections nor interference and signal strength distribution is very clean, as shown in Figure 1. However, due to the unpredictable behavior of radio signals in indoor scenarios with irregular geometries and materials, other techniques must be studied, due to the fact that the RF behavior is affected significantly by these factors. For example, in Figure 2 the RF map is shown for a corridor that has stairs in the middle, including different kinds of metallic materials. The black area corresponds to the stairs hole, where the robot cannot be positioned and where it was not possible to take any measures. Some methods such as RADAR [12] combine the empirical measurements and propagation model taking into account some geometrical characteristics of the environment, such as the presence of walls, improving the efficiency of the propagation/attenuation model. Other systems (see Youssef and Agrawala [13]) use probabilistic techniques, such as Bayesian estimation, to obtain the most probable transmitter position.

Fingerprinting methods consist of measuring the signal strength values to build a radio-frequency database model and then compare the navigation measures with those previously stored using pattern recognition techniques. These methods have the disadvantage of needing a previous training procedure for every location of a given scenario, and moreover, they adapt very well to the specific behavior of radio signals for a given space, which are affected by the particular characteristics of reflection, absorption, diffractions, and others, as explained in [14].

The *ZigBee* sensor network infrastructure is specially interesting for implementing fingerprinting localization methods, as it can be easily integrated in a building, offering many possibilities to control the radio signals characteristics such as power and frequency and enhancing the capacity of the trained radio map.

2. Introduction

In the frame of the EU GUARDIANS [15] project, a multisensor localization system has been developed in order to be able to obtain the localization of mobile robots and fire fighters inside a building during an intervention. The system, called ARIEL [16], uses different sensor inputs to calculate the positions (e.g., laser, sonar, *WiFi,* and *ZigBee* fingerprinting) and decides which one is the optimum at every moment depending on the environmental conditions (smoke density).

For example, when the smoke density is low, laser range finder sensors are still able to localize the nodes (Monte Carlo Localization method [17]), with a small positioning error of approximately 10 cm, once the building map is available and the structure of the building has not been affected. When the laser range finder detects a significant amount of smoke it considers it as an obstacle, so the *ZigBee* fingerprinting methods become a suitable alternative to have an approximate idea of the position, as we will see in the next sections.

Moreover, visual positioning based on visual servoing techniques [18] provide a fine-grain localization when the distance to the point of interest is reduced. For that, the ARIEL system provides a programmable *ZigBee* node that has a high luminosity LED panel attached on it, which can be perceived by the onboard camera in smoke conditions [16].

The present paper focuses on the radiofrequency localization method that has been implemented within the ARIEL system to obtain the nodes position in smoke-filled indoor areas. The paper compares several pattern recognition algorithms, in terms of efficiency and needed hardware complexity. Results show that the proposed fingerprinting method is

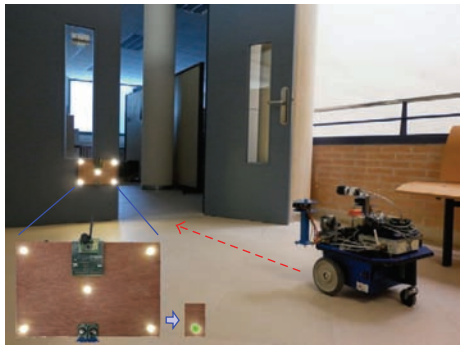


Figure 3: Mobile robot and the remotely controlled high-intensity LED panel.

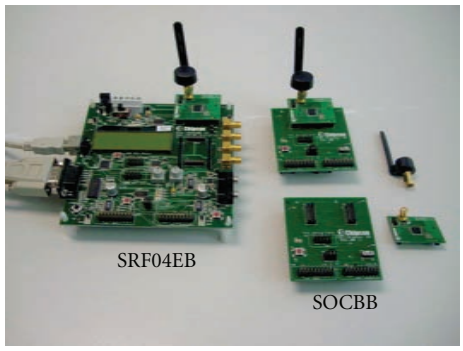


Figure 4: ZigBee communication modules.

Table 1: Channel-frequency matching.

Channel	Frequency
11	2405 MHz
13	2415 MHz
16	2430 MHz
19	2445 MHz
22	2460 MHz
26	2480 MHz

Table 2: Value-power matching.

Value	Gain
255	0.6 dB
95	−0.4 dB
19	−5.7 dB
3	−25.2 dB

suitable to be used in real interventions once the radio map for the given scenario is known through a training phase.

3. Hardware Description

The transceivers used are based on the CC2430 and CC2431 *Texas Instruments* microcontrollers and meet the *ZigBee* specification, with the capacity to obtain the RSS (Received Signal Strength) from every received packet. Moreover, 16 different channels can be configured with 256 different power levels. This fact has been used to increase the number of packets sent between the beacons and the mobile sensor at each robot position to improve the efficiency of the localization method.

On the other hand, the CC2431 microcontroller includes the *Location Engine* system that estimates the distance between each beacon and the transmitter by knowing the original signal intensity and the propagation coefficient of the medium. Then, by using three or more beacons, the system can triangulate the transmitter's position. This will allow us to compare the proposed fingerprinting localization method performance with the *Location Engine* mentioned above, as in [19], where it is easy to see that this method works well in open spaces but does not work properly indoors.

The experiments have been performed by using four transmitters in known positions (beacons) and one mobile transmitter, the position of which is going to be calculated. The whole sensor network information comes to one PC computer, carried by the robot, which calculates in real time the mobile transmitter position.

In summary, two different types of communication modules (nodes) shown in Figure 4 are involved in the measurements as follows.

(i) SRF04EB (Serial Radiofrequency Evaluation Board): this board is going to be connected to a PC through a RS232 interface and will be used as a base station to send commands to the mobile transmitter and receive measurements from it.

(ii) SOCBB (System On Chip Battery Board): this is the most simple board to hold a CC243X. They will be used for two possible functionalities.

  (a) Mobile transmitter: this node will receive commands from the base station, perform the measurements, and send the results back to the base station (Figure 3).

  (b) Simple beacon: there are four beacons located at fixed positions that simply return every packet received, including the RSS value.

4. System Training

In general, the proposed fingerprinting method works in two phases: training and localization estimation. In this section we will describe the experiments performed in order to obtain the measurements corresponding to the training phase and, in the next section, the ones used to calculate the transmitter position. Three different scenarios have been used: (1) garden (Figure 5), (2) classroom (Figure 6), and (3) corridor (Figure 7). Each of them has specific characteristics that will affect the signal propagation and RSS measurements.

The training procedure involves taking RSS measurements in different transmitter positions. In these measurements, beacons are placed at fixed positions, and the transmitter is located at every position of the scenario, using a certain density mesh (typically 50 cm by 50 cm). Then, data

Figure 5: Scenario 1: Garden. B1, B2, B3, and B4 showing the beacons positions. Green dots are the different transmitter measurement positions.

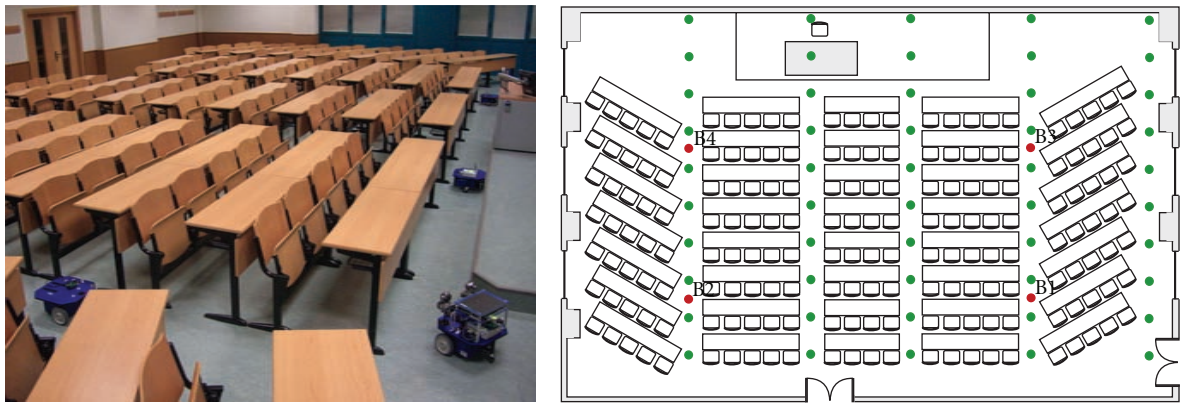


Figure 6: Scenario 2: classroom.

packet transmissions are made in different channels (frequencies) and using different power levels. In fact, for this experiment we used six channels and four power levels in order to cover the whole parameter range provided by the *Texas Instruments* transmitters used.

Specifically, for given used channels, the corresponding frequencies can be calculated with (2):

$$F = 2405 + 5 * (\text{ch} - 11) \quad \text{MHz}, \tag{2}$$

where *ch* is the channel number, which must take a value between 11 and 26. Then, channels 11, 13, 16, 19, 22, and 26, used in this experiment, correspond to the frequencies shown in Table 1. Also, the different power levels used can be seen in Table 2, where the first and last values are, respectively, the maximum and the minimum power the transmitter can generate.

The interference pattern distribution for these frequencies present a distance between nodes in the order of a few centimeters. Modifying the frequency of the transmitter will produce different interference patterns at the same transmitter location, as seen in Figure 8, and this will provide additional information to the location characterization measurements dataset.

For every combination of beacon, channel, and signal power, five packets are sent from the transmitter (mobile sensor), which are sent back to the transmitter with the measured RSS value. This is done in order to have some statistical component in the data collected, avoiding spurious values.

To perform the training procedure the mobile sensor is placed at every position of the scenario (green dots in Figures 6 and 7), so that every RSS for every combination of beacon, channel, and signal power may be stored. The actual coordinates are also saved in the data base.

When a beacon receives a packet from the transmitter, it calculates its RSS and returns as confirmation a packet with a four-byte payload, as shown in Figure 9, where the beacon x and y coordinates in decimeters are sent in the first and second bytes. The third byte contains the beacon identification number and the fourth byte contains the obtained RSS value in—dB (i.e., a positive number between 0 and 90).

If a confirmation packet from the beacon is not received by the transmitter in a configurable amount of time, the transmitter sends a retry packet. This operation is repeated a configurable number of times. Finally, if no response is received, the transmitter sets the RSS to a minimum value of −99 dB for this particular combination of power, channel and beacon.

For every received packet, the transmitter measures the RSS and, with the beacon RSS, builds a pair of values that

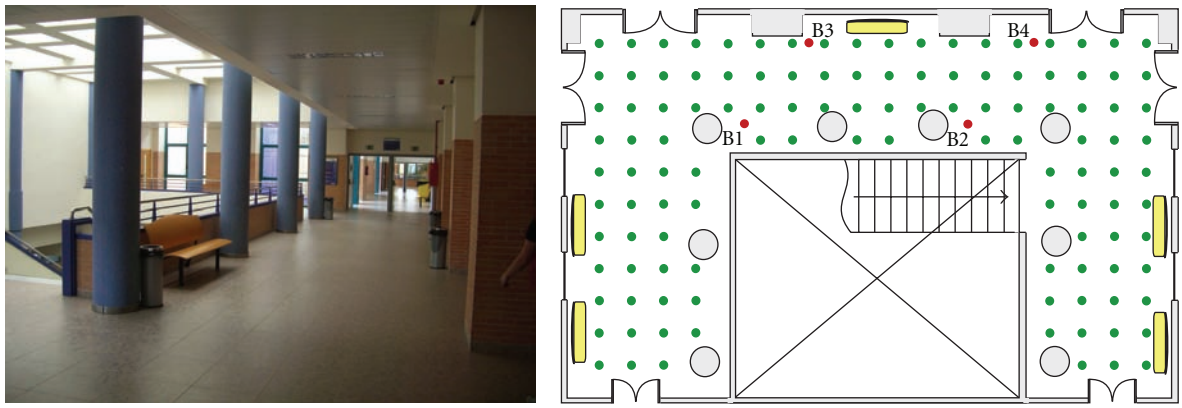


Figure 7: Scenario 3: corridor.

Table 3: Positioning error (decimeters) and calculation time (seconds). Results in *classroom* scenario.

Method	K-NN	MD	Neural Network	ARIEL Selected	ARIEL Mean
Point 1	14.14	22.36	14.32	10.00	9.85
Point 2	133.42	58.31	14.42	0.00	17.20
Point 3	0.00	36.06	27.92	0.00	20.88
⋮	⋮	⋮	⋮	⋮	⋮
Point 53	20.00	14.14	59.06	10.00	8.00
Point 54	20.00	31.62	0.00	30.00	41.23
Point 55	20.00	160.31	76.59	20.00	12.65
Err Sum	1466.93	1841.40	1448.67	727.67	926.75
Err Mean	26.67	33.48	27.07	13.23	16.85
Std Dev	31.20	35.62	23.28	14.70	15.47
Calc Time	1.7 s	0.5 s	2.4 s	0.6 s	0.6 s

will be the measurement for this power, channel and beacon combination. For every transmitter position, six different channels and four power levels are used against four beacons. This represents a total of 96 couples of values (the one measured by the beacon and the one measured by the transmitter).

The transmitter collects the measures and forwards them to the base station, who will send it to the PC through the RS232 serial port. The PC adds to each packet the transmitter actual coordinates (previously introduced by hand as reference) and generates a new entry in the signal strength database. This information will contain the transmitter characterization for every position in the scenario.

Once the whole scenario has been measured, some calculations with the received data are made in order to condense the radio map. For every set of values obtained for each location, channel, power and beacon, a mean is calculated, reducing, with this procedure, the amount of information to a fifth. This is necessary to improve the system efficiency, considering that the aim is to obtain the robot localization in real time. The calculation time is then reduced from 8 s to 1,5 s. As system performance is critical in order to obtain a valuable localization procedure, working with the whole set of samples, as would happen when applying any KNN-based algorithm, is not feasible. The ARIEL system provides this improvement, by enhancing the accuracy of the localization method and, at the same time, working with the simplified training set of radio samples.

5. Localization Estimation

Once the database is trained for a given scenario, the localization estimation procedure comes up, which consists of calculating the transmitter (mobile sensor) current position within the scenario. A mini PC in the robot stores the database and performs every calculation. Thus, the robot is completely autonomous in the localization aspect.

To accomplish this, the transmitter performs a set of measurements identical to those made in the training phase, with the corresponding channel, power, and beacon combinations. For that, the current RSS measurements set is compared with every sample stored in the database.

Several pattern recognition techniques have been compared in order to evaluate the performance of the ARIEL system.

5.1. Neural Network. Neural networks [20] have been successfully used for classification purposes (e.g., image

Table 4: Positioning error (decimeters) and calculation time (seconds). Results in *corridor* scenario.

Method	K-NN	MD	Neural Network	ARIEL Selected	ARIEL Mean
Point 1	31.62	189.74	19.66	30.00	30.08
Point 2	00.00	22.36	50.02	20.00	26.02
Point 3	00.00	22.36	14.13	31.62	26.93
⋮	⋮	⋮	⋮	⋮	⋮
Point 114	31.62	28.28	37.79	20.00	11.31
Point 115	10.00	14.14	22.49	22.36	11.40
Point 116	28.28	58.31	42.83	50.00	22.47
Err Sum	2895.58	4137.62	3657.60	1962.38	2301.44
Err Mean	24.96	35.67	31.53	16.92	19.84
Std Dev	27.76	39.46	29.84	10.38	23.45
Calc Time	3.4 s	1.3 s	3.8 s	1.5 s	1.5 s

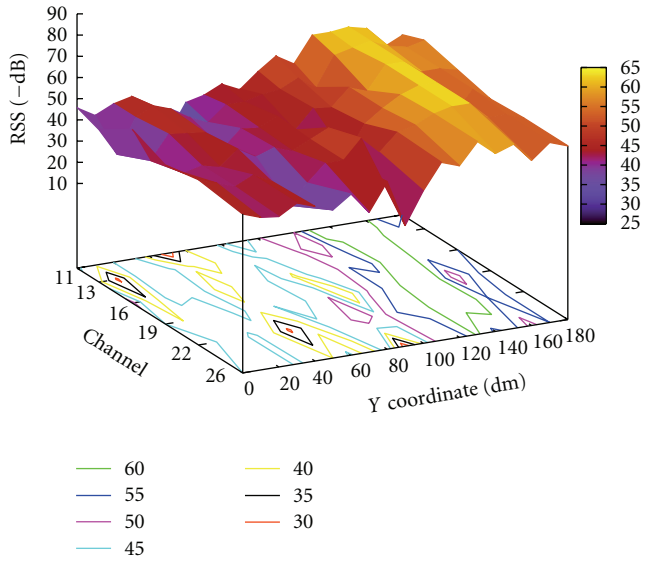


Figure 8: Signal strength distribution for different channels.

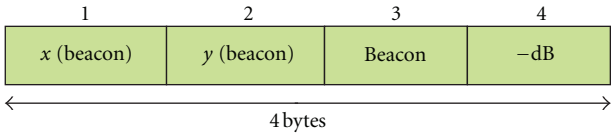


Figure 9: Beacon RSS measurement packet contents.

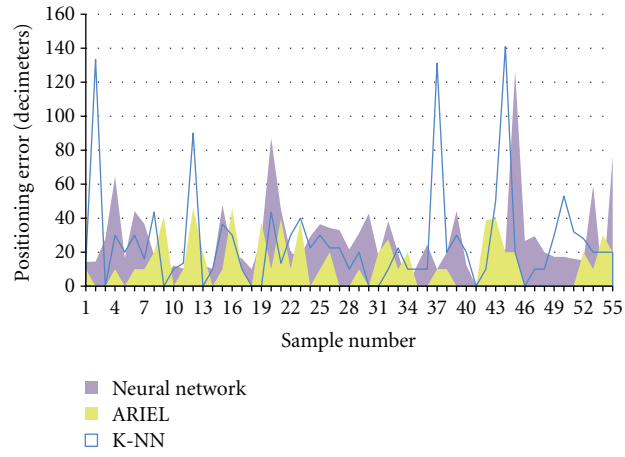


Figure 10: Results comparison between K-NN, Neural Network, and ARIEL selected methods in classroom scenario.

recognition [21, 22] or even in more sophisticated scenarios [23]).

In this paper neural networks have been used to estimate the position of the robot by taking as input the radio frequency inputs of a mobile node and the radio map that trains the network. For this, the *Resilient Backpropagation algorithm* [24] has been used, based on the results obtained in previous work [21].

In fact, the implemented neural network contains as many neurons in the last layer as available positions (i.e., 116 in the corridor scenario and 55 in the classroom scenario). Thus, each neuron will classify a given input parameter as a concrete position. For example, neuron 1 will be related with position [0, 0]. Experiments with several topologies and layers have been performed. Best results have been obtained using a 3 layers topology, with 100 neurons in the input layer and 200 in the hidden layer. Note that increasing the number of layers and number of neurons will not always lead to a performance improvement, since the error could be diminished when propagated through the network or by the creation of a local minimum; furthermore, the necessary time to converge to a solution also increments.

In the experiments, the whole set of 192 descriptors have been organized in groups of 4. Each of these subgroups represents the transmission/reception values for each beacon given a concrete configuration. Each descriptor group has been classified using a neural network with the above configuration. The position estimation has been calculated using the average of the output of each neural network for each subgroup of descriptors.

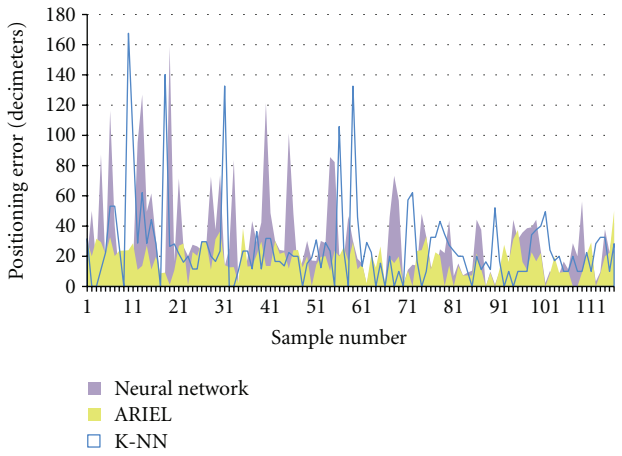


Figure 11: Results comparison between K-NN, Neural Network, and ARIEL Selected methods in corridor scenario.

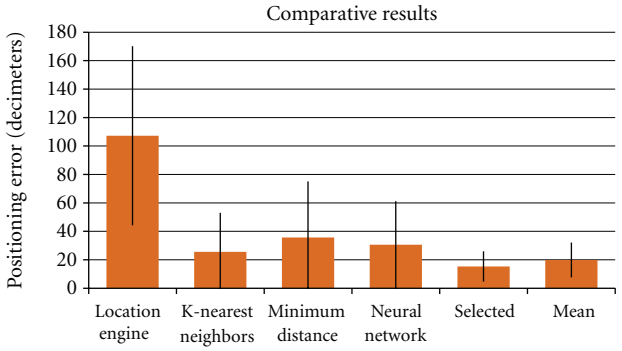


Figure 12: Results comparison between every method considered in corridor scenario.

5.2. ARIEL. The proposed method follows a similar criteria to the *k-nearest neighbors* pattern recognition method, where one calculates the k-nearest samples in the radio map that have a grater similarity to the sample obtained at the current mobile sensor position. Then, the recognition result is the more repeated position in this k-nearest vector.

Having this in mind, the following modifications have been implemented, in order to increase the whole system performance.

(i) Once we have a RSS's sample (array) for the current position, we give more weight to the RSS values received by the beacons than the one calculated from the packets received by the transmitter, since the transmitter changes its signal power and beacon does not. Then beacons will receive different values for different power while transmitter will theoretically receive every confirmation packet with the same signal strength. There are two parameters (*wfb*:weight factor for beacon and *wft*: weight factor for transmitter) to adjust this.

(ii) Two values do not need to be equal to be considered an RSS match. In fact, the parameter (*er*—equivalence radius) sets the maximum distance between two signal strength values to be considered identical.

(iii) In addition to matches, for every couple of compared values (current measurement and database stored) the difference between them is calculated and stored. This value will provide extra information for recognition since the smaller this value, the better the match.

(iv) As a result, after completing the comparison, eight candidates will be obtained (They are eight because it has been experimentally established that the correct transmitter position is between the eight best results the 95% of times) sorted by match and difference values. Depending on the matching level, the best candidate or the one with more candidate neighbors will be selected (as explained afterwards). To decide if two candidates are neighbors, the distance between them is calculated and then compared with the parameter mnd—maximum neighbor distance.

Figure 13: Visual and ZigBee positioning experiments carried out at a paraffin smoke-filled small laboratory.

Then, for every transmitter position one will go over every RSS set stored in the database and calculate the two values (*matches* and *difference*). The matches (M) value will be obtained from (3):

$$M = \sum_{b=1}^{4}\sum_{p=1}^{4}\sum_{c=1}^{6}\{\,|\,\mathrm{SSB}(b,p,c) - \mathrm{CSB}(b,p,c)\,|\,<\mathrm{er}\} * \mathrm{wfb} + \{\,|\,\mathrm{SSB}(b,p,c) - \mathrm{CST}(b,p,c)\,|\,<\mathrm{er}\} * \mathrm{wft}, \tag{3}$$

while the difference (D) value will be obtained by evaluating the equation (4):

$$D = \sum_{b=1}^{4}\sum_{p=1}^{4}\sum_{c=1}^{6}|\,\mathrm{SSB}(b,p,c) - \mathrm{CSB}(b,p,c)\,| * \mathrm{wfb} + |\,\mathrm{SST}(b,p,c) - \mathrm{CST}(b,p,c)\,| * \mathrm{wft}\,, \tag{4}$$

where b is beacon id $(1\cdots4)$, p is power id $(1\cdots4)$, c is channel id $(1\cdots6)$, SSB is stored value for signal strength received by beacon, CSB is current value for signal strength received by beacon, SST is stored value for signal strength

```
# A_DIF = Candidate A difference
# A_MATCH = Candidate A matches
# B_DIF = Candidate B difference
# B_MATCH = Candidate B matches
MST = DST = 100 # Match and Difference Soft Thresholds
MHT = DHT = 110 # Match and Difference Hard Thresholds
diff = 100 B_DIF /A_DIF
match = 100 A_MATCH / B_MATCH
# If at least one of the hard thresholds is superated
# candidate A is better than candidate B
if ((diff ≥ DHT) or (match ≥ MHT))
{
  Candidate A better than candidate B
   end
}
# If both soft thresholds are superated
# candidate A is better than candidate B
if ((diff ≥ DST) and (match ≥ MST))
{
Candidate A better than candidate B
  end
}
# If none of the previous conditions is satisfied
# candidate B is better than candidate A
Candidate B better than candidate A
end
```

Pseudocode 1: ARIEL selected method pseudocode.

received by transmitter, CST is current value for signal strength received by transmitter, er is equivalence radius, wfb is weight factor in measures received by beacon, wft is weight factor in measures received by transmitter, $A < B$: takes a "0" value if the expression is true and a "1" value if it is not

The next step consists of choosing the best candidate. From the sorted eight candidates list, if the first one (A) is much better than the second one (B), it will be considered the most probable transmitter localization. Two intermediate values are calculated to do this:

(i) cm: Candidate A matches result respect candidate B matches result: M(A)/M(B).

(ii) cd: Candidate B difference result respect candidate A difference result: D(B)/D(A).

In both cases, a higher value indicates a better result for the candidate A respect the candidate B. Four parameters are established as limits to decide:

(i) CD_HARD and CM_HARD are limit values for cd and cm. Candidate A will be selected if ONE OF THEM is overcome by the calculated value.

(ii) CD_SOFT y CM_SOFT are limit values for cd and cm. Candidate B will be selected if BOTH OF THEM are overcome by the calculated value.

In other words, if at least one of the two following conditions is accomplished, candidate A will be selected as the transmitter's nearest location.

$$(\mathrm{cd} > \mathrm{CD_{HARD}})\mathrm{OR}(\mathrm{cm} > \mathrm{CM_{HARD}}), \tag{5}$$

$$(\mathrm{cd} > \mathrm{CD_{SOFT}})\mathrm{AND}(\mathrm{cm} > \mathrm{CM_{SOFT}}). \tag{6}$$

Otherwise, a two-dimension array called *dist* (distances between candidates) will be calculated with (7):

$$\mathrm{dist}(i,j) = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}, \tag{7}$$

where i and j are the array indexes, and x_k and y_k are, respectively, the x and y coordinates of the kth candidate. From this array, one list *numneighbors* is made to store the number of neighbors of every candidate. Two candidates are considered neighbors if they are closer than *mnd*, thus the array *dist* is searched for every candidate and one neighbor that added every time a value less or equal to *mnd* is found.

Once these calculations are made, the candidate with more neighbors will be *selected* as the best result. In Pseudocode 1 the equivalent pseudocode is shown.

As an additional method, a *mean* with the selected candidate and its neighbors coordinates is provided, with an extraweight (configurable in parameter cp—central point weight) for the selected candidate.

6. Experimentation Results

In previous works, the *Location Engine* engine, integrated into the *Texas Instruments* transmitters used has been compared with the exposed methods. In open spaces, as the *garden* scenario, the results are similar with respect to the localization error, and the calculation time, as expected, is hundreds of times faster in the analytical method, so there's no point to use the empirical methods into open spaces.

The exposed methods have been used to calculate the transmitter position in the two indoor proposed scenarios (classroom and corridor). Then, distances between the actual position and the one obtained by every method (i.e., the positioning errors) have been calculated, as well as the calculation time spent on every transmitter location estimated. From this information, sum, mean, and typical deviation for every scenario and method have been calculated. All these values are shown in the Table 4 for the *classroom* scenario and in Table 3 for the *corridor* scenario. The *garden* scenario results have not been included because outdoors environment and analytical methods work quite well and are easier to implement. On the other hand, in order to appreciate the ARIEL improvement, results of K-NN and Minimum Distance (i.e., MD) original methods have been included too.

Figures 10 and 11 show in a graphical way the results obtained with the three methods in the different scenarios. The ARIEL *Selected* method provides always better results than the neural networks used and, also, a more homogeneous error distribution.

Finally, Figure 12 shows the localization error results for every method considered.

7. Conclusions and Future Work

The paper has shown a proposed fingerprint algorithm for enhancing the efficiency of localization methods in indoor environments with irregular scenarios, including different materials. The ARIEL method increases the performance of several experimented pattern recognition methods such as K-NN, Minimum Distance, and Neural Networks and shows good results in every tested scenario. Combined with a designed high luminosity visual localization panel, the system may allow a robot to navigate in a smoky atmosphere and reach specific points of interest to help a firemen. Due to the complexity of filling with smoke the explored scenarios, some measurements have carried out in a small laboratory filled with paraffin smoke, as shown in Figure 13, showing no significant reduction in the precision. Further works will use real fire smoke.

It is necessary to consider that the neural network method requires a previous training phase for every given scenario and more hardware resources in the sensor nodes in order to perform the calculations. Future work will be focused on determining which measures give the most important information to the fingerprinting pattern recognition method, in order to reduce the amount of measurements involved, improving the calculation time and allowing the ARIEL method to be implemented with simpler hardware devices. In the neural network aspect, more strategies need to be used in order to improve the recognition efficiency.

In addition, only the localization phase has been considered. In the navigation phase, once the position of the robot is reasonably known, only near positions will be searched in the database, both reducing the calculation time and the probability of significant errors in distance estimation.

Acknowledgments

This work has been funded in part by the EU-VI Framework Programme under Grant IST-045269-GUARDIANS of the EC Cognitive Systems initiative, the Bancaja-UJI research program under Grant RETA(P1-1B209-39), the Spanish National CICYT under Grant TIN2009-14475-C04, European Commission Seventh Framework Programme FP7/2007-2013 under Grant agreement 248497 (TRIDENT Project), by Spanish Ministry of Research and Innovation DPI2008-06548-C03 (RAUVI Project), and by Fundació Caixa Castelló-Bancaixa P1-1B2009-50.

References

[1] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.

[2] S. Capkun, M. Hamdi, and J. Hubaux, "GPS-free positioning in mobile ad-hoc networks," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, p. 10, January 2001.

[3] G. Cen, N. Matsuhira, J. Hirokawa, H. Ogawa, and I. Hagiwara, "Mobile robot global localization using particle filters," in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '08)*, pp. 710–713, October 2008.

[4] M. Hentschel, O. Wulf, and B. Wagner, "A GPS and laser-based localization for urban and non-urban outdoor environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08)*, pp. 149–154, September 2008.

[5] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.

[6] S. Atiya and G. D. Hager, "Real-time vision-based robot localization," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 6, pp. 785–800, 1993.

[7] D. Niculescu and B. Nath, "Ad Hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, vol. 3, pp. 1734–1743, April 2003.

[8] Z. Shan and T. S. P. Yum, "Precise localization with smart antennas in ad-hoc networks," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 1053–1057, November 2007.

[9] G. Teng, K. Zheng, and G. Yu, "A mobile-beacon-assisted sensor network localization based on RSS and connectivity observations," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 487209, 14 pages, 2011.

[10] J. Sales, M. El-Habbal, R. Marin et al., "Localization of networked mobile sensors and actuators in low-visibility conditions," in *RISE (IARP/EURON Workshop on Robotics for*

Risky Interventions and Environmental Surveillance), Hallam University, Sheffield, UK, January 2010.

[11] J. Sales, R. Maŕn, E. Cervera, S. Rodríguez, and J. Pérez, "Multi-sensor person following in low-visibility scenarios," *Sensors*, vol. 10, no. 12, pp. 10953–10966, 2010, http://www.mdpi.com/1424-8220/10/12/10953/.

[12] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '00)*, pp. 775–784, March 2000.

[13] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, pp. 205–218, June 2005.

[14] Q. Yao, F. Y. Wang, H. Gao, K. Wang, and H. Zhao, "Location estimation in ZigBee network based on fingerprinting," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES '07)*, pp. 1–6, December 2007.

[15] "GUARDIANS EU project (IST-045269) (Group of Unmanned Assistant Robots Deployed in Aggregative Navigation supported by Scent Detection)," http://www.shu.ac.uk/research/meri/research/guardians-project.

[16] J. Marti and R. Marin, "ARIEL: advanced radiofrequency indoor environment localization: Smoke conditions positioning," in *Proceedings of the 3rd International Workshop on Performance Control in Wireless Sensors Networks (PWSN '11)*, Barcelona, Spain, June 2011.

[17] Y. Wang, D. Wu, S. Seifzadeh, and J. Chen, "A moving grid cell based MCL algorithm for mobile robot localization," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '09)*, pp. 2445–2450, December 2009.

[18] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[19] J. V. M. Aviles and R. M. Prades, "Pattern recognition comparative analysis applied to fingerprint indoor mobile sensors localization," in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology (CIT '10)*, pp. 730–736, Bradford, UK, July 2010.

[20] G. P. Zhang, "Neural networks for classification: A survey," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 30, no. 4, pp. 451–462, 2000.

[21] E. Jiménez, R. Marín, and P. J. Sanz, "A soft computing classifier based on fourier descriptors within online robots context," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 4838–4843, October 2004.

[22] S. A. Nazeer, N. Omar, and M. Khalid, "Face recognition system using artificial neural networks approach," in *Proceedings of the International Conference on Signal Processing, Communications and Networking (ICSCN '07)*, pp. 420–425, February 2007.

[23] L. Shi, Z. Wang, L. Wang, and J. Zhang, "The aide diagnosis of cardiac heart diseases using a deoxyribonucleic acid based backpropagation neural network," *International Journal of Distributed Sensor Networks*, vol. 5, no. 1, pp. 38–38, 2009.

[24] M. Riedmiller and H. Braun, "Direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 586–591, April 1993.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 523787, 14 pages
doi:10.1155/2012/523787

Research Article

Robust Maximum Lifetime Routing and Energy Allocation in Wireless Sensor Networks

Ioannis Ch. Paschalidis[1] and Ruomin Wu[2]

[1] *Division of Systems Engineering, Department of Electrical and Computer Engineering, Center for Information and Systems Engineering, Boston University, 8 St. Mary's Street, Boston, MA 02215, USA*
[2] *Center for Information and Systems Engineering, Boston University, Boston, MA 02215, USA*

Correspondence should be addressed to Ioannis Ch. Paschalidis, yannisp@bu.edu

Received 28 November 2011; Accepted 10 May 2012

Academic Editor: Pedro Ruiz

We consider the maximum lifetime routing problem in wireless sensor networks in two settings: (a) when nodes' initial energy is given and (b) when it is subject to optimization. The optimal solution and objective value provide optimal flows and the corresponding predicted lifetime, respectively. We stipulate that there is uncertainty in various network parameters (available energy and energy depletion rates). In setting (a) we show that for specific, yet typical, network topologies, the actual network lifetime will reach the predicted value with a probability that converges to zero as the number of nodes grows large. In setting (b) the same result holds for all topologies. We develop a series of robust problem formulations, ranging from pessimistic to optimistic. A set of parameters enable the tuning of the conservatism of the formulation to obtain network flows with a desirably high probability that the corresponding lifetime prediction is achieved. We establish a number of properties for the robust network flows and energy allocations and provide numerical results to highlight the tradeoff between predicted lifetime and the probability achieved. Further, we analyze an interesting limiting regime of massively deployed sensor networks and essentially solve a continuous version of the problem.

1. Introduction

Wireless sensor networks (WSNETs) have emerged as an exciting new paradigm of inexpensive, easily deployable, completely untethered device networks that enable the automated and intelligent monitoring and control of physical systems. WSNET nodes can be equipped with a variety of sensors, have a built-in radio to communicate with each other, are powered by batteries, and have limited information storage and processing capabilities. WSNETs can be useful in a plethora of applications including industrial and building automation, health monitoring, wildlife monitoring, and asset and personnel tracking [1]. Battery technology, however, remains a critical bottleneck. In many applications one would like to use the WSNET for long periods, often years, without changing batteries. As a result, energy conservation is a primary concern and aggressive optimization becomes indispensable.

In this paper, we focus on the problem of selecting an optimal strategy for routing packets from data-collecting sensor nodes to a set of gateways (or sinks) in order to minimize the rate at which energy is consumed or, equivalently, to maximize the lifetime of the network. We consider two situations: (i) when the initial energy of every node is given and (ii) when it is also subject to optimization given an overall energy budget. Routing, of course, has received quite a bit of attention in WSNETs. Various aspects of the problem have been considered in [2–11], which mostly focus on finding a single path from origin to destination. A more static view is adopted in [12], followed by [13], and [14], which provide a linear programming formulation for optimizing average flows between nodes.

Our starting point is the flow optimizing formulation of [12, 14]. A different but equivalent formulation using optimal control ideas is in [15]. Key data to solve this problem include the total available energy at the nodes and

the energy consumption rates. These quantities are hardly known with any degree of certainty or accuracy. Yet, they affect both the optimal flows and the corresponding optimal objective value, that is, the predicted network lifetime. The latter value will in fact be equal to the actual network lifetime if all problem data are known with certainty. We note that both these quantities are quite important for the network designer. The predicted network lifetime is useful for planning purposes, and the optimal flows indicate how routing should be done to achieve such a lifetime.

Uncertainty, though, renders the predicted lifetime overly optimistic. For the case without energy allocation, we show that for specific, yet typical, topologies including linear and two-dimensional grid-like networks, the actual lifetime will reach the predicted value with a probability that converges to zero as the number of nodes grows large. This suggests that the predicted network lifetime is not a particularly useful estimate under uncertainty.

For the energy allocation case, we show the same result without any topological assumptions. We also find that uncertainty impacts the optimal policy as well, and one needs to use a different set of "robust" flows to protect against uncertainty. To that end, we develop a series of alternative robust problem formulations, ranging from pessimistic to optimistic. A set of parameters enable the tuning of the conservatism of the formulation with a desirably high probability that the corresponding lifetime prediction will be achieved—a lifetime guarantee probability. Our robust formulations are based on recent work in robust linear programming in [16, 17]. However, the problem we consider has special structure which we exploit to establish a number of interesting properties. Robust optimization has in general received a lot of attention lately and has found applications in many areas. It started with [18] with more recent contributions in [16, 19].

To gain more insight, we consider maximum lifetime routing with energy allocation in a continuous setting of massively dense WSNETs. Related limiting regimes have previously been considered in [8, 20, 21]. For a single point source and a single point sink, we show that the optimal route is a straight line from the source to the sink. For multiple sources and sinks, we show that sources send their flows to the closest sink, again over a straight line.

The rest of the paper is organized as follows. In Section 2, we tackle the maximum lifetime routing problem without energy allocation, introducing robust formulations and characterizing their solutions. Section 3 incorporates the energy allocation into the problem. In Section 4, we develop the continuous version of the problem with energy allocation. Numerical examples are in Section 5. Conclusions are in Section 6.

2. Maximum Lifetime Routing without Node Energy Allocation

We represent a WSNET as a directed graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the node set and $\mathcal{A}$ is the set of directed links (i, j) with $i, j \in \mathcal{N}$. Link (i, j) exists if and only if $j \in \mathcal{S}_i$, where $\mathcal{S}_i$ is the set of nodes that can be reached by i. Each node i has an initial battery energy of E_i and consumes e_{ij}^t per data unit to transmit to j, while j consumes e_{ij}^r per data unit to receive from i. We assume that the nodes are able to relay packets and to adjust the transmit power level to the minimum required in order to reach the intended receiver. Origin nodes (or sources) $\mathcal{O}$ include all $i \in \mathcal{N}$ with a positive (constant) information generation rate Q_i. $\mathcal{D}$ is the set of sink nodes (or sinks) responsible for collecting all data. Assume $\mathcal{O} \cap \mathcal{D} = \emptyset$; we refer to nodes in $\mathcal{N} \setminus \mathcal{D}$ simply as sensor nodes.

Every source node seeks to send its data to one of the sinks, not necessarily the same one for each data unit generated. To that end, node i may use multiple other nodes as relays. Let q_{ij} be the information transmission rate from i to j. We write $\mathbf{q}$ for the vector of all q_{ij}'s. (We use bold letters to denote vectors and all vectors are assumed to be column vectors unless explicitly stated otherwise.) Note that routing and power control are intrinsically coupled since the power level is adjusted depending on the choice of the next hop.

In the sequel, we only consider the energy spent for communications since this is the dominant energy consumption term in WSNETs (see [22]). Additional energy consumption terms could be incorporated into e_{ij}^t, e_{ij}^r. For example, a sensing/processing energy cost at transmissions or receptions per data unit can be incorporated into e_{ij}^t and e_{ij}^r. We also assume that e_{ij}^t is monotonically increasing with the distance between i and j. Finally, sink nodes are assumed to be powered by line power.

The lifetime of a sensor node i under a given set of flows $\mathbf{q}$ is given by

$$T_i(\mathbf{q}) = \frac{E_i}{\sum_{j \in \mathcal{S}_i} e_{ij}^t q_{ij} + \sum_{j \mid i \in \mathcal{S}_j} e_{ji}^r q_{ji}}, \quad \forall i \in \mathcal{N} \setminus \mathcal{D}. \tag{1}$$

Define the network lifetime under flow $\mathbf{q}$ as the minimum lifetime over all nodes, that is,

$$T_{\text{net}}(\mathbf{q}) \triangleq \min_{i \in \mathcal{N} \setminus \mathcal{D}} T_i(\mathbf{q}). \tag{2}$$

The network lifetime is equivalent to the earliest time, a sensor node runs out of energy.

2.1. Problem Formulations. The maximum lifetime routing problem without node energy allocation is the problem of selecting flows $\mathbf{q}$ to maximize $T_{\text{net}}(\mathbf{q})$. Letting $\hat{q}_{ij} = q_{ij} T$ denote the amount of information transmitted from i to j over the lifetime T, [12] formulated the problem as a linear program as follows:

$$\max \quad T, \tag{3}$$

$$\text{s.t.} \quad \sum_{j \mid i \in \mathcal{S}_j} \hat{q}_{ji} + Q_i T = \sum_{j \in \mathcal{S}_i} \hat{q}_{ij}, \quad \forall i \in \mathcal{N} \setminus \mathcal{D}, \tag{4}$$

$$\sum_{j \in \mathcal{S}_i} e_{ij}^t \hat{q}_{ij} + \sum_{j \mid i \in \mathcal{S}_j} e_{ji}^r \hat{q}_{ji} \le E_i, \quad \forall i \in \mathcal{N} \setminus \mathcal{D}, \tag{5}$$

$$T \ge 0, \quad \hat{q}_{ij} \ge 0, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{S}_i, \tag{6}$$

where the decision variables are T and the $\hat{q}_{ij}$'s. On a notational remark, we will use $\hat{\mathbf{q}}$ to denote flow over the

lifetime T and $\mathbf{q}$ to denote flow per unit of time. Thus, when we refer to an optimal solution $\mathbf{q}^*$ (resp. $\hat{\mathbf{q}}^*$) of (3) we mean optimal flow per unit of time (resp. over the lifetime). The first set of constraints correspond to flow conservation and the second set of constraints follows from the definition of lifetime. We note that this formulation can also account for the energy consumed while the node's radio is listening. Specifically, we can add $e_i^{\mathrm{ON}}\lambda_i T$ to the lefthand side of (5), where e_i^{ON} is the energy consumption rate by the radio while listening, λ_i is the fraction of time node and i is "awake" and listening. We refer to (3) as the nominal problem. Note that it is always feasible if for every sensor node there exists a path to a sink node. We assume that this will always be the case. We note that problem (3) can be solved in a distributed manner using subgradient optimization techniques for the dual [23]. This is appealing for WSNET applications. Here, however, we concentrate on the impact of uncertainty and do not focus on distributed solution approaches. It can be also argued that in several application contexts a distributed approach is not critical since (3) is solved during a planning/deployment stage of the WSNET.

The data for the nominal problem are e_{ij}^t, e_{ij}^r, and E_i and these affect both the optimal solution and the optimal value. As these may be uncertain, we model them as symmetrically bounded nonnegative random variables (r.v.'s) with ranges given by: $e_{ij}^t \in [\overline{e}_{ij}^t - \Delta e_{ij}^t, \overline{e}_{ij}^t + \Delta e_{ij}^t]$, $e_{ij}^r \in [\overline{e}_{ij}^r - \Delta e_{ij}^r, \overline{e}_{ij}^r + \Delta e_{ij}^r]$, and $E_i \in [\overline{E}_i - \Delta E_i, \overline{E}_i + \Delta E_i]$. We will call $\overline{e}_{ij}^t$, $\overline{e}_{ij}^r$, and $\overline{E}_i$ the nominal values and assume that they are the means of the corresponding r.v.'s. The values Δe_{ij}^t, Δe_{ij}^r, and ΔE_i represent the maximum deviations from the mean which are assumed to be identical left and right from the mean (hence, the term symmetrically bounded r.v.'s). These deviations are defined so that all r.v.'s have positive support. We also define the uncertainty *sets* $J_i^t \triangleq \{j \mid \Delta e_{ij}^t > 0,\ j \in \mathcal{S}_i\}$ and $J_i^r \triangleq \{j \mid \Delta e_{ji}^r > 0,\ i \in \mathcal{S}_j\}$, for all $i \in \mathcal{N} \setminus \mathcal{D}$.

Due to data uncertainty, the optimal solution of (3) may not be feasible. It can be easily seen that the following worst-case formulation guarantees feasibility for any realization of the following data:

$$\begin{aligned}
&\max \quad T, && (7)\\
&\text{s.t.} \quad (4),\ (6),\\
&\sum_{j\in\mathcal{S}_i} \overline{e}_{ij}^t \hat{q}_{ij} + \sum_{j\in J_i^t} \Delta e_{ij}^t \hat{q}_{ij} + \sum_{j\mid i\in\mathcal{S}_j} \overline{e}_{ji}^r \hat{q}_{ji} && (8)\\
&\quad + \sum_{j\in J_i^r} \Delta e_{ji}^r \hat{q}_{ji} \le \overline{E}_i - \Delta E_i,\ \forall i \in \mathcal{N} \setminus \mathcal{D}.
\end{aligned}$$

We refer to the above as the fat problem. By construction, its optimal solution is feasible for any data realization but it may be overly conservative. Intuitively, the probability that all parameters take their "extreme" value should be small, thus, motivating a less conservative formulation.

We introduce the uncertainty budget $\Gamma_i^e \in [0, |J_i^t| + |J_i^r|]$ for every sensor node i and define the restricted uncertainty set $\mathcal{R}_i(\Gamma_i^e)$ as

$$\begin{aligned}
\mathcal{R}_i(\Gamma_i^e) = \Bigg\{ & e_{ij}^t, e_{ji}^r \mid e_{ij}^t \in \left[\overline{e}_{ij}^t - \Delta e_{ij}^t, \overline{e}_{ij}^t + \Delta e_{ij}^t\right],\\
& e_{ij}^r \in \left[\overline{e}_{ij}^r - \Delta e_{ij}^r, \overline{e}_{ij}^r + \Delta e_{ij}^r\right], \qquad (9)\\
& \sum_{j\in J_i^t} \frac{\left|e_{ij}^t - \overline{e}_{ij}^t\right|}{\Delta e_{ij}^t} + \sum_{j\in J_i^r} \frac{\left|e_{ji}^r - \overline{e}_{ji}^r\right|}{\Delta e_{ji}^r} \le \Gamma_i^e \Bigg\}.
\end{aligned}$$

We view the uncertainty budget as an ℓ_1-norm constraint for the vector

$$\left(\left(\frac{e_{ij}^t - \overline{e}_{ij}^t}{\Delta e_{ij}^t}\right)_{j\in J_i^t}, \left(\frac{e_{ji}^r - \overline{e}_{ji}^r}{\Delta e_{ji}^r}\right)_{j\in J_i^r}\right). \qquad (10)$$

Similarly, let $\Gamma_i^E \in [0,1]$ be the uncertainty budget for E_i, namely, $E_i \in [\overline{E}_i - \Gamma_i^E \Delta E_i, \overline{E}_i + \Gamma_i^E \Delta E_i]$. The following robust maximum lifetime routing problem is formulated so that we can guarantee feasibility for all data realizations in the following restricted uncertainty sets:

$$\begin{aligned}
&\max \quad T, && (11)\\
&\text{s.t.} \quad (4),\ (6),\\
&\max_{e_{ij}^t, e_{ji}^r \in \mathcal{R}_i(\Gamma_i^e)} \left\{ \sum_{j\in\mathcal{S}_i} e_{ij}^t \hat{q}_{ij} + \sum_{j\mid i\in\mathcal{S}_j} e_{ji}^r \hat{q}_{ji} \right\} && (12)\\
&\quad \le \overline{E}_i - \Gamma_i^E \Delta E_i,\ \forall i \in \mathcal{N} \setminus \mathcal{D}.
\end{aligned}$$

In the Appendix, we show that the above is equivalent to a linear programming problem.

Theorem 1. *The robust problem* (11) *is equivalent to the linear following programming formulation:*

$$\begin{aligned}
&\max \quad T\\
&\text{s.t.} \quad \sum_{j\in\mathcal{S}_i} \overline{e}_{ij}^t \hat{q}_{ij} + \sum_{j\mid i\in\mathcal{S}_j} \overline{e}_{ji}^r \hat{q}_{ji} + \sum_{j\in J_i^t} \omega_{ij} + \sum_{j\in J_i^r} \nu_{ji}\\
&\quad + \Gamma_i^e p_i \le \overline{E}_i - \Gamma_i^E \Delta E_i,\ \forall i \in \mathcal{N} \setminus \mathcal{D},\\
&\quad (4),\ (6),\ \ p_i \ge 0,\ \forall i \in \mathcal{N} \setminus \mathcal{D},\\
&\quad p_i + \omega_{ij} \ge \Delta e_{ij}^t \hat{q}_{ij}, \quad \omega_{ij} \ge 0,\ \forall i \in \mathcal{N} \setminus \mathcal{D},\ j \in J_i^t,\\
&\quad p_i + \nu_{ji} \ge \Delta e_{ji}^r \hat{q}_{ji}, \quad \nu_{ji} \ge 0,\ \forall i \in \mathcal{N} \setminus \mathcal{D},\ j \in J_i^r. \qquad (13)
\end{aligned}$$

Furthermore, solving (13) *one obtains an optimal solution* $(\hat{\mathbf{q}}^R, T^R, \mathbf{p}^R, \omega^R, \nu^R)$ *so that* $(\hat{\mathbf{q}}^R, T^R)$ *is feasible for* (11) *and* T^R *is equal to the optimal value of* (11).

2.2. Properties of Optimal Solutions. Next, we study the relationships between the three formulations and establish properties of the optimal solutions. We also introduce a metric—the lifetime guarantee probability—to quantify how likely it is for the predicted lifetime to be achieved.

2.2.1. Optimal Lifetime. Let T_N^*, T_F^*, T_R^* denote the optimal values of the nominal, fat, and robust problems, respectively. Let $\mathbf{\Gamma}^e = (\Gamma_1^e, \ldots, \Gamma_{|\mathcal{N}\setminus\mathcal{D}|}^e)$ and $\mathbf{\Gamma}^E = (\Gamma_1^E, \ldots, \Gamma_{|\mathcal{N}\setminus\mathcal{D}|}^E)$. Note that T_R^* depends on $\mathbf{\Gamma}^e$ and $\mathbf{\Gamma}^E$. To express this dependence, we write $T_R^*(\mathbf{\Gamma}^e, \mathbf{\Gamma}^E)$. The following proposition is almost immediate. It simply states that by adjusting the uncertainty budgets one can generate a continuum of formulations whose predicted lifetime ranges from the fat to the nominal.

Proposition 2. *$T_R^*(\mathbf{\Gamma}^e, \mathbf{\Gamma}^E)$ is a nonincreasing function of both $\mathbf{\Gamma}^e$ and $\mathbf{\Gamma}^E$. Furthermore, $T_F^* \le T_R^*(\mathbf{\Gamma}^e, \mathbf{\Gamma}^E) \le T_N^*$.*

Proof. Fix $\mathbf{\Gamma}^{e1}, \mathbf{\Gamma}^{e2}, \mathbf{\Gamma}^{E1}, \mathbf{\Gamma}^{E2}$ so that $\mathbf{\Gamma}^{e1} \le \mathbf{\Gamma}^{e2}$, and $\mathbf{\Gamma}^{E1} \le \mathbf{\Gamma}^{E2}$. It follows that $\mathcal{R}_i(\Gamma_i^{e1}) \subseteq \mathcal{R}_i(\Gamma_i^{e2})$, for all $i \in \mathcal{N}\setminus\mathcal{D}$. Let $\hat{\mathbf{q}}^2$ be an optimal flow for the robust routing problem under $\mathbf{\Gamma}^{e2}, \mathbf{\Gamma}^{E2}$. For all $i \in \mathcal{N}\setminus\mathcal{D}$, we have

$$\begin{aligned}
&\max_{e_{ij}^t, e_{ji}^r \in \mathcal{R}_i(\Gamma_i^{e1})} \left\{ \sum_{j\in\mathcal{S}_i} e_{ij}^t \hat{q}_{ij}^2 + \sum_{j|i\in\mathcal{S}_j} e_{ji}^r \hat{q}_{ji}^2 \right\} \\
&\quad \le \max_{e_{ij}^t, e_{ji}^r \in \mathcal{R}_i(\Gamma_i^{e2})} \left\{ \sum_{j\in\mathcal{S}_i} e_{ij}^t \hat{q}_{ij}^2 + \sum_{j|i\in\mathcal{S}_j} e_{ji}^r \hat{q}_{ji}^2 \right\} \\
&\quad \le \overline{E}_i - \Gamma_i^{E2}\Delta E_i \le \overline{E}_i - \Gamma_i^{E1}\Delta E_i,
\end{aligned} \tag{14}$$

which suggests that $\hat{\mathbf{q}}^2$ is a feasible flow vector for the robust routing problem under $\mathbf{\Gamma}^{e1}, \mathbf{\Gamma}^{E1}$. It follows that $T_R^*(\mathbf{\Gamma}^{e1}, \mathbf{\Gamma}^{E1}) \ge T_R^*(\mathbf{\Gamma}^{e2}, \mathbf{\Gamma}^{E2})$.

Next notice that when $\mathbf{\Gamma}^e = \mathbf{0}$, $\mathbf{\Gamma}^E = \mathbf{0}$, the uncertainty set becomes $\mathcal{R}_i(\Gamma_i^e) = \{e_{ij}^t, e_{ji}^r \mid e_{ij}^t = \overline{e}_{ij}^t, e_{ji}^r = \overline{e}_{ji}^r\}$ and the robust routing problem (11) reduces to the nominal routing problem (3), that is, $T_R^*(\mathbf{0}, \mathbf{0}) = T_N^*$.

When $\mathbf{\Gamma}^e = (|J_1^t|+|J_1^r|, \ldots, |J_{|\mathcal{N}\setminus\mathcal{D}|}^t|+|J_{|\mathcal{N}\setminus\mathcal{D}|}^r|)$ and $\Gamma_i^E = 1$ for all i the uncertainty sets becomes $\mathcal{R}_i(\Gamma_i^e) = \{e_{ij}^t, e_{ji}^r \mid e_{ij}^t \in [\overline{e}_{ij}^t - \Delta e_{ij}^t, \overline{e}_{ij}^t + \Delta e_{ij}^t], e_{ij}^r \in [\overline{e}_{ij}^r - \Delta e_{ij}^r, \overline{e}_{ij}^r + \Delta e_{ij}^r]\}$ and $E_i \in [\overline{E}_i - \Delta E_i, \overline{E}_i + \Delta E_i]$ for all i, which implies that the robust routing problem (11) reduces to the fat one (7). □

Standard sensitivity analysis results from linear programming yield the following corollary.

Corollary 3. *$T_R^*(\mathbf{\Gamma}^e, \mathbf{\Gamma}^E)$ is a concave function of $\mathbf{\Gamma}^E$.*

Observe now that at optimality at least one of the energy constraints (5), (8), and (12) will be active. This is stated in the following proposition. We will call *dead* the nodes that correspond to active constraints at optimality. The lifetime of a dead node equals the lifetime of the network.

Proposition 4. *At optimality, at least one of the energy constraints in each of the nominal* (3)*, fat* (7)*, and robust* (11) *formulations will be active.*

2.2.2. Optimal Flows. Consider an optimal flow vector $\hat{\mathbf{q}}$ obtained by solving one of the three formulations. Recall that $\hat{\mathbf{q}}$ denotes total flow over the lifetime and $\mathbf{q}$ flow per unit of time. We associate a directed graph (subgraph of $\mathcal{G}$) $\mathcal{G}_{\mathbf{q}} = (\mathcal{N}, \mathcal{A}_{\mathbf{q}})$ to $\mathbf{q}$, where $\mathcal{A}_{\mathbf{q}}$ contains all (i, j) with $q_{ij} > 0$. We say that a flow $\mathbf{q}$ is acyclic (resp., cyclic) if $\mathcal{G}_{\mathbf{q}}$ contains no cycles (resp., otherwise).

Theorem 5. *For all three formulations* (3)*,* (7)*, and* (11)*, there exist acyclic optimal flows.*

Proof. Let $(\mathbf{q}^*, T^*)$ be an optimal solution with $q_{i_1i_2}^*, q_{i_2i_3}^*, q_{i_3i_4}^*, \ldots, q_{i_ki_1}^*$ forming a cycle in $\mathcal{G}_{\mathbf{q}^*}$. Let $\Delta q = \min\{q_{i_1i_2}^*, q_{i_2i_3}^*, \ldots, q_{i_ki_1}^*\}$. Subtract Δq from all the flows on the cycle. At least one of $q_{i_1i_2}^*, q_{i_2i_3}^*, q_{i_3i_4}^*, \ldots, q_{i_ki_1}^*$ becomes zero and all other flows remain nonnegative. Because both the inflow and outflow at each node is reduced by the same amount, the flow conservation condition for all the nodes $i_1, \ldots, i_k$ still holds. Since the above operation only reduces flows, all the energy constraints remain satisfied. Hence, the reduced flows remain optimal. We can repeat the same process to eliminate any other cycle. □

Since (i, j, i) is a trivial cycle, we obtain the following corollary.

Corollary 6. *For all three routing formulations* (3)*,* (7)*, and* (11)*, there exists an optimal flow $\mathbf{q}$ which satisfies $q_{ij}q_{ji} = 0$ for all possible links (i, j) and (j, i).*

Corollary 7. *For all three routing formulations* (3)*,* (7)*, and* (11)*, there exists an optimal flow $\mathbf{q}$ satisfying $q_{ij} = 0$, for all $i \in \mathcal{D}$, which means no flow out of sinks.*

Proof. Let $\mathbf{q}^*$ be an acyclic optimal flow (cf. Theorem 5). Suppose there are sinks with positive flows emanating from them. Let $i \in \mathcal{D}$ such that $S_i' \triangleq \{k \mid q_{ik}^* > 0\}$ is not empty. For $j \in S_i'$ let $S_j' = \{k \mid q_{jk}^* > 0\}$. We reduce q_{ij}^* to zero by proportionally allocating this flow reduction to all outflows from node j. To be specific, for all $k_0 \in S_j'$ we set the new reduced flow as $\tilde{q}_{jk_0}^* := q_{jk_0}^* - q_{ij}^*(q_{jk_0}^*/(\sum_{k\in S_j'} q_{jk}^*))$ which maintains the nonnegativity of the resulting flow. The flow reduction $q_{ij}^*(q_{jk_0}^*/(\sum_{k\in S_j'} q_{jk}^*))$ can be propagated to the node downstream from j in a similar way. Since $\mathbf{q}^*$ is acyclic and the network is finite, propagating the flow reduction as described above terminates at some other sink nodes. During this process, flow conservation and energy constraints are maintained. This yields a new optimal flow vector with no flows out of sinks. □

2.2.3. Lifetime Guarantee Probability. Consider one of the three formulations (3), (7), and (11) and let $\mathbf{q}^*, T^*$ be an optimal solution. We will refer to the probability

$$\mathbf{P}\left[\min_{i\in\mathcal{N}\setminus\mathcal{D}} \frac{E_i}{\sum_{j\in\mathcal{S}_i} e_{ij}^t q_{ij}^* + \sum_{j|i\in\mathcal{S}_j} e_{ji}^r q_{ji}^*} \ge T^*\right], \tag{15}$$

evaluated under the distributions of the r.v.'s E_i, e_{ij}^t, e_{ji}^r, as the lifetime guarantee probability. This is the probability that

the actual lifetime obtained by applying the optimal flow $\mathbf{q}^*$ achieves the predicted optimal lifetime. We denote by P_N, P_F, P_R the lifetime guarantee probabilities for the nominal (3), fat (7), and robust (11) formulations, respectively. By design, the fat formulation provides an "absolute" guarantee; we omit the proof.

Theorem 8. *It holds that* $P_F = 1$.

The straightforward observation is that when $\Gamma_i^e \to |J_i^t| + |J_i^r|$, $\Gamma_i^E \to 1$, for all $i \in \mathcal{N} \setminus \mathcal{D}$, then $P_R \to P_F$; while when $\Gamma_i^e \to 0$, $\Gamma_i^E \to 0$, for all $i \in \mathcal{N} \setminus \mathcal{D}$, then $P_R \to P_N$.

Now let $\mathcal{A}^N$ be the set of nodes having active energy constraints at optimality in the nominal formulation (3). For any random variable a with mean $\overline{a}$ and support in $[\overline{a} - \Delta a, \overline{a} + \Delta a]$, we say that it is *symmetrically distributed* if $F_a(\overline{a} - \delta) = 1 - F_a(\overline{a} + \delta)$ for all $\delta \in [0, \Delta a]$, where F_a is the cumulative distribution function of a.

Theorem 9. *If* E_i, e_{ij}^t, e_{ji}^r *are independent symmetrically distributed r.v.'s, then* $P_N \leq (1/2)^{|\mathcal{A}^N|}$.

Proof. Let $(\mathbf{q}^{*N}, T_N^*)$ be an optimal solution to the nominal problem (3). We have

$$P_N \leq \mathbf{P}\left[E_i \geq \sum_{j \in \mathcal{S}_i} e_{ij}^t \hat{q}_{ij}^{*N} + \sum_{j \mid i \in \mathcal{S}_j} e_{ji}^r \hat{q}_{ji}^{*N},\ \forall i \in \mathcal{A}^N\right]. \quad (16)$$

For $i \in \mathcal{A}^N$ and because $\mathbf{q}^{*N}$ is feasible for the nominal problem it holds $\overline{E}_i = \sum_{j \in \mathcal{S}_i} \overline{e}_{ij}^t \hat{q}_{ij}^{*N} + \sum_{j \mid i \in \mathcal{S}_j} \overline{e}_{ji}^r \hat{q}_{ji}^{*N}$. Since $E_i, e_{ij}^t \hat{q}_{ij}^{*N}, e_{ji}^r \hat{q}_{ji}^{*N}$ are independent symmetrically distributed r.v.'s with means $\overline{E}_i, \overline{e}_{ij}^t \hat{q}_{ij}^{*N}, \overline{e}_{ji}^r \hat{q}_{ji}^{*N}$, respectively, it follows that

$$\mathbf{P}\left[E_i \geq \sum_{j \in \mathcal{S}_i} e_{ij}^t \hat{q}_{ij}^{*N} + \sum_{j \mid i \in \mathcal{S}_j} e_{ji}^r \hat{q}_{ji}^{*N}\right] = \frac{1}{2}. \quad (17)$$

By independence, we have $P_N \leq (1/2)^{|\mathcal{A}^N|}$. □

2.3. Linear and Square Arrays. In this section, we study two regular network topologies: linear and square arrays. Linear arrays appear, for instance, in pipeline monitoring applications and square arrays are applicable in environmental monitoring applications.

2.3.1. Linear Arrays. We consider a linear array segment where one sink node is at the center and an equal number k of sensor nodes are aligned one by one on both sides of the sink. The distance between neighboring nodes is d. The radio range is in $[2d, 3d)$, that is, every node can only communicate with its very next 4 neighbors. Lining up such multiple segments, we can build a linear array network. We grow the network in this manner since one would need a sink per given number of sensor nodes. We assume that all sensor nodes have identical characteristics, that is, E_i has the same distribution for all i, e_{ij}^t and e_{ij}^r have the same distribution among equidistant nodes, and the information generation rate Q_i is identical for all i. The network we described is motivated by oil or gas pipeline monitoring applications. The following theorem establishes a decomposition property.

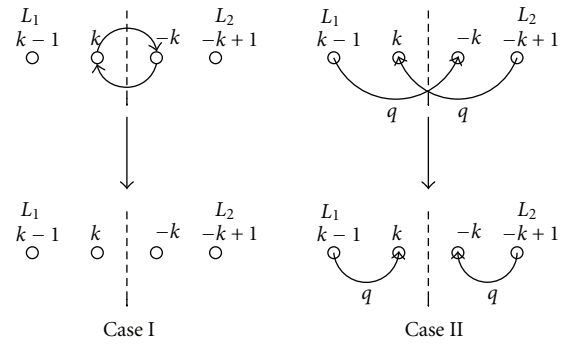


Figure 1: Flow reconstruction for an optimal flow of L.

Theorem 10. *The maximum lifetime routing problem under either the nominal* (3)*, fat* (7)*, or robust formulation* (11) *for a linear array network described above can be decomposed into the corresponding subproblems for each one of its segments.*

Proof. Without loss of generality, consider a linear array network denoted by L consisting of two segments L_1 and L_2. Consider any of the three routing formulations and let $T_{L_1}^*$, $T_{L_2}^*$, T_L^* be the optimal values for networks L_1, L_2, and, L, respectively. Clearly, $T_{L_1}^* = T_{L_2}^* \leq T_L^*$ since by combining the optimal flow vectors for L_1 and L_2 we obtain a feasible flow vector for L.

Due to homogeneity and symmetry in L, there exists an optimal flow vector which is symmetric about the center of L. Flows in the interface between the two segments L_1 and L_2 can fall into one out of two possible cases shown in Figure 1 (top). In each case, we can reconstruct the optimal flows between nodes k and $k-1$ of L_1 and nodes $-k$ and $-k+1$ of L_2 as shown in Figure 1 (bottom). This flow reconstruction process maintains feasibility and eliminates any communication between segments L_1 and L_2. Then $T_L^* = \min\{T_{L_1}, T_{L_2}\} \leq T_{L_1}^* = T_{L_2}^*$. Together with our earlier observation it follows $T_L^* = T_{L_1}^* = T_{L_2}^*$, which establishes the result. □

The following theorem establishes that the nominal formulation (3) is not particularly useful since its predicted lifetime will be achieved with a diminishing probability as the size of the network increases.

Theorem 11. *Consider a WSNET formed by aligning* 2^n *linear arrays as described before. Assume* E_i, e_{ij}^t, e_{ji}^r *are i.i.d. and nondegenerate r.v.'s (i.e., not equal to a constant). Then, as* $n \to \infty$, $P_N \to 0$.

Proof. By applying Theorem 10 n times, we decompose the network L into 2^n identical segments. With this decomposition, we have identical optimal flows in all 2^n linear segments. As we have seen before, each segment has at least one node with a binding energy constraint. Let $\mathcal{K}$ denote a set which

contains one node from each segment with a binding energy constraint. It follows that

$$\begin{aligned} P_N &= \prod_{i\in L}\mathbf{P}\left[E_i \geq \sum_{j\in\mathcal{S}_i} e_{ij}^t \hat{q}_{ij}^{*N} + \sum_{j|i\in\mathcal{S}_j} e_{ji}^r \hat{q}_{ji}^{*N}\right] \\ &\leq \prod_{k\in\mathcal{K}}\mathbf{P}\left[E_k \geq \sum_{j\in\mathcal{S}_k} e_{kj}^t \hat{q}_{kj}^{*N} + \sum_{j|k\in\mathcal{S}_j} e_{jk}^r \hat{q}_{jk}^{*N}\right] \\ &= \prod_{k\in\mathcal{K}}\mathbf{P}\left[E_k \geq \overline{E}_k\right], \end{aligned} \tag{18}$$

where the last equality follows from the fact that every $k \in \mathcal{K}$ corresponds to a binding energy constraint. Notice that $\mathbf{P}[E_k \geq \overline{E}_k] < 1$ for nondegenerate r.v.'s and that $|\mathcal{K}| = 2^n$. Hence, as $n \to \infty$, $P_N \to 0$. □

2.3.2. Square Arrays. A square array network consists of square array segments. Each segment is a two-dimensional (square) grid of a given dimension with a node at each point in the grid and a sink node located at the center point of the grid. The vertical and horizontal distance between neighboring nodes is d and we assume that the radio range is slightly less than $\sqrt{5}d$. As with linear arrays, we assume that all sensor nodes have identical characteristics, that is, E_i has the same distribution for all i, e_{ij}^t and e_{ij}^r have the same distribution among equidistant nodes, and the information generation rate Q_i is identical for all i. We grow a square network in both dimensions by stitching together segments. As an example, a network S with four segments $S_1, \ldots, S_4$ can be formed by placing segment S_1 in the northeast orthant, segment S_2 in the southeast orthant, S_3 in the southwest orthant, and S_4 in the northwest orthant. The following result is analogous to Theorem 10.

Theorem 12. *Consider a network S consisting of 4 segments $S_1, \ldots, S_4$ as outlined above. The maximum lifetime routing problem under either formulation ((3), (7), or (11)) for S can be decomposed into the corresponding problems for $S_1, \ldots, S_4$.*

Proof. Fix a particular formulation, fat, nominal, or robust. Let $T_{S_i}^*$, T_S^* be the optimal values for network S_i, $i = 1, \ldots, 4$, and S, respectively. As in the proof of Theorem 10 $T_{S_i}^* \leq T_S^*$ for all i.

Due to the homogeneity and symmetry of S, there exists an optimal flow vector for S with no flows out of sinks which is symmetric about the vertical line that separates (S_4, S_3) and (S_1, S_2). As in Theorem 10, we consider all possible cases and reconstruct the optimal flow as shown in Figure 2 (right), resulting in the new flow with no communication between (S_1, S_2) and (S_4, S_3). A similar flow reconstruction process can result in a flow with no communication between (S_1, S_4) and (S_2, S_3). These flow reconstruction steps maintain flow conservation and do not violate the energy constraints, so the resulting flow is optimal. It follows that $T_S^* \leq T_{S_i}^*$ for all i which concludes the proof. □

Analogous to the linear array case, we can now show that the nominal formulation does not provide a useful lifetime prediction. We omit the proof as it is similar to the proof of Theorem 11.

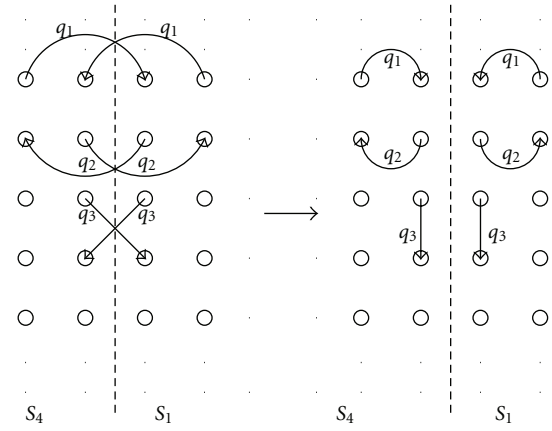


Figure 2: Flow reconstruction for an optimal flow of S.

Theorem 13. *Let a square network be constructed by repeating n times the process of constructing S from $S_1, \ldots, S_4$. Assume E_i, e_{ij}^t, e_{ji}^r are i.i.d. and nondegenerate r.v.'s (i.e., not equal to a constant). Then, as $n \to \infty$, $P_N \to 0$.*

2.4. Uncertainty Only in E_i. Here we focus on the case where uncertainty appears only in the initial available energy E_i. Namely, for all results in this subsection we assume that e_{ij}^t's and e_{ji}^r's are known with certainty.

We define a global robustness budget $\Gamma = \sum_{\text{for all } i\in\mathcal{N}\setminus\mathcal{D}} \Gamma_i$ and incorporate the allocation of Γ to individual Γ_i into the following robust formulation:

$$\max \quad T, \tag{19}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{S}_i} e_{ij}^t \hat{q}_{ij} + \sum_{j|i\in\mathcal{S}_j} e_{ji}^r \hat{q}_{ji} \leq \overline{E}_i - \Gamma_i \Delta E_i, \quad \forall i \in \mathcal{N}\setminus\mathcal{D},$$

$$(4),\ (6), \tag{20}$$

$$\sum_{\forall i\in\mathcal{N}\setminus\mathcal{D}} \Gamma_i = \Gamma, 0 \leq \Gamma_i \leq 1, \quad \forall i \in \mathcal{N}\setminus\mathcal{D},$$

where the decision variables are T, the $\hat{q}_{ij}$'s, and the Γ_i's. The following monotonicity property is immediate. Concavity follows from the fact that (19) maximizes a concave (linear) objective over linear constraints and Γ appears in the right hand side of these constraints.

Proposition 14. *The optimal value T_R^* of (19) is monotonically nonincreasing and concave as a function of the global robustness budget Γ.*

2.4.1. Optimizing $\boldsymbol{P}[T \geq T^*]$ over the Optimal Flows $\boldsymbol{q}^*$. When the uncertainty is only in E_i's, we can maximize the

lifetime guarantee probability $\mathbf{P}[T \geq T^*]$ over the set of optimal flows $\mathbf{q}^*$ while guaranteeing that we achieve the corresponding predicted lifetime. One can think of this optimization as maximizing "robustness" while guaranteeing the same objective (predicted lifetime). We next show that this problem is a well-structured concave optimization problem. We only treat the robust case. For the fat case we have already shown that $P_F = 1$ and the nominal case is similar to the robust.

Assume that only E_i's are uncertain, and let $T_R^*, \mathbf{s}^*, \mathbf{q}^*, \Gamma^*$ form an optimal solution of the robust formulation (19), where $\mathbf{s}^*$ denotes the vector of slack variables corresponding to the energy constraints. Suppose all E_i's are independent, then

$$\begin{aligned} P_R &= \mathbf{P}\left[\min_{i\in\mathcal{N}\setminus\mathcal{D}} \frac{E_i}{\sum_{j\in\mathcal{S}_i} \overline{e}_{ij}^t q_{ij}^* + \sum_{j|i\in\mathcal{S}_j} \overline{e}_{ji}^r q_{ji}^*} \geq T_R^*\right] \\ &= \prod_{i\in\mathcal{N}\setminus\mathcal{D}} \mathbf{P}\left[E_i \geq \overline{E}_i - s_i^* - \Gamma_i^* \Delta E_i\right]. \end{aligned} \tag{21}$$

Taking the E_i's to be uniformly distributed in $[\overline{E}_i - \Delta E_i, \overline{E}_i + \Delta E_i]$:

$$\begin{aligned} &\mathbf{P}\left[E_i \geq \overline{E}_i - s_i^* - \Gamma_i^* \Delta E_i\right] \\ &\quad = \frac{\Delta E_i + \min\{s_i^* + \Gamma_i^* \Delta E_i, \Delta E_i\}}{2\Delta E_i} \triangleq p_i, \end{aligned} \tag{22}$$

and $P_R = \prod_{i\in\mathcal{N}\setminus\mathcal{D}} p_i$, where we defined $2p_i\Delta E_i = \Delta E_i + \min\{s_i^* + \Gamma_i^* \Delta E_i, \Delta E_i\}$.

To maximize P_R while achieving the optimal lifetime T_R^*, we can equivalently maximize $\ln(P_R)$ which yields the following concave optimization problem:

$$\begin{aligned} \max\ & \ln(P_R) = \sum_{i\in\mathcal{N}\setminus\mathcal{D}} \ln p_i \\ \text{s.t.}\ & (4),\ \sum_{j\in\mathcal{S}_i} \overline{e}_{ij}^t \hat{q}_{ij} + \sum_{j|i\in\mathcal{S}_j} \overline{e}_{ji}^r \hat{q}_{ji} + s_i = \overline{E}_i - \Gamma_i \Delta E_i, \\ & \qquad \forall i \in \mathcal{N}\setminus\mathcal{D}, \\ & T \geq T_R^*, s_i \geq 0, \hat{q}_{ij} \geq 0,\ \forall i \in \mathcal{N},\ \forall j \in \mathcal{S}_i, \\ & 2p_i\Delta E_i - \Delta E_i \leq s_i + \Gamma_i \Delta E_i,\ \forall i \in \mathcal{N}\setminus\mathcal{D}, \\ & (20),\ \ 2p_i\Delta E_i - \Delta E_i \leq \Delta E_i,\ \forall i \in \mathcal{N}\setminus\mathcal{D}. \end{aligned} \tag{23}$$

3. Maximum Lifetime Routing with Energy Allocation

In this section, we consider the problem of maximizing the WSNET lifetime by jointly optimizing the routing decisions and the initial energy allocated to the nodes. Suppose E is the total available energy for a WSNET. Similar to formulation (3) we have the nominal problem:

$$\begin{aligned} \max\ & T \\ \text{s.t.}\ & (4),\ (5),\ (6), \\ & \sum_{i\in\mathcal{N}\setminus\mathcal{D}} E_i = E, \quad E_i \geq 0,\ \forall i \in \mathcal{N}\setminus\mathcal{D}. \end{aligned} \tag{24}$$

Here the E_i's (appearing in (5) and above) are decision variables. The corresponding fat and robust formulations, respectively, are

$$\begin{aligned} \max\ & T \\ \text{s.t.}\ & \sum_{j\in\mathcal{S}_i} \overline{e}_{ij}^t \hat{q}_{ij} + \sum_{j\in J_i^t} \Delta e_{ij}^t \hat{q}_{ij} + \sum_{j|i\in\mathcal{S}_j} \overline{e}_{ji}^r \hat{q}_{ji} + \\ & \sum_{j\in J_i^r} \Delta e_{ji}^r \hat{q}_{ji} \leq E_i,\ \forall i \in \mathcal{N}\setminus\mathcal{D}, \\ & (4),\ (6), \\ & \sum_{i\in\mathcal{N}\setminus\mathcal{D}} E_i = E, \quad E_i \geq 0,\ \forall i \in \mathcal{N}\setminus\mathcal{D}, \end{aligned} \tag{25}$$

$$\begin{aligned} \max\ & T, \\ \text{s.t.}\ & \max_{e_{ij}^t, e_{ji}^r \in \mathcal{R}_i(\Gamma_i^e)} \left\{ \sum_{j\in\mathcal{S}_i} e_{ij}^t \hat{q}_{ij} + \sum_{j|i\in\mathcal{S}_j} e_{ji}^r \hat{q}_{ji} \right\} \leq E_i, \\ & \qquad \forall i \in \mathcal{N}\setminus\mathcal{D}, \\ & (4),\ (6), \\ & \sum_{i\in\mathcal{N}\setminus\mathcal{D}} E_i = E, \quad E_i \geq 0,\ \forall i \in \mathcal{N}\setminus\mathcal{D}. \end{aligned} \tag{26}$$

(27)

As before, the robust problem (26) can be shown to be equivalent to a linear programming problem; we omit the details for brevity. From the structure of the formulation with energy allocation, we have the following result.

Proposition 15. *At optimality, all the energy constraints for nonsink nodes are active and the total energy constraint is also binding. This holds for all three formulations.*

Proof. Consider first the robust problem (26). We will use contradiction. Assume that at optimality, the energy constraint (27) for some nonsink node k is not active. Notice that we can decrease E_k and increase all the other E_i while maintaining their sum. This improves the lifetime which contradicts optimality. Similarly, the total energy constraint is also binding at optimality. If not, we can increase all E_i to achieve a better lifetime, which again contradicts optimality. The nominal and fat cases are almost identical. □

3.1. Properties of Optimal Solutions. As before, let T_N^*, T_F^*, T_R^* denote the optimal values of the nominal, fat, and robust

routing problems, respectively. Let $\mathbf{\Gamma}^e = (\Gamma_1^e, \ldots, \Gamma_{|\mathcal{N}\setminus\mathcal{D}|}^e)$. Note that T_R^* depends on $\mathbf{\Gamma}^e$. To express this dependence, we write $T_R^*(\mathbf{\Gamma}^e)$. The following result is similar to Proposition 2.

Proposition 16. *$T_R^*(\mathbf{\Gamma}^e)$ is a nonincreasing function of $\mathbf{\Gamma}^e$ and $T_F^* \le T_R^*(\mathbf{\Gamma}^e) \le T_N^*$.*

As in Section 2.2, one associates a directed graph $\mathcal{G}_\mathbf{q} = (\mathcal{N}, \mathcal{A}_\mathbf{q})$ to a feasible flow vector $\mathbf{q}$, where $\mathcal{A}_\mathbf{q}$ contains all (i, j) with $q_{i,j} > 0$. Recall that we name $\mathbf{q}$ as *acyclic* when $\mathcal{G}_\mathbf{q}$ contains no cycles. The following results are similar to Theorem 5 and Corollary 7; we omit the proofs.

Proposition 17. *For all three formulations* (24), (25), *and* (26), *the optimal flows are acyclic.*

Proposition 18. *For all three formulations the optimal flows $\mathbf{q}$ satisfy $q_{ij} = 0$, for all $i \in \mathcal{D}$.*

3.2. Lifetime Guarantee Probability. The development in this section is similar to Section 2. We have the following results; we omit the details in the interest of brevity.

Proposition 19. *It holds $P_F = 1$.*

Note that when $\Gamma_i^e \to |J_i^t| + |J_i^r|$, for all $i \in \mathcal{N} \setminus \mathcal{D}$, then $P_R \to P_F$; while when $\Gamma_i^e \to 0$, for all $i \in \mathcal{N} \setminus \mathcal{D}$, then $P_R \to P_N$.

Proposition 20. *If e_{ij}^t, e_{ji}^r are independent symmetrically distributed r.v.'s, then $P_N = (1/2)^{|\mathcal{N}\setminus\mathcal{D}|}$.*

It follows that as $|\mathcal{N} \setminus \mathcal{D}| \to \infty$ we have $P_N \to 0$, and this now holds for all topologies.

4. Routing and Energy Allocation in Massively Dense WSNETs

It is straightforward that the joint problem of routing and energy allocation (24) is equivalent to finding paths from sources to sinks with lowest energy consumption rate. If we consider the energy consumed by both the sender and the receiver over a link as the cost (or length) of the link, the problem is reduced to finding shortest paths between sources and sinks. Imagine now that the WSNET is scaled by uniformly deploying an increasing number of nodes while decreasing their radio range in order to maintain a fixed density of one-hop-reachable neighbors. Although the approach we developed so far scales well since we are dealing with linear programming problems, it is of interest to consider whether the scaled problem exhibits, in the limit, a structure that simplifies its solution and deepens our understanding. In particular, we will consider a limiting regime of massively dense WSNETs and study maximum lifetime routing formulations with energy allocation. Such WSNETs can only be described by macroscopic parameters, such as the information generation and energy distribution densities.

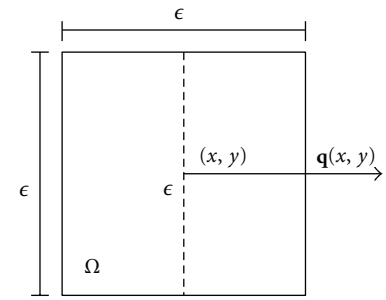


Figure 3: The information traffic flow function $\mathbf{q}(x, y)$.

4.1. Problem Formulation. Let $\mathcal{M}$ be the planar area where a massively dense WSNET is deployed. Mathematically, $\mathcal{M}$ is a convex set in $\mathbb{R}^2$. We assume that the WSNET is uniformly deployed over $\mathcal{M}$.

Let $Q(x, y)$ represent the information generation density function defined on $\mathcal{M}$ whose units are bits/(sec·m^2). We assume $Q(x, y)$ is known. Denote by $S(x, y)$ the information consumption density function defined on $\mathcal{M}$ whose units are bits/(sec·m^2). In the next subsection we will consider the special cases of "point" sources and sinks where $Q(x, y)$ and $S(x, y)$ become Dirac functions on the plane. Let $e(x, y)$ be the energy density function defined on $\mathcal{M}$ whose units are J/m^2. The energy density function $e(x, y)$ characterizes the distribution of the globally available energy E over $\mathcal{M}$. Define the information traffic flow function as $\mathbf{q}(x, y) = (q_x(x, y), q_y(x, y))$. The interpretation of $\mathbf{q}(x, y)$ is as follows: $\epsilon\|\mathbf{q}(x, y)\|$ is the rate at which information crosses a linear segment of infinitesimal length ϵ which is centered on (x, y) and perpendicular to $\mathbf{q}(x, y)$ (see Figure 3). The units of $\|\mathbf{q}\|$ are bits/(sec · m).

The continuous maximum life routing problem with energy allocation can be formulated as:

$$\max \quad T, \tag{28}$$

$$\text{s.t.} \quad \frac{\partial q_x(x, y)}{\partial x} + \frac{\partial q_y(x, y)}{\partial y} = Q(x, y) - S(x, y), \quad \forall (x, y) \in \mathcal{M}, \tag{29}$$

$$T \le \lim_{\epsilon \to 0} \frac{\int_{\Omega(\epsilon)} e(x, y)\,\mathrm{d}\sigma}{\alpha\, \epsilon\, ||\mathbf{q}(x, y)||}, \quad \forall (x, y) \in \mathcal{M}, \tag{30}$$

$$\int_{\mathcal{M}} e(x, y)\,\mathrm{d}\sigma = E, \tag{31}$$

$$\int_{\mathcal{M}} (Q(x, y) - S(x, y))\,\mathrm{d}\sigma = 0,$$

$$T \ge 0, \quad S(x, y) \ge 0, \quad e(x, y) \ge 0, \quad \forall (x, y) \in \mathcal{M}, \tag{32}$$

where $S(x, y)$, $e(x, y)$, $\mathbf{q}(x, y)$, and T are decision functions and variables. Using an argument in [21], (29) states that the

divergence of the traffic flow function measures the degree with which the traffic increases or decreases; we can think of this as a detailed flow conservation equation. (31) is a global energy constraint while (32) can be seen as a global flow conservation constraint. As for (30), consider a point $(x, y) \in \mathcal{M}$ and let $\Omega(\epsilon)$ denote an infinitesimal square centered at (x, y) with a side length equal to ϵ and one of its sides parallel to $\mathbf{q}(x, y)$. Let α (in J/(bit · sec)) be a constant indicating how much energy is consumed per unit of transmitted information per second. Then, (30) expresses the fact that the total energy consumed when the traffic flow $\mathbf{q}(x, y)$ passes through $\Omega(\epsilon)$ during a period of time T should be no more than the total energy available in this area.

In this section, we are only interested in the structure of the optimal solutions to (28), hence we only consider the nominal version of the problem. Uncertainty in E can be easily incorporated as we have done with the discrete instances. This will only change the right hand side of the total energy constraint and would not affect the optimal solution structure. Uncertainty in $e(x, y)$ can also be incorporated but that is beyond the main focus of this section.

From the structure of (28), we have the following results. The proof is immediate as whenever $\|\mathbf{q}(x, y)\| = 0$ and $e(x, y) > 0$ we can reduce $e(x, y)$ to zero while maintaining feasibility. The energy savings can be allocated to other points resulting in a potential increase of the lifetime.

Proposition 21. (28) *has optimal solutions such that* $e(x, y) = 0$ *whenever* $\|\mathbf{q}(x, y)\| = 0$.

Similar to Proposition 15 we can showthe following.

Proposition 22. *For problem* (28)*, there exist optimal solutions such that the detailed energy constraints* (30) *are all active.*

4.2. Single Point Source and Sink. In this subsection, we focus on the scenario where there is a single point source and a single point sink in a massively dense WSNET. We start with the definition of a point source/sink.

Definition 23 (point source on (x_o, y_o)). Let $\mathbf{o} = (x_o, y_o)$ be the location of the source on $\mathcal{M}$ and denote by Q its information generation rate. The point information generation density function $Q_{\mathbf{o}}(x, y)$ satisfies

$$Q_{\mathbf{o}}(x, y) = \begin{cases} 0 & (x, y) \neq (x_o, y_o), \\ +\infty & (x, y) = (x_o, y_o), \end{cases} \qquad \int_{\mathbb{R}^2} Q_{\mathbf{o}}(x, y)\,\mathrm{d}\sigma = Q. \tag{33}$$

Similarly, we define the information consumption density function $S_{\mathbf{s}}(x, y)$ for a point sink at (x_s, y_s) with a sink rate equal to S. These are Dirac impulse functions on $\mathbb{R}^2$.

In the single point source and single point sink case, let $\mathbf{o} = (x_o, y_o)$ and $\mathbf{s} = (x_s, y_s)$ be the source and sink locations, respectively. Denote by $Q_o(x, y)$ and $S_s(x, y)$ the corresponding information generation/consumption density function with rates Q and S, respectively. It follows from the global flow balance equation (32) that $Q = S$. We next define the notion of a marginal density function; its units are J/m.

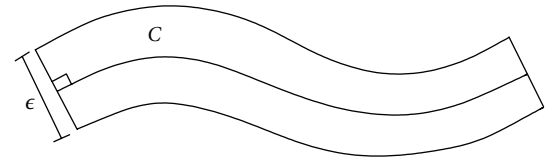


Figure 4: C and its ϵ-tube.

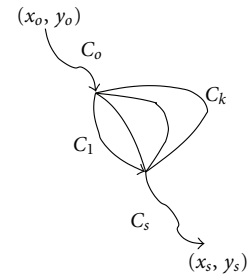


Figure 5: An arbitrary set of paths from **o** to **s**.

Definition 24 (marginal energy density function on curve C). Let C be a continuous curve connecting two points **o** and **s**, and let $\mathcal{T}(C, \epsilon)$ denote an ϵ-tube around C as shown in Figure 4. The marginal energy density function $e_C(x, y)$ on curve C satisfies

$$e_C(x, y) = \begin{cases} 0, & (x, y) \notin C, \\ \lim\limits_{\epsilon \to 0} \dfrac{\int_{\Omega(\epsilon)} e(x, y)\,\mathrm{d}\sigma}{\epsilon}, & (x, y) \in C. \end{cases} \tag{34}$$

4.2.1. Properties of Optimal Solutions. Let $\mathbf{o} = (x_o, y_o)$ and $\mathbf{s} = (x_s, y_s)$ be the source and sink positions, respectively. Information generated at the source (with rate Q) gets consumed at the sink (with rate S, where it follows that $Q = S$).

Consider an arbitrary set of paths (see Figure 5) traversed by the traffic as it flows from **o** to **s**. Specifically, the traffic flow first follows curve C_o then forks into branches $C_1, C_2, \ldots, C_k$ which merge into C_s. Denote by $Q_o, \ldots, Q_k, Q_s$ the traffic flows on $C_o, \ldots, C_k, C_s$, respectively, where $\sum_{i=1}^{k} Q_i = Q_o = Q_s$. We have already established that at optimality we have an energy density function that is nonzero only on the curves $Q_o, \ldots, Q_k, Q_s$ and all energy constraints are active. The problem reduces to

$$\begin{aligned} \max \quad & T \\ \text{s.t.} \quad & \alpha T Q_i = e_{C_i}, \quad i \in \{o, 1, 2, \ldots, k, s\} \\ & \sum_i e_{C_i} \ell(C_i) = E, \quad T \geq 0, \quad e_{C_i} \geq 0, \ \forall (x, y) \in C_i, \end{aligned} \tag{35}$$

where $\ell(C_i)$ denotes the length of the curve C_i. Note this is a linear program. Given the allocation of Q into $Q_1, \ldots, Q_k$ an optimal lifetime is

$$T = \frac{E}{\sum_{i\in\{o,1,2,\ldots,k,s\}} \alpha Q_i \ell(C_i)}. \quad (36)$$

It follows that to maximize T, the branches $C_o, \ldots, C_k, C_s$ should all be straight lines (minimal length). Furthermore, the best T can be achieved by a straight line from **o** to **s**. The result is summarized below.

Proposition 25. *The path that maximizes the network's lifetime is the straight line from* **o** *to* **s**. *The corresponding energy distribution function $e(x, y)$ is nonzero only on this line with a uniform marginal energy density function.*

One notes that the argument above can be extended to handle an infinite number of (forked and merged) paths. The key idea is the same, that is, one can show that any solution using an infinite number of paths is no better than the straight line connecting **o** with **s**. one will omit the details to avoid obfuscating the discussion.

4.3. Multiple Point Sources and Sinks. The result in the previous subsection readily generalizes to the situation where we have n point sources, say $o_1, \ldots, o_n$, and m sinks, denoted by $s_1, \ldots, s_m$. The result is provided in the following proposition; we omit the details because it follows the same line of reasoning.

Proposition 26. *For problem* (28) *with multiple point sources and sinks, there exist an optimal solution such that every source sends its information to its nearest sink along the straight line segment connecting them, and the corresponding marginal energy density function on the line segment is uniform.*

The result implies that sinks generate a Voronoi tessellation of the deployment area, and the sources send their flows over straight lines to the sink in the cell they reside in, thus, resulting in a star-like network within each cell.

5. Numerical Experiments

In this section, we present a set of numerical examples. For all examples we adopt the communication energy consumption model from [12].

Let d^r be the transmission range of each node. Then $j \in \mathcal{S}_i$ if and only if $d_{ij} \le d^r$, where d_{ij} is the distance between nodes i and j. The energy expenditure per data unit transmitted from i to j satisfies $e^t_{ij} = e^\circ + \epsilon_{\text{amp}} d^4_{ij}$, $e^r_{ij} = e^R$, where $e^\circ = 50$ nJ/bit and $e^R = 150$ nJ/bit denote the energy consumed in the transceiver circuitry at the transmitter and the receiver, respectively, and $\epsilon_{\text{amp}} = 100$ pJ/bit/m^4 is the energy consumed at the output transmitter antenna for transmitting a bit over one meter. The receiver circuitry is in general more complex and consumes more energy than the transmitter circuitry within the same order of magnitude. The path loss exponent of four is chosen to account for multipath reflections. In all the numerical experiments P_R is estimated by Monte-Carlo simulation with 10^6 samples, thus P_R is accurate with a ± 0.005 error and 99% confidence (by Chebyshev's inequality).

5.1. A 4-Node WSNET. We start with a toy example to give some intuition on the routing policies produced by each formulation. The WSNET consists of one origin node O, two relay nodes, R_1 and R_2, and one sink node S, where $Q_O = 500$ bits/sec and the radio range is 30 m. The origin node O has to use relays R_1 and R_2 to reach the sink S. Further, $20\,\text{m} = d_{OR_1} = d_{R_1S} < d_{OR_2} = d_{R_2S} = 21\,\text{m}$. First, we consider the case without energy allocation.

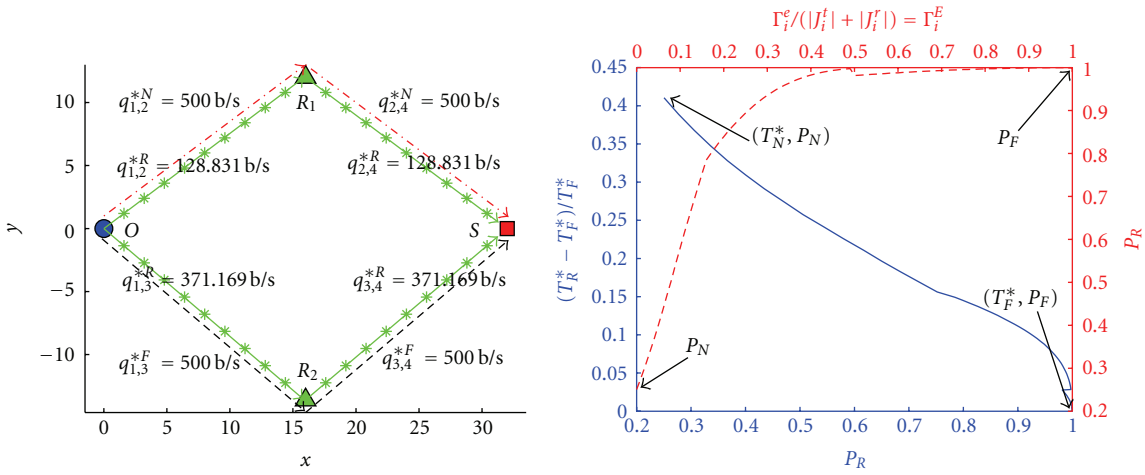
5.1.1. Routing without Energy Allocation. All E_i, e^t_{ij}, e^r_{ji} are uniformly distributed with $E_O \in [9.0, 10.0]$ J where $\overline{E}_O = 9.5$ J, $E_{R_1} \in [8.1, 11.1]$ J where $\overline{E}_{R_1} = 9.6$J, $E_{R_2} \in [9.5, 10.5]$ J with $\overline{E}_{R_2} = 10.0$J, $\Delta e^t_{OR_1}/\overline{e}^t_{OR_1} = \Delta e^t_{R_1S}/\overline{e}^t_{R_1S} = 0.35$, and $\Delta e^t_{ij}/\overline{e}^t_{ij} = 0.1$ for all other appropriate i and j. $\Delta e^r_{ji}/\overline{e}^r_{ji} = 0.1$ for all appropriate i and j. Note that $\Gamma^e_O \in [0, 4]$, $\Gamma^e_{R_1} \in [0, 6]$, $\Gamma^e_{R_2} \in [0, 6]$, and $\Gamma^E_i \in [0, 1]$ for all i. Take $\Gamma^e_O = 1.04$, $\Gamma^e_{R_1} = \Gamma^e_{R_2} = 1.56$, and $\Gamma^E_i = 0.26$ for all i.

In Figure 6(a), the red (dot-dash), black (dash), and green (solid-star) lines with arrows represent the nominal, fat, and robust optimal flows, respectively. Note the difference in the selected routes: the nominal picks the shorter path $O - R_1 - S$, the fat picks the more "stable" but a little longer path $O - R_2 - S$, while the robust balances the two to maintain a relatively high lifetime guarantee probability while not suffering too much from the low predicted lifetime.

As we adjust $\Gamma^e_i/(|J^t_i| + |J^r_i|) = \Gamma^E_i$, P_R and T^*_R will change accordingly. The solid blue curve in Figure 6(b) describes the relationships between P_R and $(T^*_R - T^*_F)/T^*_F$ (the percentage predicted lifetime gain of the robust formulation over the fat). It can be seen that there is significant predicted lifetime gain (e.g., 15%) while the lifetime guarantee probability remains high (e.g., close to 0.8). The red dash curve represents the relationship $\Gamma^e_i/(|J^t_i| + |J^r_i|) = \Gamma^E_i$ versus P_R. It can be seen that as we protect more against the randomness, the predicted lifetime T^*_R goes down and the lifetime guarantee probability P_R gets enhanced. The two extreme cases of no protection and full protection correspond to the nominal and fat situations.

To gain further insight on the impact of uncertainty on the nominal formulation, consider the probability distribution of the actual lifetime T achieved by applying the nominal optimal policy $\mathbf{q}^*_N$ to random instances (where e^t_{ij}, e^r_{ji}, and E_i are randomly selected). Figure 7 shows the histogram of T generated from a million instances. We can see that T can be substantially smaller than T^*_N and in fact most of the probability mass corresponds to such T's. The nominal lifetime guarantee probability $P_N = \mathbf{P}[T \ge T^*_N]$ would be fairly low but that does not capture how far from T^*_N the actual lifetime T can be.

5.1.2. Routing with Energy Allocation. If energy allocation is an option, set the global available energy $E = 30$J. As before, $(\Delta e^t_{OR_1})/(\overline{e}^t_{OR_1}) = (\Delta e^t_{R_1S})/(\overline{e}^t_{R_1S}) = 0.35$, and $(\Delta e^t_{ij})/(\overline{e}^t_{ij}) =$



(a) (b)

FIGURE 6: (a) Optimal Flows. (b) P_R versus $(T^*_R - T^*_F)/T^*_F$ (solid blue) and $\Gamma^e_i/(|J^t_i| + |J^r_i|) = \Gamma^E_i$ versus P_R (dash red).

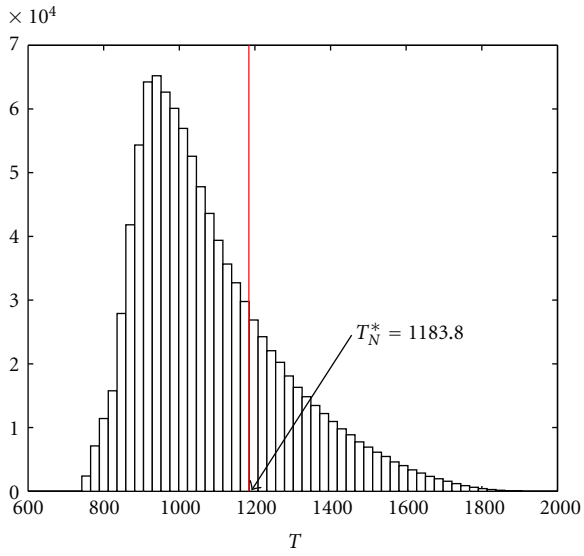


FIGURE 7: Histogram of T under $\mathbf{q}^*_N$ for the 4-node WSNET.

TABLE 1: T^* and lifetime guarantee probability for the 4-node WSNET.

No energy allocation	Nominal	Fat	Robust
$T^*((T^*_R - T^*_F)/T^*_F \times 100\%)$	1183.8	839.24	931.64 (11%)
Lifetime guarantee prob.	0.25	1.0	0.90
With energy allocation	Nominal	Fat	Robust
$T^*((T^*_R - T^*_F)/T^*_F \times 100\%)$	1860.47	1393.38	1448.4 (4%)
Lifetime guarantee prob.	0.25	1.0	0.82

TABLE 2: T^* and lifetime guarantee probability for the randomly deployed WSNET.

Routing without energy allocation for the random WSNETs			
	Nominal	Fat	Robust
$T^*((T^*_R - T^*_F)/T^*_F \times 100\%)$	2249.84	1151.67	1287.74 (11.8%)
Lifetime guarantee prob.	0.063	1.0	0.905
Routing without energy allocation for the random WSNETs			
$T^*((T^*_R - T^*_F)/T^*_F \times 100\%)$	19039.9	15693.8	16056.2 (2.31%)
Lifetime guarantee prob.	5.09×10^{-4}	1.0	0.89

0.1 for all other appropriate i and j. $\Delta e^r_{ji}/\overline{e}^r_{ji} = 0.1$ for all appropriate i and j. Note that $\Gamma^e_O \in [0, 4]$, $\Gamma^e_{R_1} \in [0, 6]$, $\Gamma^e_{R_2} \in [0, 6]$. Take $\Gamma^e_O = 0.8$, $\Gamma^e_{R_1} = \Gamma^e_{R_2} = 1.2$.

Figure 8(a) presents the nominal, fat, and robust optimal flows and energy allocation. The situation is very similar as before but energy allocation improves the predicted lifetime since no energy is wasted. Optimal values in a number of nominal, fat, and robust cases with and without energy allocation are listed in Table 1.

5.2. A Randomly Deployed WSNET. In this case, we have 20 nodes (4 sinks, 10 origins, 6 relays) uniformly deployed on a 50 × 50 m^2 square. $d^r = 25$ m. $Q_i = 500$ bits/sec, for all $i \in \mathcal{O}$. All E_i, e^t_{ij}, e^r_{ji} are uniformly distributed and $\overline{E}_i = 10$J, $\Delta E_i/\overline{E}_i$ is uniformly sampled from $[0, 0.3]$. $J^t_i = \mathcal{S}_i$, $J^r_i = \{j \mid i \in \mathcal{S}_j\}$, $\Delta e^t_{ij}/\overline{e}^t_{ij}$ and $\Delta e^r_{ji}/\overline{e}^r_{ji}$ are uniformly sampled from $[0, 0.4]$.

5.2.1. Routing without Energy Allocation. We use $\Gamma^e_i/(|J^t_i| + |J^r_i|) = \Gamma^E_i = 0.71$ and solve the three routing problems without energy allocation. The policies are quite different since we compute $(\|\mathbf{q}^*_N - \mathbf{q}^*_F\|)/\|\mathbf{q}^*_F\| = 0.33$ and $(\|\mathbf{q}^*_R - \mathbf{q}^*_F\|)/\|\mathbf{q}^*_F\| = 0.92$.

5.2.2. Routing with Energy Allocation. Let now $\Gamma^e_i/(|J^t_i| + |J^r_i|) = 0.09$. Solving the problems with energy allocation we obtain $(\|\mathbf{q}^*_N - \mathbf{q}^*_F\|)/\|\mathbf{q}^*_F\| = 0.32$ and $(\|\mathbf{q}^*_R - \mathbf{q}^*_F\|)/\|\mathbf{q}^*_F\| = 0.19$. The results for both cases are presented in Table 2.

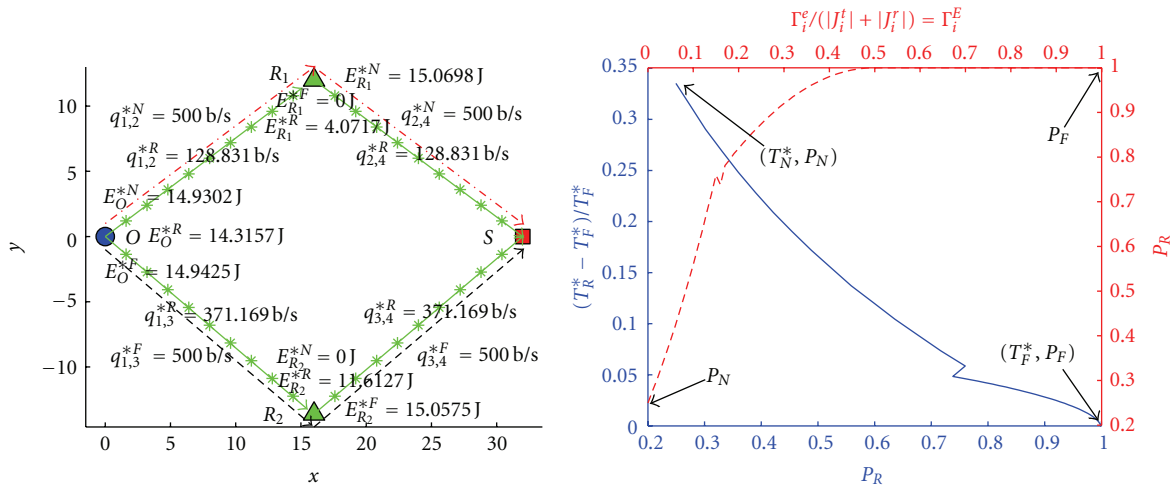


FIGURE 8: (a) Optimal flows and energy. The red (dot-dash), black (dash), and green (solid-star) lines depict the nominal, fat, and robust optimal flows, respectively. (b) P_R versus $(T^*_R - T^*_F)/T^*_F$ (solid blue) and $\Gamma^e_i/(|J^t_i| + |J^r_i|) = \Gamma^E_i$ versus P_R (dash red).

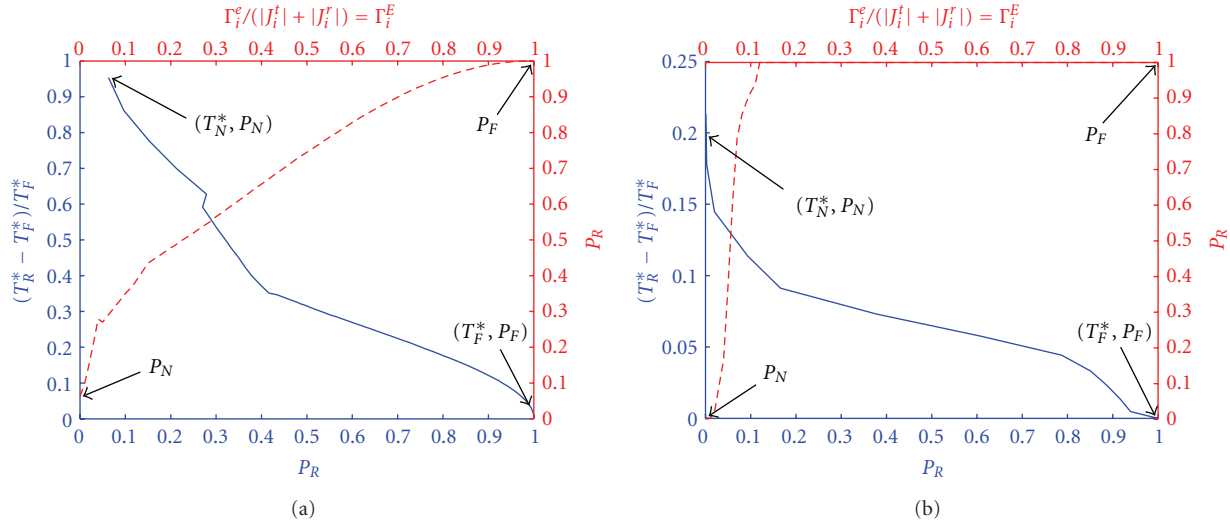


FIGURE 9: (a) P_R versus $(T^*_R - T^*_F)/T^*_F$ (blue-solid) and $\Gamma^e_i/(|J^t_i| + |J^r_i|) = \Gamma^E_i$ versus P_R (red-dash) without energy allocation. (b) P_R versus $(T^*_R - T^*_F)/T^*_F$ (blue-solid) and $\Gamma^e_i/(|J^t_i| + |J^r_i|) = \Gamma^E_i$ versus P_R (red-dash) with energy allocation.

Again adjusting $\Gamma^e_i/(|J^t_i| + |J^r_i|) = \Gamma^E_i$ or $\Gamma^e_i/(|J^t_i| + |J^r_i|)$, respectively, for the two cases, changes P_R and T^*_R accordingly (see Figures 9(a) and 9(b)). It can be seen that as we protect more against the randomness, the predicted lifetime T^*_R goes down and the lifetime guarantee probability P_R gets enhanced. For energy allocation problems, since at optimality all energy constraints are active, the lifetime guarantee probability gets reduced but still the gain over the fat formulation is nonnegligible.

As we did in the 4-node example, we plot in Figure 10 the histogram of T achieved by $\mathbf{q}^*_N$ computed from a million random instances of the problem (without energy allocation). It is clear that as the number of nodes grows the probability mass for T shifts away from T^*_N and the actual T is typically substantially smaller than T^*_N. This is consistent with our result that $P_N = \mathbf{P}[T \geq T^*_N] \to 0$.

6. Conclusions

We presented a new framework to accommodate uncertainty in designing maximum lifetime routing policies for WSNETs. We considered two scenarios—one (Scenario A) assuming that energy is already allocated to various nodes and the other (Scenario B) where such allocation is also subject to optimization. We formulated a worst case (fat) problem and compared it with the nominal problem that makes certainty equivalence assumptions and ignores uncertainty. As a compromise between the two, we also devised a robust

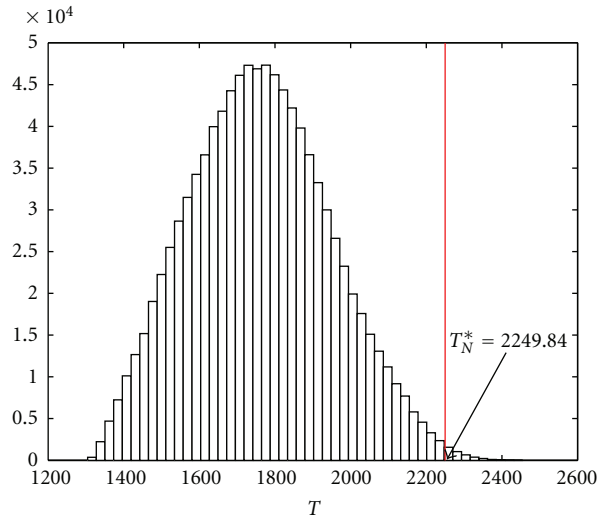


FIGURE 10: Histogram of T under $\mathbf{q}_N^*$ for the 20 node randomly deployed WSNET.

formulation. We established, analytically and numerically, that the nominal solutions are always too optimistic. Specifically, for common Scenario A topologies (like regular linear arrays and grid-like WSNETs) the nominal formulation predicts a lifetime that is (almost) never achieved in the presence of uncertainty. In Scenario B, the same result holds for all topologies. The robust solutions, on the other hand, provide a useful and practical way to tradeoff performance versus robustness. We extended our analysis to massively dense WSNETs and characterized optimal solutions of the routing problems.

Appendix

Proof of Theorem 1

Let $(\hat{\mathbf{q}}^*, T^*, \mathbf{p}^*, \omega^*, \nu^*)$ and $(\hat{\mathbf{q}}^\circ, T^\circ)$ be optimal solutions of (13) and (11), respectively. We will show that $(\hat{\mathbf{q}}^*, T^*)$ is a feasible solution of (11) with $T^* = T^\circ$. For any $\hat{\mathbf{q}} \geq \mathbf{0}$, the maximization problem in the energy constraint for node i in the robust problem (11) is

$$\max_{e_{ij}^t, e_{ji}^r \in \mathcal{R}_i(\Gamma_i^e)} \left\{ \sum_{j \in \mathcal{S}_i} e_{ij}^t \hat{q}_{ij} + \sum_{j:i \in \mathcal{S}_j} e_{ji}^r \hat{q}_{ji} \right\}. \tag{A.1}$$

Note that (A.1) is equivalent to the following linear optimization problem:

$$\begin{aligned}
\max \quad & \sum_{j \in \mathcal{S}_i} \overline{e}_{ij}^t \hat{q}_{ij} + \sum_{j \in J_i^t} \Delta e_{ij}^t \hat{q}_{ij} z_{ij} + \sum_{j|i \in \mathcal{S}_j} \overline{e}_{ji}^r \hat{q}_{ji} + \sum_{j \in J_i^r} \Delta e_{ji}^r \hat{q}_{ji} \lambda_{ji} \\
\text{s.t.} \quad & \sum_{j \in J_i^t} z_{ij} + \sum_{j \in J_i^r} \lambda_{ji} \leq \Gamma_i^e, \\
& 0 \leq z_{ij} \leq 1, \ \forall j \in J_i^t, \\
& 0 \leq \lambda_{ji} \leq 1, \ \forall j \in J_i^r,
\end{aligned} \tag{A.2}$$

where (z_{ij}, λ_{ji}) are the decision variables. Then the dual of (A.2) is:

$$\begin{aligned}
\min \quad & \sum_{j \in \mathcal{S}_i} \overline{e}_{ij}^t \hat{q}_{ij} + \sum_{j|i \in \mathcal{S}_j} \overline{e}_{ji}^r \hat{q}_{ji} + \Gamma_i^e p_i + \sum_{j \in J_i^t} \omega_{ij} + \sum_{j \in J_i^r} \nu_{ji} \\
\text{s.t.} \quad & p_i + \omega_{ij} \geq \Delta e_{ij}^t \hat{q}_{ij}, \ \forall j \in J_i^t, \\
& p_i + \nu_{ji} \geq \Delta e_{ji}^r \hat{q}_{ji}, \ \forall j \in J_i^r, \\
& \omega_{ij} \geq 0, \ \forall j \in J_i^t, \qquad \nu_{ji} \geq 0, \ \forall j \in J_i^r, \\
& p_i \geq 0,
\end{aligned} \tag{A.3}$$

where $(p_i, \omega_{ij}, \nu_{ji})$ are the dual variables. Fix $\hat{\mathbf{q}} = \hat{\mathbf{q}}^*$ in (A.2) and (A.3), and let $(z_{ij}^*, \lambda_{ji}^*)$ be an optimal solution of (A.2). Note that $(p_i^*, \omega_{ij}^*, \nu_{ji}^*)$ is feasible for (A.3). For all $i \in \mathcal{N} \setminus \mathcal{D}$ we have

$$\begin{aligned}
& \max_{e_{ij}^t, e_{ji}^r \in \mathcal{R}_i(\Gamma_i^e)} \left\{ \sum_{j \in \mathcal{S}_i} e_{ij}^t \hat{q}_{ij}^* + \sum_{j:i \in \mathcal{S}_j} e_{ji}^r \hat{q}_{ji}^* \right\} \\
& = \sum_{j \in \mathcal{S}_i} \overline{e}_{ij}^t \hat{q}_{ij}^* + \sum_{j \in J_i^t} \Delta e_{ij}^t \hat{q}_{ij}^* z_{ij}^* + \sum_{j|i \in \mathcal{S}_j} \overline{e}_{ji}^r \hat{q}_{ji}^* + \sum_{j \in J_i^r} \Delta e_{ji}^r \hat{q}_{ji}^* \lambda_{ji}^* \\
& \leq \sum_{j \in \mathcal{S}_i} \overline{e}_{ij}^t \hat{q}_{ij}^* + \sum_{j|i \in \mathcal{S}_j} \overline{e}_{ji}^r \hat{q}_{ji}^* + \Gamma_i^e p_i^* + \sum_{j \in J_i^t} \omega_{ij}^* + \sum_{j \in J_i^r} \nu_{ji}^* \\
& \leq \overline{E}_i - \Gamma_i^E \Delta E_i,
\end{aligned} \tag{A.4}$$

where the first equation is due to the equivalence of (A.1) and (A.2), the following inequality is due to the weak duality between (A.2) and (A.3), and the second inequality is due to the feasibility of $(\hat{\mathbf{q}}^*, T^*, \mathbf{p}^*, \omega^*, \nu^*)$ in (13). This shows that $(\hat{\mathbf{q}}^*, T^*)$ is feasible to (11), implying that $T^* \leq T^\circ$.

Next, set $\hat{\mathbf{q}} = \hat{\mathbf{q}}^\circ$ in (A.2) and (A.3), and let $(z_{ij}^\circ, \lambda_{ji}^\circ)$ be an optimal solution to (A.2). By strong duality, there exists a feasible $(p_i^\circ, \omega_{ij}^\circ, \nu_{ji}^\circ)$ to (A.3) such that for all $i \in \mathcal{N} \setminus \mathcal{D}$ we have

$$\begin{aligned}
\overline{E}_i - \Gamma_i^E \Delta E_i & \geq \max_{e_{ij}^t, e_{ji}^r \in \mathcal{R}_i(\Gamma_i^e)} \left\{ \sum_{j \in \mathcal{S}_i} e_{ij}^t \hat{q}_{ij}^\circ + \sum_{j:i \in \mathcal{S}_j} e_{ji}^r \hat{q}_{ji}^\circ \right\} \\
& = \sum_{j \in \mathcal{S}_i} \overline{e}_{ij}^t \hat{q}_{ij}^\circ + \sum_{j \in J_i^t} \Delta e_{ij}^t \hat{q}_{ij}^\circ z_{ij}^\circ \\
& \quad + \sum_{j|i \in \mathcal{S}_j} \overline{e}_{ji}^r \hat{q}_{ji}^\circ + \sum_{j \in J_i^r} \Delta e_{ji}^r \hat{q}_{ji}^\circ \lambda_{ji}^\circ \\
& = \sum_{j \in \mathcal{S}_i} \overline{e}_{ij}^t \hat{q}_{ij}^\circ + \sum_{j|i \in \mathcal{S}_j} \overline{e}_{ji}^r \hat{q}_{ji}^\circ + \Gamma_i^e p_i^\circ \\
& \quad + \sum_{j \in J_i^t} \omega_{ij}^\circ + \sum_{j \in J_i^r} \nu_{ji}^\circ.
\end{aligned} \tag{A.5}$$

Thus, $(\hat{\mathbf{q}}^\circ, T^\circ, \mathbf{p}^\circ, \omega^\circ, \nu^\circ)$ satisfies the second set of constraints of (13). Since the remaining constraints are also satisfied, $(\hat{\mathbf{q}}^\circ, T^\circ, \mathbf{p}^\circ, \omega^\circ, \nu^\circ)$ is a feasible solution of (13), hence, $T^\circ \leq T^*$.

Acknowledgments

Research was partially supported by the NSF under Grant no. EFRI-0735974, by the DOE under Grant no. DE-FG52-06NA27490, by the ARO under Grant no. W911NF-09-1-0492, and by the ODDR&E MURI10 program under Grant no. N00014-10-1-0952.

References

[1] S. Ray, W. Lai, and I. C. Paschalidis, "Statistical location detection with sensor networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2670–2683, 2006.

[2] D. J. Baker and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm, communications," *IEEE Transactions on Communications*, vol. 29, no. 11, pp. 1694–1701, 1981.

[3] A. Michail and A. Ephremides, "A distributed routing algorithm for supporting connection-oriented service in wireless networks with time-varying connectivity," in *Proceedings of the 3rd IEEE International Symposium on Communications and Control*, 1998.

[4] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian, "Power-aware routing in wireless packet networks, Mobile Multimedia Communications," in *Proceedings of the IEEE InternationalWorkshop (MoMuC '99)*, pp. 380–383, 1999.

[5] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, 1999.

[6] L. Li and J. Y. Halpern, "Minimum-energy mobile wireless networks revisited," in *Proceedings of the International Conference on Communications (ICC '01)*, pp. 278–283, June 2001.

[7] R. Wattenhofer, L. Li, P. Bahl, and Y. M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1388–1397, April 2001.

[8] L. Li, J. Y. Halpern, P. Bahl, Y. M. Wang, and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," in *Proceedings of the 20th Annual ACM Symposium on Principles of Distributed Computing*, pp. 264–273, August 2001.

[9] C. K. Toh, H. Cobb, and D. A. Scott, "Performance evaluation of battery-life-aware routing schemes for wireless ad hoc networks," in *Proceedings of the International Conference on Communications (ICC '01)*, pp. 2824–2829, June 2001.

[10] J. Aslam, Q. Li, and D. Rus, "Three power-aware routing algorithms for sensor networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 187–208, 2003.

[11] W. Lai and I. C. Paschalidis, "Optimally balancing energy consumption versus latency in sensor network routing," *ACM Transactions on Sensor Networks*, vol. 4, no. 4, article 21, 2008.

[12] J. H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, 2004.

[13] A. Giridhar and P. R. Kumar, "Maximizing the functional lifetime of sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 5–12, April 2005.

[14] H. Nama and N. Mandayam, "Sensor networks over information fields: optimal energy and node distributions," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '05)*, vol. 3, pp. 1842–1847, March 2005.

[15] X. Wu and C. G. Cassandras, "A maximum time optimal control approach to routing in sensor networks," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC '05)*, pp. 1137–1142, December 2005.

[16] D. Bertsimas and M. Sim, "The price of robustness," *Operations Research*, vol. 52, no. 1, pp. 35–53, 2004.

[17] I. C. Paschalidis and S.-C. Kang, "Robust linear optimization: on the benefits of distributional information and applications in inventory control," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC '05)*, pp. 4416–4421, esp, December 2005.

[18] A. L. Soyster, "Convex programming with set-inclusive constraints and applications to inexact linear programming," *Operations Research*, vol. 21, pp. 1154–1157, 1973.

[19] A. Ben-Tal and A. Nemirovski, "Robust solutions of uncertain linear programs," *Operations Research Letters*, vol. 25, no. 1, pp. 1–13, 1999.

[20] P. Jacquet, "Geometry of information propagation in massively dense ad hoc networks," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MoBiHoc '04)*, pp. 157–162, May 2004.

[21] S. Toumpis and L. Tassiulas, "Optimal deployment of large wireless sensor networks," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2935–2953, 2006.

[22] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 188–200, November 2004.

[23] R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 8, pp. 2185–2193, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 568240, 8 pages
doi:10.1155/2012/568240

Research Article

A Wireless Sensor Network System with a Jumping Node for Unfriendly Environments

Jun Zhang, Guangming Song, Guifang Qiao, Zhen Li, and Aimin Wang

School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

Correspondence should be addressed to Guangming Song, mikesong@seu.edu.cn

Received 24 December 2011; Accepted 16 May 2012

Academic Editor: Jiming Chen

Mobile robots have been adopted to repair failed wireless sensor network systems for node damage, battery exhaustion, or obstacles. But most of the robots use wheeled locomotion manner, which does not work well or even fails when confronted with obstacles in uneven terrains. To solve this problem, this paper presents the design of a jumping robot to serve as a robotic node for wireless sensor networks. The robot can jump up to or over obstacles to repair the broken network connections. The robot senses its posture angle by using an acceleration sensor and self-rights automatically by using a pole leg after falling down on the ground. The robot also can steer and adjust its take-off angle by the pole leg. A network monitoring system with the proposed robot is built to test its basic locomotion capabilities and the network repair function. Experimental results show that the robot can jump about 90 cm in height and traverse 50 cm far at a take-off angle of 75 degrees. The robot can repair the network by jumping up to a 10 cm high platform. The proposed system with a jumping node can provide powerful support for applications in unfriendly environments.

1. Introduction

With the development of microelectronics and sensor technology, wireless sensor networks technology is widely used in many fields such as environment monitoring [1], home automation [2], industrial control [3], and military application [4]. But network deployment and network repair are difficult tasks when the working environments are dangerous or even cannot be reached by humans. This problem can be solved by mobile sensor networks technology.

Mobile sensor network systems usually use mobile robots as sensor nodes to execute network deployment and monitoring tasks [5]. But mobile sensor networks need lots of mobile robots. The costs increase with the increase of the number of mobile sensor nodes. So it is a feasible way to combine a large number of cheap static sensor nodes with some mobile sensor nodes. It is efficient to deploy the static sensor nodes and mobile nodes cooperatively. The mobile nodes are able to enlarge coverage range and enhance connectivity of the network. They also can repair the network when there are disabled static sensor nodes.

Traditional mobile robots use wheeled locomotion manner. In [6, 7] wheeled robots are used in WSN systems to detect intruders in indoor environments. In [8], the authors use a wheeled mobile robot as sink node to cooperate with other static sensor nodes and form a mesh network. Wheeled robots have always been confronted with the problem of locomotion in uneven terrains with many obstacles. If obstacles are higher than the radius of their wheels, wheeled robots cannot overcome obstacles efficiently.

Researchers are beginning to study creatures and their locomotion manners to design robots with more efficient locomotion capabilities. For example, the bipedal robot in [9] based on studying human locomotion, the quadruped walking robot Little Dog in [10] based on studying four-legged animals, and the hexapod robot RHex in [11] based on studying six-legged insects. In [12], a flying robot is used to deploy and repair a sensor network.

In the nature, many creatures have the capabilities of jumping locomotion such as frogs, locusts, and kangaroos. These inspire researchers to design some robots with jumping capabilities such as the Jollbot in [13], the JPL Hopper

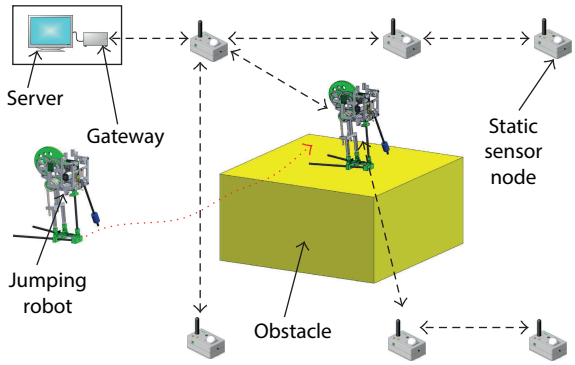


Figure 1: Architecture of the proposed wireless sensor network system.

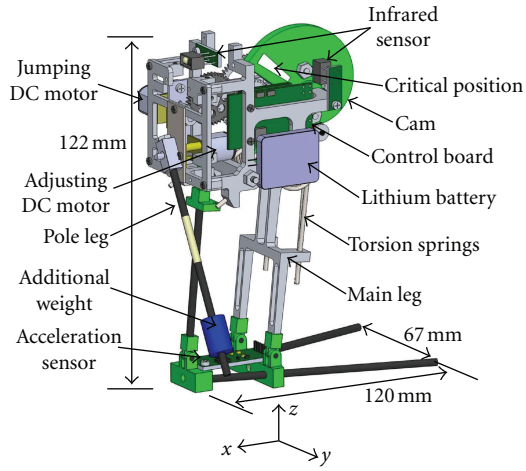


Figure 2: The 3D model of the jumping robot.

(generation II) in [14], the Mini-Whegs in [15], the Grillo in [16], the EPFL jumping robot in [17], and the MSU robot in [18]. With the capabilities of quickly overcoming obstacles and avoiding risks, jumping robots can be applied in many fields such as planet exploring [14], search and rescue [19], surveillance operations, and scouting [20]. These robots mounted with sensors and wireless communication devices are able to enter into dangerous and unfriendly environments to execute their missions.

Jumping robotic nodes also can cooperate with wheeled robotic nodes to monitor environments more flexibly than traditional mobile sensor networks only with wheeled mobile nodes. In [21], a wheeled Pioneer 2-AT robot is used to carry a team of small jumping robots. This is a typical application combining jumping robots with wheeled robots. Jumping robotic nodes are platforms carrying wireless communication modules in WSN systems. The communication modules can be selected according to the application requirements from short- to long-range communication techniques.

In this paper we design a jumping robot to serve as a robotic node for wireless sensor network systems. The short-range wireless communication technology of ZigBee is selected as the communication protocols. The robot is able to repair the network when one sensor node fails from battery exhaustion or obstacles. The system architecture is introduced in Section 2. The system design and working principles are presented in Section 3. The testbed setup and several experiments are given in Section 4. Section 5 will conclude the presented work, give some discussions, and introduce the future work.

2. System Overview

The architecture of the WSN system is shown in Figure 1. It is composed of a server, a gateway, six static sensor nodes, and a jumping robot. The server runs software for saving sensor data and controlling the system. The gateway connects with the server. The gateway builds a ZigBee network for data collection. The static sensor nodes can be deployed by humans. If the working environment is dangerous or even cannot be reached by humans, then Unmanned Aerial Vehicle (UAV) can be used to deploy static sensor nodes. The static sensor nodes join into the network to form a multihop communication network.

The static sensor nodes and the jumping robot detect temperature, humidity, light, and other information of the environment. These nodes and the robot send sensor data to the gateway. The gateway receives sensor data and transmits them to the server. The gateway also can analyze and calculate the signal intensity of every sensor node and the robot. If one sensor node does not send sensor data to the gateway or its signal intensity is too weak, the jumping robot can move to the position of the sensor node to replace it. The network will be repaired by the robot when the network interrupts because one routing node is disabled or sheltered by obstacles.

3. System Design and Working Principles

3.1. Jumping Robot Design. The CAD model of the proposed jumping robot is shown in Figure 2. It is 120 mm × 67 mm × 122 mm in size. The robot is composed of a body frame, a jumping mechanism, an adjusting mechanism, a three-axis acceleration sensor, two infrared sensors, a control board, and a lithium battery.

The jumping mechanism consists of a DC motor, a reduction gear box, a cam, a main leg, and four torsion springs. Inspired by the sudden jumping locomotion of locusts, torsion springs are selected as the energy storage components. The torsion springs are installed between the main leg and the body frame of the robot. There is a bearing on the main leg, which is tangential to the cam. The DC motor with the reduction gear mechanism can obtain high torque to drive the cam to rotate. The cam compresses the torsion springs to store elastic potential energy. The contour shape of the cam is specially designed with quick-return characteristics. It allows sudden release of the elastic potential energy to drive the robot to take off. The detailed mechanical design work is presented in [22].

The adjusting mechanism is composed of a DC motor, a pole leg, and an additional weight. The robot always falls

Table 1: The three-axis acceleration changes according to state transition of the jumping robot when self-righting.

Direction	x	y	z
Left side	0 → 0	Max → 0	0 → Max
Right side	0 → 0	Min → 0	0 → Max
Front side	Max → 0	0 → Max → 0	0 → Max

down on its left, right, or front side after landing on the ground. The automatic self-righting principle is as follows: when the jumping robot falls down on one of the three sides, the acceleration sensor can provide posture angle information for the robot. The pole leg rotates and the robot body will be propped up. The robot detects its posture angle periodically. When standing up, the robot will stop rotating the pole leg and start rotating it in the opposite direction to fold the pole leg up. The infrared sensor in the front part of the body frame will detect passing by of the pole leg. When the sensor finds the pole leg, the microprogrammed control unit (MCU) on the control board will control the pole leg to stop rotating.

The acceleration sensor is mounted on the foot of the robot. The surface of the sensor is parallel with the XY plane as shown in Figure 2. In theory, the three-axis acceleration values change with variations of the posture angle when the robot is self-righting as shown in Table 1. When the robot falls down on its left side the accelerations of the three axes are $x = 0$, $y =$ Max, and $z = 0$. When the robot falls down on its right side, $x = 0$, $y =$ Min, and $z = 0$. When the robot falls down on its front side, $x =$ Max, $y = 0$, and $z = 0$. These characteristics are used by the robot to judge its initial states when it is falling down. After self-righting, $x = 0$, $y = 0$, and $z =$ Max, this can be used by the robot to judge its righting state.

During self-righting from left side, y decreases from Max to 0, z increases from 0 to Max, and x is 0 all the time. During self-righting from right side, y increases from Min to 0, z increases from 0 to Max, and x is also 0 all the while. During self-righting from front side, and x decreases from Max to 0, y increases from 0 to Max and then decreases to 0 again, and z increases from 0 to Max.

Changes of the robot posture angle during self-righting are used to design self-righting control algorithm. The automatic self-righting algorithm is shown in Figure 3. After parameter initialization, the MCU samples sensor data periodically and judges the acceleration z (AZ) to see if it is larger than the threshold T_{Max}. If this is true, the robot will judge the absolute value of y abs (AY). If it is larger than the threshold T_{Min}, the robot will know it falls down on its left or right side. If AY > 0, the robot will consider that it has fallen down on its left side and start to rotate adjusting motor clockwise. If AY < 0, the robot will consider that it has fallen down on its right side and start to rotate adjusting motor anticlockwise. If abs (AY) $< T_{\text{Min}}$, or the robot finishes one step of self-righting, the robot will judge acceleration x (AX) to see if it is larger than T_{Min}, If it is true, the robot considers that it falls down on its front side. Then the jumping motor starts to rotate until the infrared sensor in the left part of

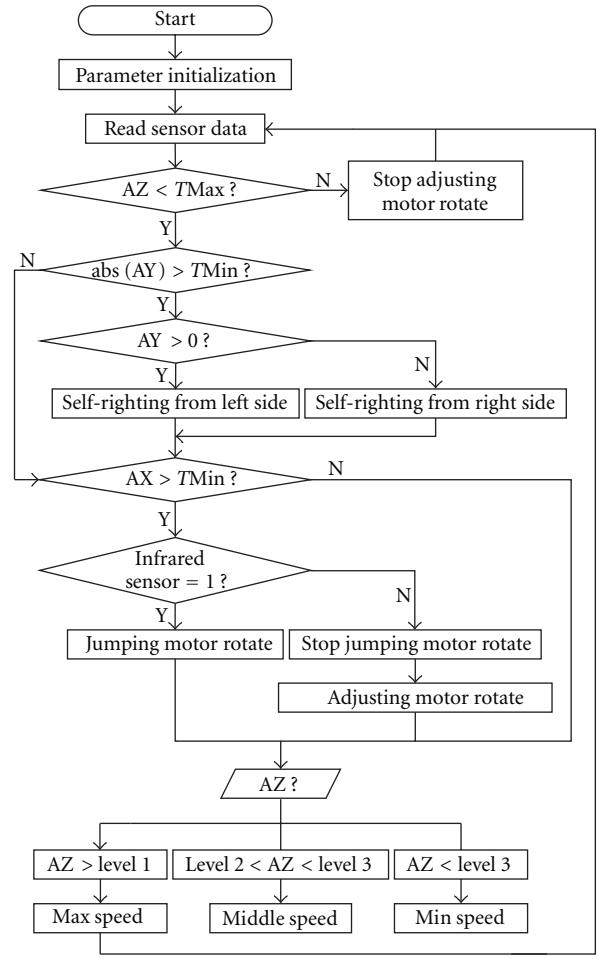


Figure 3: The automatic self-righting algorithm of the jumping robot.

the body frame has detected the critical position of the cam. Then the adjusting motor rotates counterclockwise to make the robot shift to the state that its right side contacts the ground. Then the robot will stand up using the same method as it falls down on its right side. The robot needs higher speed when self-righting in the beginning than in the end. The robot adjusts the speed of adjusting motor according to AZ. The three levels in Figure 3 are decided by posture angles of the robot.

The adjusting mechanism also can adjust jumping direction and take-off angle. The length of the pole leg is larger than the distance between the axis of the adjusting motor and the ground. The steering principle is that the robot can rotate the pole leg to steer after self-righting. The pole leg rotates to contact and leave the ground. The front part of the robot is propped up and moves a step in the ground while the foot of the robot is still in the same position. After this motion process, the robot can steer an angle in the ground. This angle also can be adjusted by adjusting the rotation of the jumping motor. The stable and efficient steering needs cooperation between the adjusting motor and the jumping motor.

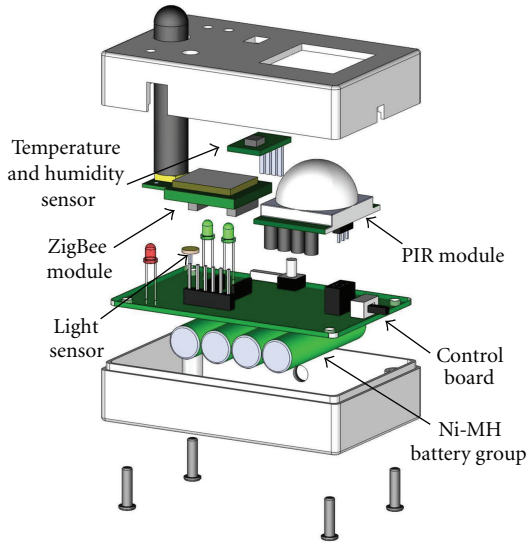


Figure 4: CAD model of the static sensor node.

Figure 5: Prototype of the sensor node.

An additional weight (AW) is mounted at the end of the pole leg to be used for take-off angle adjusting. After self-righting and steering, the robot adjusts the pole leg in different heights to get different center of mass (COM) positions of the robot. Different COM heights correspond to different take-off angles. So the take-off angle can be adjusted by rotating the pole leg in different positions.

3.2. Static Sensor Node Design. The CAD model of the static sensor node is shown in Figure 4. It is composed of a temperature & humidity sensor module, a light sensor, a pyroelectric infrared (PIR) sensor module, a ZigBee wireless communication module, a control board, and a Ni-MH battery group. The prototype of the static sensor node is shown in Figure 5. The static sensor nodes are used to detect whether there is a person passing by them and sense the temperature, humidity, and light of their working environment. These sensor data will be transmitted to the gateway by the ZigBee module. The nodes also periodically sample the voltage themselves. When finding the voltage lower than the preset threshold, the ZigBee module will send this message to the gateway. The gateway will control the jumping robot to take the place of the static node which will fail to work in the near future.

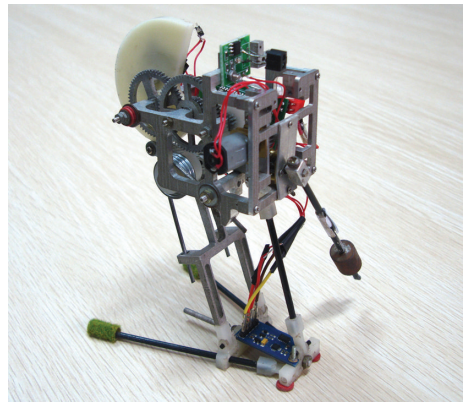


Figure 6: Prototype of the proposed jumping robot.

3.3. Network Repair Principle. The ground and other objects can reflect and absorb signal of wireless communication especially when sensor nodes with small size are deployed in uneven terrains of outdoor environments. The signal intensity of wireless communication is influenced by the deployment height of sensor nodes. Jumping robot can jump up to a relative high position to route sensor data. This will enhance communication quality of networks.

The network repair mentioned here supposes that the positions of the static sensor nodes and the initial position of the robot are known. When one sensor node is disabled or sheltered by obstacles, this may affect communication between the node and the gateway, and even result in interruption of the network. When the gateway finds that one sensor node does not send environment data continually to it, the gateway will send the inquiring command to the sensor node. If the sensor node will not acknowledge the command, the gateway will consider that the sensor node is disabled. When this happens, a jumping robot can be controlled by the gateway to move to the position of the disabled sensor node. The jumping robot then replaces the sensor node to monitor this area and route data of other sensor nodes to the gateway.

4. Experiment

A testbed is built to test the jumping and automatic self-righting capabilities of the proposed jumping robot. The steering and take-off angle adjusting functions are controlled by the operator through the graphic user interface running on the server. Two network interruption scenarios are set by the operator. The network repair functions of the WSN system in the two scenarios are tested.

4.1. Jumping Test. The prototype of the proposed jumping robot is shown in Figure 6. The jumping height of this prototype can be adjusted by installing two or four torsion springs. In this test four springs are used to get a maximum

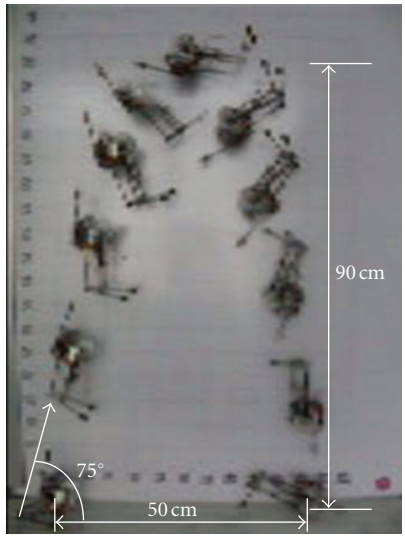


Figure 7: Jumping trajectory of the prototype robot.

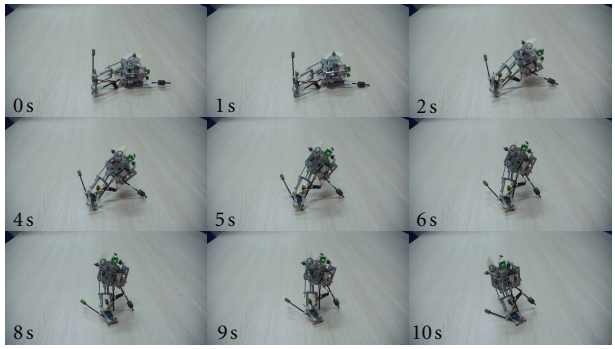


Figure 8: The automatic self-righting sequences of the jumping robot from left side.

jumping height. The torque of the motor is 93 N·mm. The total mass of the robot is 154 g. The jumping trajectories of the prototype robot have been recorded as shown in Figure 7. It can jump about 90 cm in height and traverse 50 cm far at a take-off angle of 75°. This test verifies that the jumping robot has powerful obstacle overcoming capabilities.

4.2. Automatic Self-Righting Test. In this test, the robot uses the control algorithm presented in Section 3.1 to stand up automatically after falling down. The result of self-righting sequences from left side is shown in Figure 8. The robot self-rights step by step. The changes of three-axis acceleration values during the self-righting process are shown in Figure 9. In the beginning, the falling on the left side is judged by the condition that acceleration z is smaller than its threshold 900 mm/s^2 and acceleration y is bigger than its threshold 300 mm/s^2. During the 10 s self-righting, y is decreasing, z is increasing, and x is not changing evidently. In the end, y is smaller than its threshold, and z is bigger than its threshold. The robot stands up successfully.

The self-righting sequences from right side are shown in Figure 10. The accelerations changes during the self-righting

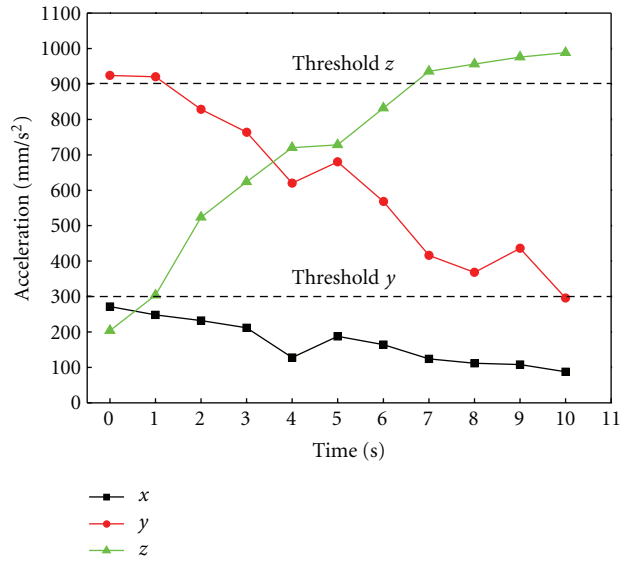


Figure 9: Acceleration changes during the self-righting process from left side.

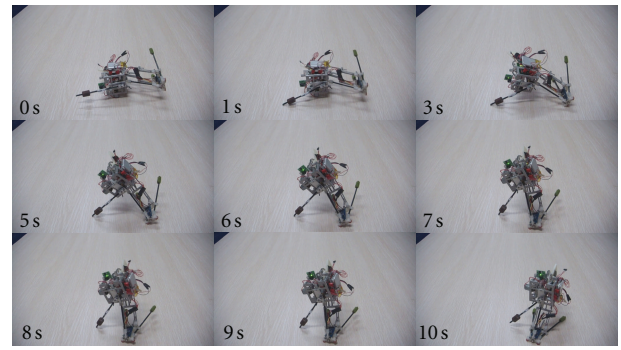


Figure 10: The automatic self-righting sequences of the jumping robot from right side.

process are shown in Figure 11. In the beginning, the falling on the right side is judged by the condition that acceleration z is smaller than its threshold 900 mm/s^2 and acceleration y is smaller than its threshold −300 mm/s^2. During self-righting, y and z are increasing and x is not changing evidently. In the end, y and z are bigger than their thresholds. The robot stands up successfully.

The self-righting sequences from front side are shown in Figure 12. The acceleration changes are shown in Figure 13. In the beginning, the falling on the front side is judged by the condition that acceleration z is smaller than its threshold 900 mm/s^2 and acceleration x is bigger than its threshold 300 mm/s^2. Firstly, the jumping motor rotates to make the robot shift from frontside to rightside. In 20 s, the robot shifts to right side. The acceleration of x changes from about 800 mm/s^2 to 200 mm/s^2. The acceleration of y changes from about −200 mm/s^2 to −1000 mm/s^2. The acceleration z is still smaller than its threshold. Then, the robot judges this shift costing about 20 s and starts to self-right from right side

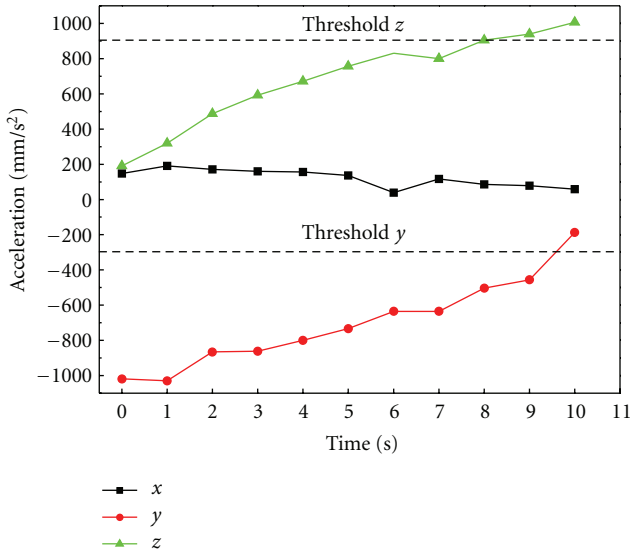


Figure 11: Accelerations changes during the self-righting process from right side.

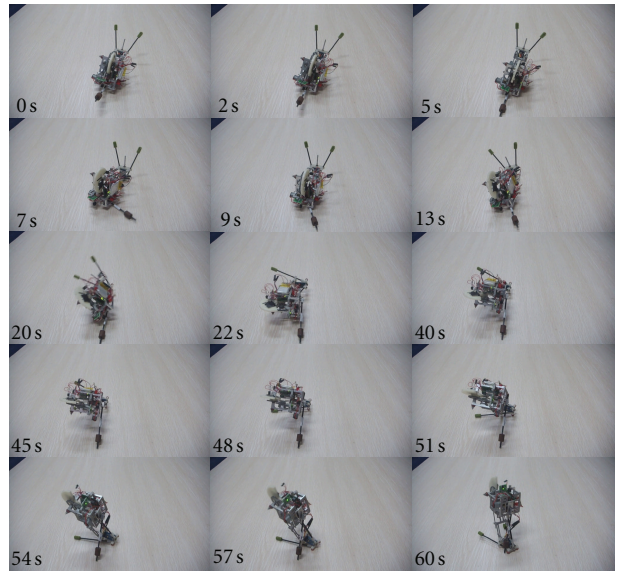


Figure 12: The automatic self-righting sequences of the jumping robot from front side.

in 40 s. In the end, y and z are bigger than their thresholds. The robot stands up successfully.

Self-righting of all the three sides are tested 40 times. The self-righting success rates are 82.50% from left side, 97.50% from right side, and 87.50% from front side. The reason of the difference between left and right sides is that the COM of the robot is not symmetrical in left and right sides. This can be solved by adjusting the position of the lithium battery. The success rate of the front side is also low for the strong vibration of the shift from front side to right side. This problem can be solved by optimizing the control algorithm.

4.3. Network Repair Test. The testbed setup of network repair experiment is shown in Figure 14. The network interruption is caused by obstacle. Because the distance between every two sensor nodes is in one-hop range, we set the max children node number of the gateway and sensor nodes to 1 and takeoff the 3 dB antenna of sensor nodes to implement mandatory multi-hop transmission. The gateway is powered up first and the sensor nodes are powered up one by one to ensure that a multihop chain network topology can be established.

We can see sensor data from every sensor node in the graphic user interface running on the server. The link quality indicator (LQI) of every node is recorded in the database. The results are shown in Figure 15. A 10 cm high obstacle is deployed between node 4 and nodes 5 and 6 in 30 s. The sensor data of nodes 5 and 6 cannot be received because they are sheltered by the obstacle. Then the robot is controlled by the gateway to jump up to the obstacle to repair the network. After about 50 s, sensor data of nodes 5 and 6 are seen in the graphic user interface again.

The testbed setup of network repair experiment is shown in Figure 16. The network interruption is caused by turning off one routing node. The sensor nodes are deployed in a line. Two desk lamps are used to make light sensor data of every sensor node different. The light sensor data during network repair process are recorded. The results are shown in Figure 17. Node 4 was turned off by the operator in 90 s. Then we can see in the graphic user interface that nodes 4, 5, and 6 do not send data to the gateway. The robot is powered up and controlled by the gateway to move to the position of node 4. After joining into the network, the robot sends its sensor data to the gateway. The gateway records the LQI value of the robot. The robot jumped to the position of node 4 to repair the network. When the robot reached the position of

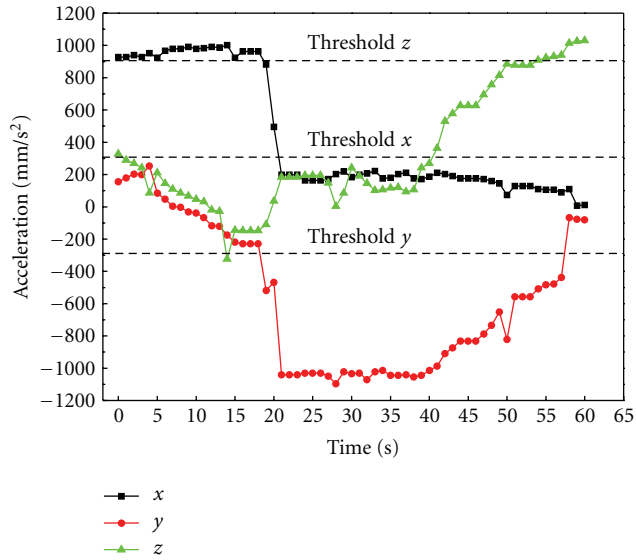


Figure 13: Accelerations changes during the self-righting process from front side.

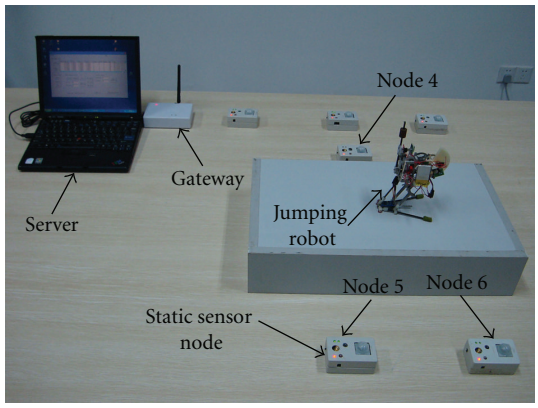


Figure 14: Testbed setup of the network repair experiment. The network interruption is caused by a cubic obstacle.

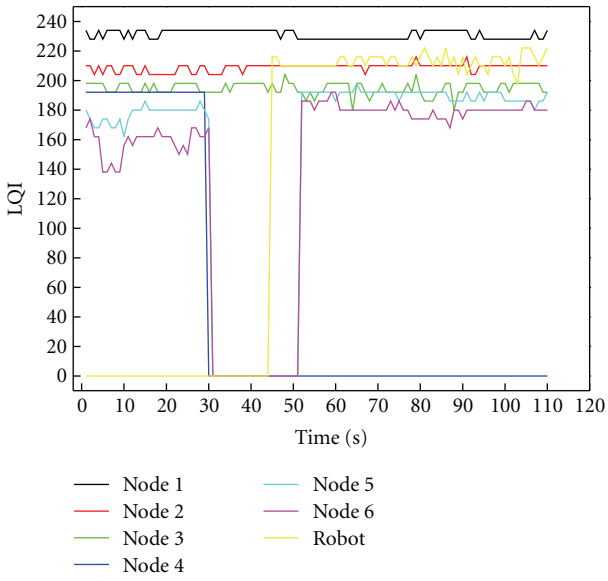


Figure 15: Result of network repair test. The network interruption is caused by obstacle sheltered.

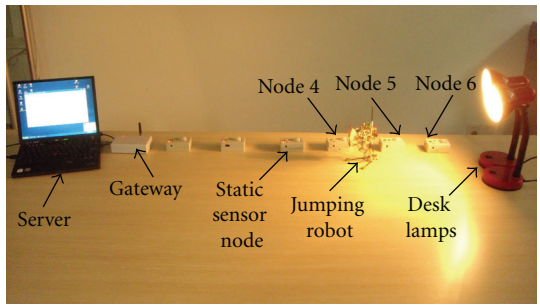


Figure 16: Testbed setup of network repair experiment. The network interruption is caused by turning off one routing node.

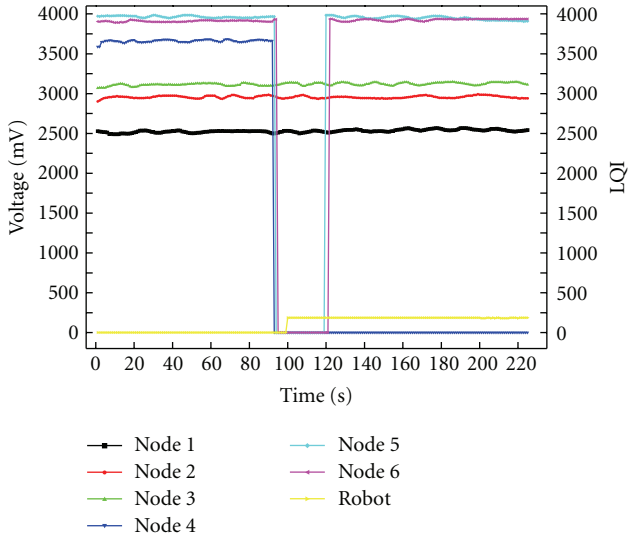


Figure 17: Result of network repair test. The network interruption is caused by turning off one routing sensor node.

node 4, then nodes 5 and 6 restarted to send data to gateway by routing through the robot.

5. Conclusions

We have presented a WSN system with a jumping robot as the mobile node. The robot uses the cam with quick-return characteristic to save and release energy. A pole leg is used to self-right after falling down. The pole leg also can be used to adjust the jumping direction and take-off angle of the robot. Based on the jumping robot design and some static sensor node design, a WSN monitoring system is built to test the functions of the robot and the function of the network repair. Results show that the jumping robot can jump 90 cm in height and traverse 50 cm far. The robot is able to automatically self-right after falling down on the ground. The steering and take-off angle adjusting of the robot can be controlled by the operator through the graphic user interface running on the server. The robot can be controlled to move to the position of the disabled static sensor node to repair the network and route sensor data.

The self-righting capability of the jumping robot is not very robust. It only can work in relatively smooth surfaces. It is difficult for this generation of the robot prototype to keep standing in uneven terrains. The pole leg of the robot will be designed to provide powerful support for the robot in the next generation. The total energy saved in the torsion springs is fixed during every jumping. The take-off angle adjusting method in this generation also has limitations. Changeable energy storage solutions and wide range of take-off angles adjusting methods will be investigated in the future to make the robot jump according to the height of obstacles for the purpose of energy saving.

Future work will focus on the following three aspects. (1) We plan to add a compass to sense the current direction of the robot in order to achieve the function of automatic steering. (2) A code wheel will be pasted on the surface of the cam to make steering and take-off angle adjusting more precise. (3) The solutions of localization and trajectory

planning for continuous jumping of our robot will be investigated. Therefore the prospective multijumping robots will be widely used in WSN systems.

Acknowledgments

The research reported in this paper was carried out at the Robotic Sensor and Control Lab, School of Instrument Science and Engineering, Southeast University, Nanjing, Jiangsu, China. This work was supported in part by the Scientific Research Foundation of Graduate School of Southeast University under Grant YBJJ-1221, the Research & Innovation Program for Graduate Student in Universities of Jiangsu Province under Grant CXLX-0102, Program for New Century Excellent Talents in University under Grant NCET-10-0330, and Natural Science Foundation of Jiangsu Province under Grant BK2011254.

References

[1] P. Jiang, H. Xia, Z. He, and Z. Wang, "Design of a water environment monitoring system based on wireless sensor networks," *Sensors*, vol. 9, no. 8, pp. 6411–6434, 2009.

[2] G. Song, Y. Zhou, W. Zhang, and A. Song, "A multi-interface gateway architecture for home automation networks," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1110–1113, 2008.

[3] C. A. Boano, N. Tsiftes, T. Voigt, J. Brown, and U. Roedig, "The impact of temperature on outdoor industrial sensornet applications," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 451–459, 2010.

[4] M. A. Hussain, P. Khan, and K. K. Sup, "WSN research activities for military application," in *Proceedings of 11th International Conference on Advanced Communication Technology (ICACT '09)*, pp. 271–274, Phoenix Park, Republic of Korea, February 2009.

[5] G. Song, Y. Zhou, Z. Wei, and A. Song, "A smart node architecture for adding mobility to wireless sensor networks," *Sensors and Actuators A*, vol. 147, no. 1, pp. 216–221, 2008.

[6] C. H. Lin, S. H. Yang, H. T. Chen, and K. T. Song, "Mobile robot intruder detection based on a zigbee sensor network," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC '08)*, pp. 2786–2791, Singapore, October 2008.

[7] Y. Li and L. E. Parker, "Detecting and monitoring time-related abnormal events using a wireless sensor network and mobile robot," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS' 08)*, pp. 3292–3298, Nice, France, September 2008.

[8] J. Takahashi, T. Yamaguchi, K. Sekiyama, and T. Fukuda, "Communication timing control and topology reconfiguration of a sink-free meshed sensor network with mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 187–197, 2009.

[9] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[10] F. Doshi, E. Brunskill, A. Shkolnik et al., "Collision detection in legged locomotion using supervised learning," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 317–322, San Diego, Calif, USA, November 2007.

[11] E. Sayginer, T. Akbey, Y. Yazicioglu, and A. Saranli, "Task oriented kinematic analysis for a legged robot with half-circular leg morphology," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 4088–4093, May 2009.

[12] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3602–3608, New Orleans, LA, USA, May 2004.

[13] R. Armour, K. Paskins, A. Bowyer, J. Vincent, and W. Megill, "Jumping robots: A biomimetic solution to locomotion across rough terrain," *Bioinspiration and Biomimetics*, vol. 2, no. 3, article S01, pp. S65–S82, 2007.

[14] J. Burdick and P. Fiorini, "Minimalist jumping robots for celestial exploration," *International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 653–674, 2003.

[15] B. G. A. Lambrecht, A. D. Horchler, and R. D. Quinn, "A small, insect-inspired robot that runs and jumps," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1240–1245, Barcelona, Spain, April 2005.

[16] U. Scarfogliero, C. Stefanini, and P. Dario, "The use of compliant joints and elastic energy storage in bio-inspired legged robots," *Mechanism and Machine Theory*, vol. 44, no. 3, pp. 580–590, 2009.

[17] H. Tsukagoshi, M. Sasaki, A. Kitagawa, and T. Tanaka, "Design of a higher jumping rescue robot with the optimized pneumatic drive," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1276–1283, Barcelona, Spain, April 2005.

[18] J. Zhao, N. Xi, B. Gao, M. Mutka, and L. Xiao, "Development of a controllable and continuous jumping robot," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4614–4619, Shanghai, China, 2011.

[19] S. A. Stoeter and N. Papanikolopoulos, "Kinematic motion model for jumping scout robots," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 398–403, 2006.

[20] M. Kovač, M. Schlegel, J. C. Zufferey, and D. Floreano, "Steerable miniature jumping robot," *Autonomous Robots*, vol. 28, no. 3, pp. 295–306, 2010.

[21] E. Kadioglu and N. Papanikolopoulos, "A Method for transporting a team of miniature robots," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2297–2302, Las Vegas, Nev, USA, October 2003.

[22] J. Zhang, Y. Zhu, H. Wang, and J. Zhang, "Design of a bio-inspired jumping robot for rough terrain," in *Proceedings of the 2nd Conference on Environmental Science and Information Application Technology (ESIAT '10)*, pp. 40–43, Wu Han, China, July 2010.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 842107, 15 pages
doi:10.1155/2012/842107

Research Article

Mobile Sensor Waypoint Navigation via RF-Based Angle of Arrival Localization

Isaac Amundson, Janos Sallai, Xenofon Koutsoukos, and Akos Ledeczi

Department of Electrical Engineering and Computer Science, Institute for Software Integrated Systems (ISIS), Vanderbilt University, Nashville, TN 37235, USA

Correspondence should be addressed to Isaac Amundson, iamundson@gmail.com

Received 25 November 2011; Revised 23 April 2012; Accepted 3 May 2012

Academic Editor: Jiming Chen

Over the past decade, wireless sensor networks have advanced in terms of hardware design, communication protocols, and resource efficiency. Recently, there has been growing interest in mobility, and several small-profile sensing devices that control their own movement have been developed. Unfortunately, resource constraints inhibit the use of traditional navigation methods because these typically require bulky, expensive sensors, substantial memory, and a generous power supply. Therefore, alternative navigation techniques are required. In this paper, we present a navigation system implemented entirely on resource-constrained sensors. Localization is realized using triangulation in conjunction with radio interferometric angle-of-arrival estimation. A digital compass is employed to keep the mobile node on the desired trajectory. We also present a variation of the approach that uses a Kalman filter to estimate heading without using the compass. We demonstrate that a resource-constrained mobile sensor can accurately perform waypoint navigation with an average position error of 0.95 m.

1. Introduction

Typically, autonomous navigation is performed by robots equipped with cameras, laser rangefinders, sonar arrays, and other sophisticated sensors for collecting range and bearing information. These sensor data are then used to compute spatial relationships such as position and proximity, which enable the robot to follow a given trajectory. However, these sensors are large, expensive, have considerable power requirements, and/or require a powerful computing platform to analyze sensor data. In recent years, mote-sized mobile sensor platforms have been developed, that are unable to use traditional navigation methods because of their small size and limited resources [1–5]. This emerging class of mobile sensor would greatly benefit from navigation techniques geared towards resource-constrained devices.

In order to enable navigation in mobile wireless sensor networks (MWSNs), we must develop new methods for estimating position and deriving motion vectors that are rapid and accurate in spite of the limited resources available. Localization in wireless sensor networks has been studied extensively, and several techniques exist that provide submeter accuracy. However, these techniques are often unacceptable for mobile sensor localization due to algorithm complexity and cost. For example, although GPS receivers are available for mote-scale devices, they are still relatively expensive [6]. The Cricket location-support system requires customized hardware with ultrasonic sensors [7]. Other techniques such as the radio interferometric positioning system (RIPS) do not require additional hardware support; however, localization latency is prohibitively high for mobile devices [8].

In this paper, we propose a localization and waypoint navigation system called TripNav, in which a mobile sensor node follows a path by navigating between position coordinates. Position estimates are obtained using a localization technique we developed that combines radio interferometric angle-of-arrival estimation [9] with least squares triangulation [10]. We use this approach because it provides rapid and accurate position estimates and runs on resource-constrained sensor nodes without the need for hardware modifications. These properties are desirable because they

enable such a system to be assembled and deployed quickly and inexpensively using commercial off-the-shelf (COTS) components.

Way-finding represents a major category of navigational behavior [11]. Simple waypoint navigation scenarios include automated transportation routes and sentries that patrol a path along the perimeter of a secure area. For our research, the mobile sensor node is provided with a target speed and a set of waypoints and is instructed to pass by each waypoint in the order they are given. The node is comprised of an XSM mote [12] mounted to an iRobot Create [13]. The Create is a programmable robot that hosts a small suite of sensors; however, we use it only as a mobile platform, and all localization and navigation control operations are performed on the attached mote. In addition, we employ a digital compass for estimating heading, from which we can calculate the heading error of the mobile sensor with respect to the desired trajectory. An alternate method is also presented, in which the digital compass is not required, and heading is estimated using an extended Kalman filter (EKF). A simple controller, implemented in software, is then used to derive the necessary wheel speeds for maintaining the correct heading.

In previous work [9], we presented a system for estimating the angle-of-arrival of an interference signal. The system is comprised entirely of COTS sensor nodes, it is completely distributed, bearing can be estimated rapidly, and no additional hardware is required. Our present research builds on this technique by estimating bearing to multiple anchors and then determining position using triangulation. Because the technique is rapid, it is appropriate for mobile devices, which must continuously update their position estimates for navigation. By implementing our angle-of-arrival technique on a mobile platform and using a simple waypoint navigation approach for determining motion vectors, we are able to satisfy the main criteria for a successful MWSN [14]. These criteria include (1) no hardware modifications, (2) submeter position accuracy, (3) rapid position estimation on the order of seconds, and (4) implementation on a resource-constrained system.

The contributions of this work are as follows:

(1) we describe TripNav, a lightweight localization and waypoint navigation system for resource-constrained mobile wireless sensor networks, and demonstrate that our localization method is indeed suitable for mobile sensor navigation;

(2) we perform error and timing analyses that show that location error, heading error, and latency do not significantly impact navigation;

(3) we provide simulation results that show how waypoint navigation using TripNav can be performed without employing a digital compass;

(4) we show experimentally that TripNav works reliably and has a trajectory error of less than one meter.

The remainder of this paper is organized as follows. In Section 2, we review other MWSN research that has recently appeared in the literature. In Section 3, we describe the radio interferometric positioning system and radio interferometric angle-of-arrival estimation, key components of our proposed navigation system. We then present the system design of our TripNav waypoint navigation method in Section 4. In Section 5, we analyze the main sources of TripNav error, as well as provide simulation results of the system performance using computational methods to estimate heading. We describe our real-world implementation in Section 6 and evaluate the performance of the system in Section 7. Finally, in Section 8, we conclude.

2. Related Work

To date, most mobile wireless sensor navigation applications deal with tracking a mobile embedded sensor (a mobile sensor that does not control its own movement) [15–17]. Tracking is the process of taking a series of measurements, and using that information to determine the history, current position, and potential future positions of the object. Tracking can be cooperative (i.e., the tracked object participates in its localization) or noncooperative. Mobile-actuated sensors, on the other hand, control their own movement. Navigation requires in-the-loop processing of location data to determine a motion vector that will keep the mobile entity on the desired trajectory. There are two main approaches for using mobile sensors for navigation: dead reckoning and reference-based [18].

Dead reckoning uses onboard sensors to determine the distance traveled over a designated time interval. Distance can be obtained using odometry via encoders or by inertial navigation techniques using accelerometers and gyroscopes. The main benefit of using dead reckoning systems is that no external infrastructure is required. Position can be inferred by integrating velocity, or doubly integrating acceleration, with respect to time; however, error will accrue unbounded unless the mobile node can periodically reset the error by using known reference positions.

In reference-based systems, mobile entities use landmarks in the region for correct positioning and orientation. Landmarks can be active beacons, such as sensor nodes and satellites, or physical structures, such as mountains and buildings. A common use for reference-based systems is model-matching, also referred to as mapping [18]. Mapping requires the ability to detect landmarks in the environment and match them to a representation of the environment that was obtained a priori and stored in the memory of the mobile device. For mobile robots, landmarks are typically detected using cameras. Landmarks do not need to be structural, however. Received signal strength (RSS) profiling is a type of model-matching technique, in which the observed signal strengths from multiple wireless access points are used to estimate position [19, 20]. Simultaneous localization and mapping (SLAM) is also a type of mapping, in which the mobile device builds a map of the environment at the same time as it determines its position [21]. Similarly, simultaneous localization and tracking (SLAT) is a technique to localize a mobile entity while keeping track of the path it has taken to arrive at its present position [22].

Most reference-based navigation techniques are difficult to implement on resource-constrained mobile sensors without increasing cost or modifying hardware. For example, in [4], position is determined using an overhead camera system. In many instances, the cost of the camera system alone can be higher than the rest of the sensor network, making this localization approach undesirable. Millibots use a combination of dead-reckoning and ultrasonic ranging [5]. Supporting ranging in this manner requires customized hardware with ultrasonic sensors, a feature typically not found on COTS sensor nodes. Another technique uses static sensors to guide the mobile sensor to a specific area [3, 23]; however, this approach can only achieve course-grained accuracy.

3. Background

3.1. The Radio Interferometric Positioning System. Our work is based on the Radio Interferometric Positioning System (RIPS), an RF-based localization method presented in [8]. RIPS was developed as a means for accurately determining the relative positions of sensor nodes over a wide area by only using the onboard radio hardware. It was originally implemented on the COTS Mica2 mote platform [24], which has a 7.4 MHz processor, 4 kB RAM, and a CC1000 tunable radio transceiver that operates in the 433 MHz range [25]. Although the radio hardware is quite versatile for its size and cost, 433 MHz is too high to analyze the received signal directly. Instead, RIPS employs transmitter pairs at close frequencies for generating an interference signal. The phase and frequency of the resulting beat signal can be measured by making successive reads of the received signal strength indicator (RSSI).

Figure 1 illustrates the approach. Two nodes, A and B, transmit pure sinusoids at respective frequencies f_A and f_B, such that $f_B < f_A$. The two signals interfere, resulting in a beat signal with frequency $|f_A - f_B|$. The phase difference between receiver pairs is a linear combination of the distances between the four participating nodes:

$$\Delta\varphi = \frac{2\pi}{\lambda}(d_{AC} - d_{BC} + d_{BD} - d_{AD}) \,(\mathrm{mod}\, 2\pi), \qquad (1)$$

where $\Delta\varphi$ is the phase difference, λ is the wavelength of the transmitted signal, and d_{AD}, d_{BD}, d_{BC}, and d_{AC} are the respective distances between node pairs (A, D), (B, D), (B, C), and (A, C).

The distance measurement $(d_{AC} - d_{BC} + d_{BD} - d_{AD})$ is referred to as a *quad-range*. Because phase wraps to 0 at 2π, an ambiguity exists, where an observed signal phase difference could correspond to several different quad-ranges. To resolve this, RIPS samples at multiple frequencies and searches for a unique quad-range that satisfies (1) for each measured phase difference and corresponding wavelength.

A single quad-range is not sufficient to determine the positions of the four nodes involved in the radio interferometric measurement. Instead, a genetic optimization algorithm is used that takes into consideration all participating nodes in the sensing region. The algorithm is able to simultaneously remove bad measurements while accurately estimating the position of the sensors. The quad-ranges between a sufficient number of participating nodes constrain each node to a unique position in the sensing region. RIPS was shown to have an accuracy of 3 cm at a range of up to 160 meters; however, it could take up to several minutes in large networks and thus is not suitable for localizing mobile nodes.

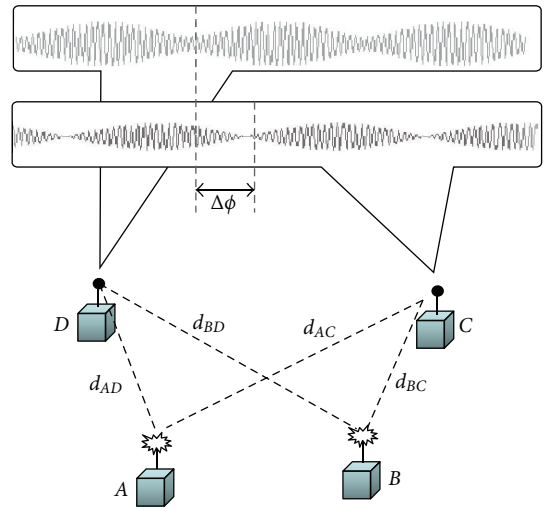


Figure 1: The radio interferometric positioning system.

In order to achieve accurate localization in wireless sensor networks, fine-grained resolution clock synchronization is required. RIPS employs the elapsed time on arrival (ETA) SyncEvent primitive [26], which provides synchronization with an accuracy on the order of microseconds. The SyncEvent primitive declares a time in the future to begin the clock synchronization process. A node that wishes to coordinate its clock with the clocks of several other nodes broadcasts a SyncEvent message. Encoded in the message is the timestamp of the message sender (typically the localization coordinator), which is inserted into the message immediately before transmission, thus reducing the amount of nondeterministic latency involved in the synchronization. All nodes within broadcast range will receive the message at approximately the same time instant, and assuming a negligible transit time of the radio signal through air, will be able to transform the sender timestamp into their local timescale with minimal synchronization error.

3.2. Radio Interferometric Angle of Arrival Estimation. In [9], we presented a rapid technique for determining bearing to a target node at an unknown position from a stationary anchor node. The technique uses the same radio interferometric method as RIPS, but takes less than a second to complete.

The system consists of stationary antenna arrays and cooperating target (possibly mobile) wireless sensor nodes. The array contains three nodes: a primary (P) and two assistants (A_1, A_2), as shown in Figure 2. At a predetermined time, the primary, P, and one of the assistants, A_1, transmit a pure sinusoidal signal at slightly different frequencies, which interfere to create a low-frequency beat signal whose phase is

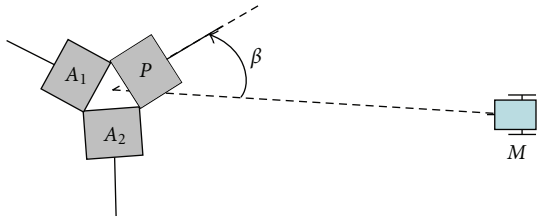


Figure 2: Array containing a primary node (P) and two assistant nodes (A_1, A_2). A target node (M) computes its bearing (β) from the array.

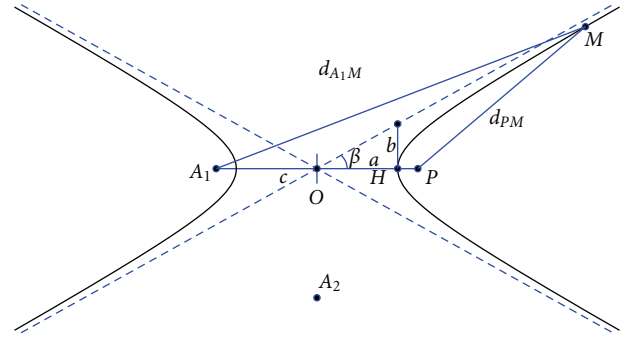


Figure 3: The t-range defines a hyperbola that intersects target node M, and whose asymptote passes through the midpoint of the two transmitters in the array, A_1 and P.

measured by the other assistant in the array, A_2, and a target node, M, at an unknown position. Such a measurement is termed a *radio interferometric measurement* (RIM).

The difference in phase measured by receiver nodes M and A_2 is a linear combination of the distances between the transmitters and receivers, and using (1), we have

$$\Delta\varphi = \frac{2\pi}{\lambda}\left(d_{PA_2} - d_{A_1A_2} + d_{A_1M} - d_{PM}\right) (\mathrm{mod}\, 2\pi), \quad (2)$$

where $\Delta\varphi$ is the phase difference, λ is the wavelength of the carrier frequency, d_{PM} is the distance between the primary node and mobile node, d_{A_1M} is the distance between the assistant transmitter and the mobile node, and d_{PA_1}, d_{PA_2}, and $d_{A_1A_2}$ are the respective distances between all pairs of nodes in the array.

Note that the nodes in the array are equidistant from each other, and therefore $d_{PA_2} - d_{A_1A_2} = 0$. In addition, we can eliminate the modulo 2π phase ambiguity by requiring the distance between antennas in the array to be less than half the wavelength. We can therefore rearrange (2) so that known values are on the right-hand side:

$$d_{A_1PM} = d_{A_1M} - d_{PM} = \frac{\Delta\varphi\lambda}{2\pi}. \quad (3)$$

We refer to d_{A_1PM} as a *t-range* [16]. The t-range is significant because it defines the arm of a hyperbola that intersects the position of mobile node M, and whose asymptote passes through the midpoint of the line $\overline{A_1P}$, connecting the primary and assistant nodes. Figure 3 illustrates such a hyperbola with foci A_1 and P. From the figure, we see that the bearing of the asymptote is $\beta = \tan^{-1}(b/a)$, where $a = d_{A_1PM}/2$, $b = \sqrt{c^2 - a^2}$, and $c = d_{A_1P}/2$. In terms of known distances, the bearing of the asymptote is defined as

$$\beta = \tan^{-1}\left(\frac{\sqrt{\left(d_{A_1P}/2\right)^2 - \left(d_{A_1PM}/2\right)^2}}{\left(d_{A_1PM}/2\right)}\right). \quad (4)$$

In [9], we demonstrated that we can estimate β with an average accuracy of 3.2° using this technique.

4. The TripNav Waypoint Navigation System

The TripNav waypoint navigation system consists of anchor nodes as described in Section 3.2 and a mobile sensor that traverses a region in order to perform some task. In order for a mobile node to travel between waypoints, it is necessary to know the node's current position. By approximating the bearing of the mobile node from a sufficient number of landmarks, node position can be estimated using triangulation. Figure 4 illustrates a simple waypoint navigation scenario.

Figure 4: Waypoint navigation. A mobile device traverses the sensing region by navigating between position coordinates.

Determining spatial relationships for mobile sensor nodes is nontrivial due to the extreme resource limitations inherent in these types of devices. Designing appropriate localization and navigation algorithms becomes challenging in what would otherwise be a fairly straightforward process. Consequently, we make some assumptions about the system. We assume that all participating sensor nodes have wireless antennas and that we can use these to observe the phase of a transmitted sinusoidal interference signal. We also assume that the mobile platform is equipped with a digital compass or has some other capability for estimating its current orientation. Finally, we assume that a sufficient number of anchors are within range of the mobile node at all times. A minimum of two anchor bearings are required for triangulation; however, a greater number of bearings will result in a more accurate position estimate.

Figure 5 is a diagram of the control loop, which illustrates how the waypoint navigation system works. A mobile sensor node traverses the sensing region by moving from one waypoint to another. The waypoint coordinates are stored in the mote's memory. The mobile node observes the phase of interference signals transmitted sequentially by anchor nodes at known positions within the sensing region. Anchor

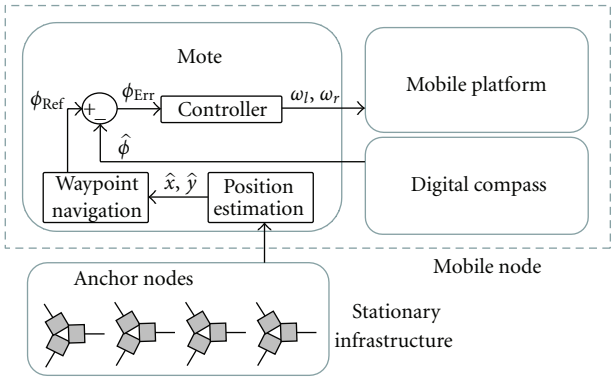


Figure 5: Control loop for waypoint navigation.

bearings are estimated, from which the position of the mobile node, $(\hat{x}, \hat{y})$, is calculated using triangulation. These coordinates are then used by the waypoint navigation logic to determine if the mobile node has reached the next waypoint. If this is the case, a new heading ϕ_{Ref} is computed and a course correction is determined based on the difference between the current estimated heading (obtained by the onboard digital compass) and the new computed heading. This heading offset, ϕ_{Err}, is input into a simple controller, which appropriately updates the angular velocities of the wheels, ω_l and ω_r, in order to keep the mobile node on the correct trajectory to intercept the waypoint. This process runs continuously until the last waypoint is reached. We describe each step of this process in detail as follows.

4.1. Mobile Platform Kinematics. We use the following equations to describe the kinematic model of our two-wheel mobile platform with differential steering:

$$\begin{aligned} \dot{x} &= \frac{r(\omega_r + \omega_l)}{2} \cos\phi, \\ \dot{y} &= \frac{r(\omega_r + \omega_l)}{2} \sin\phi, \\ \dot{\phi} &= \frac{r(\omega_r - \omega_l)}{2b}, \end{aligned} \tag{5}$$

where x and y constitute the mobile node's position, ϕ is the heading, r is the wheel radius, b is the distance between the hub center of the driving wheel and mobile platform axis of symmetry, and ω_r and ω_l are the right and left wheel angular velocities, respectively. The speed of the mobile node is the magnitude of the velocity, and in terms of wheel angular velocity is represented as $|v| = r(\omega_r + \omega_l)/2$.

4.2. Position and Heading Estimation. In order for a mobile node to travel between arbitrary waypoints, it is necessary to know its current position and heading. Having approximated the bearing of the mobile node from a sufficient number of anchors, we can estimate its position using triangulation. Triangulation is the process of determining the position of an object by using the bearings from known reference positions. When two reference points are used (Figure 6(a)), the target position will be identified as the third point in a triangle of two known angles (the bearings from each reference point) and the length of one side (the distance between reference points).

The intersection of bearings can be calculated using the following equations:

$$\begin{aligned} x &= x_2 + \cos(\alpha_2) \frac{y_2 - y_1 - \tan(\alpha_1)(x_2 - x_1)}{\cos(\alpha_2)\tan(\alpha_1) - \sin(\alpha_2)}, \\ y &= y_2 + \sin(\alpha_2) \frac{y_2 - y_1 - \tan(\alpha_1)(x_2 - x_1)}{\cos(\alpha_2)\tan(\alpha_1) - \sin(\alpha_2)}, \end{aligned} \tag{6}$$

where (x, y) are the coordinates of the intersecting bearings (i.e., the position estimate of the mobile node), (x_i, y_i) are the coordinates of the reference position (i.e., the anchor), and α_i is the bearing of the mobile node relative to the anchor.

When the position of the mobile node is directly between the two reference points (Figure 6(b)), two bearings are not sufficient to determine position because the mobile node could be located at any point on that axis. Therefore, a third bearing is required to disambiguate. However, three bearings may not intersect at the same point if any bearing is inaccurate (Figure 6(c)). Triangulation techniques are presented in [27–30], in which position estimation using more than two bearings is considered. The method we use is a least squares orthogonal error vector solution based on [10, 31], which is rapid, has low complexity, and still provides accurate position estimates from noisy bearing measurements.

Least squares triangulation using orthogonal error vectors works as follows. Figure 7 illustrates a simplified setup with a single anchor (T_i) and mobile node (M). The actual bearing from the anchor to the mobile node is denoted by β_i and the estimate by $\hat{\beta}_i$. Similarly, the vector pointing from the anchor position to the actual mobile node position is denoted by $\mathbf{v}_i$ and the vector pointing to the estimated mobile node position by $\hat{\mathbf{v}}_i$. Finally, we denote the difference between the actual and estimated bearing vectors as the orthogonal error vector $\mathbf{e}_i$ such that $\mathbf{e}_i^{\mathsf{T}} \hat{\mathbf{v}}_i = 0$.

If we let $\mathbf{a}_i = \begin{bmatrix} \sin\hat{\beta}_i \\ -\cos\hat{\beta}_i \end{bmatrix}$, then the orthogonal error vector is formally defined as

$$\mathbf{e}_i = \|\mathbf{M} - \mathbf{T}_i\| \sin\left(\hat{\beta}_i - \beta_i\right) \mathbf{a}_i, \tag{7}$$

where $\|\mathbf{M} - \mathbf{T}_i\|$ is the distance between the mobile node and anchor position vectors, $\hat{\beta}_i - \beta_i$ is the Gaussian bearing noise with zero mean and variance σ_i^2, and $\mathbf{a}_i$ is the unit vector orthogonal to $\hat{\mathbf{v}}_i$.

The position of the mobile node can be represented as $\mathbf{M} = \mathbf{T}_i + \hat{\mathbf{v}}_i + \mathbf{e}_i$. To remove $\hat{\mathbf{v}}_i$, we multiply by the transpose of $\mathbf{a}_i$, resulting in

$$\mathbf{a}_i^{\mathsf{T}} \mathbf{M} = \mathbf{a}_i^{\mathsf{T}} \mathbf{T}_i + \eta_i, \tag{8}$$

where $\eta_i = \|\mathbf{M} - \mathbf{T}_i\| \sin(\hat{\beta}_i - \beta_i)$. Considering all anchors $(i = 1, \ldots, N)$, we have a system of equations that takes the form:

$$\mathbf{A}\mathbf{M} = \mathbf{b} + \eta, \tag{9}$$

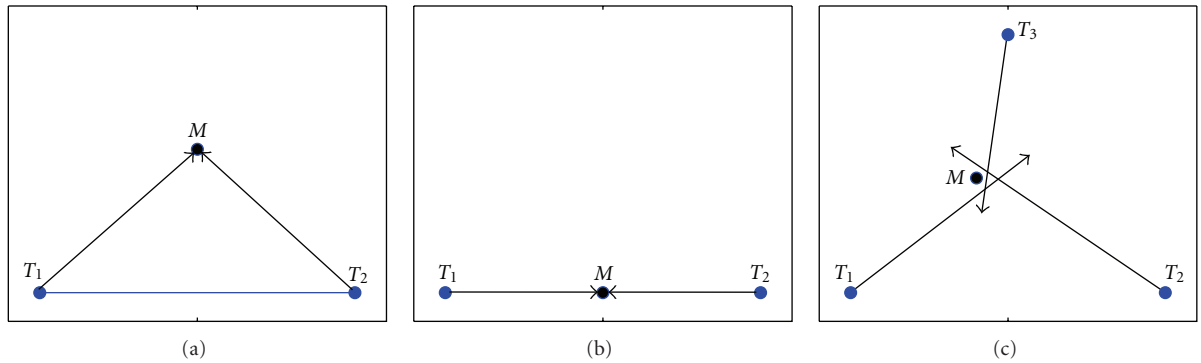


Figure 6: Triangulation. (a) As few as two bearings from known positions are required to estimate the position of a target. (b) Degenerate case where a third bearing is needed to disambiguate position. (c) Three bearings may not intersect at the same position.

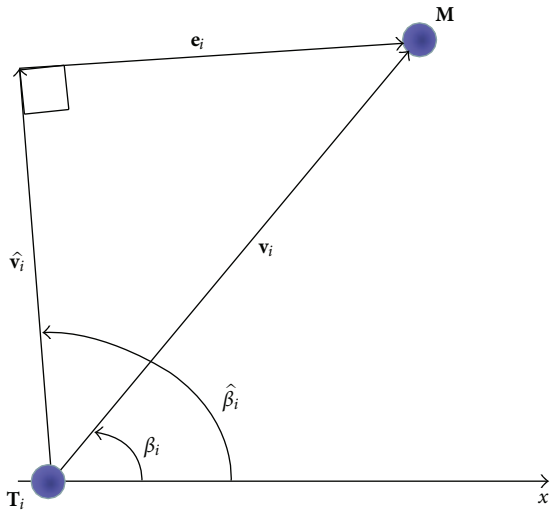


Figure 7: Least squares triangulation using orthogonal error vectors.

where $\mathbf{A} = [\mathbf{a}_1{}^\intercal, \mathbf{a}_2{}^\intercal, \ldots, \mathbf{a}_N{}^\intercal]^\intercal$ and $\mathbf{b} = [\mathbf{a}_1{}^\intercal\mathbf{T}_1, \mathbf{a}_2{}^\intercal\mathbf{T}_2, \ldots, \mathbf{a}_N{}^\intercal\mathbf{T}_N]^\intercal$. A least squares solution for estimating $\mathbf{M}$ is given by

$$\widehat{\mathbf{M}} = (\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{A}^\intercal\mathbf{b}, \tag{10}$$

where $\widehat{\mathbf{M}}$ is the position estimate returned by the triangulation using noisy bearing measurements from N anchors.

Using this method, a node can determine its position with as little as two anchors, the minimum required for triangulation. The localization algorithm outputs the estimated position of the mobile nodes: $\hat{x}$ and $\hat{y}$; however, it is unable to estimate orientation. Therefore, to obtain the heading estimate $\hat{\phi}$, we use a digital compass attached to the mobile platform. An alternative software-based method to estimate heading without a digital compass is presented in Section 5.4.

4.3. Waypoint Navigation. The mobile node needs to follow a trajectory (reference heading) that will lead it to the next waypoint. The bearing from the node's current position to the waypoint is one such trajectory. However, when the mobile node is close to the waypoint, a small localization error can contribute to large reference heading error. Instead, we define the reference heading as the bearing from the previous waypoint (or the initial position estimate of the mobile node) to the next waypoint:

$$\phi_{\text{Ref}} = \begin{cases} \tan^{-1}\left(\dfrac{w_{y_i} - \hat{y}}{w_{x_i} - \hat{x}}\right) & \text{if } i = 1 \\ \tan^{-1}\left(\dfrac{w_{y_i} - w_{y_{i-1}}}{w_{x_i} - w_{x_{i-1}}}\right) & \text{if } i > 1, \end{cases} \tag{11}$$

where w_{x_i} and w_{y_i} are the coordinates of waypoint i and $\hat{x}$ and $\hat{y}$ are the estimated position of the mobile node. Initially, ϕ_{Ref} is computed based on the position of the mobile node and the first waypoint. After the mobile node has reached the first waypoint, ϕ_{Ref} is calculated once for each waypoint i at the time waypoint $i - 1$ is reached.

Heading error is then determined by subtracting the mobile node's heading estimate, $\hat{\phi}$, from the reference heading, ϕ_{Ref}:

$$\phi_{\text{Err}} = \phi_{\text{Ref}} - \hat{\phi}. \tag{12}$$

4.4. Mobile Sensor Control. To arrive at the wheel angular velocities that will keep the mobile sensor on the reference trajectory, we use a PI controller that takes the heading error ϕ_{Err} as an input. Because the heading wraps to 0 at 2π, we shift the heading error to fall between $-\pi$ and π:

$$\phi_{\text{Err}} = \begin{cases} \phi_{\text{Err}} - 2\pi & \text{if } \phi_{\text{Err}} > \pi \\ \phi_{\text{Err}} + 2\pi & \text{if } \phi_{\text{Err}} < -\pi \\ \phi_{\text{Err}} & \text{otherwise.} \end{cases} \tag{13}$$

The controller then takes the following form:

$$\dot{\phi} = K_p\phi_{\text{Err}}(T) + K_iT_e\sum_{t=1}^{T}\phi_{\text{Err}}(t), \tag{14}$$

where K_p and K_i are constant proportional and integral gains, respectively, T is the current sample number, $\phi_{\text{Err}}(t)$ is the heading error for sample t, and T_e is the time elapsed

from the previous sample. The output of the controller, $\dot{\phi}$, is the updated angular velocity of the mobile node; however, the mobile platform is commanded by specifying an angular velocity for each wheel. Consequently, we convert $\dot{\phi}$ into individual wheel angular velocities, ω_l and ω_r, as follows:

$$\omega_l = \frac{|v| - b\dot{\phi}}{r}, \qquad \omega_r = \frac{|v| + b\dot{\phi}}{r}. \tag{15}$$

Here, r and b (defined in Section 4.1) are system parameters with known values. $|v|$ is an input parameter to this system and does not change even though the mobile platform may not actually achieve the desired value. This is because we are only interested in regulating the heading, and not the speed, of the mobile platform.

The effect of the above transformation is that both wheels will be set with an equal desired base speed. If heading error exists, the controller will minimize it by turning one wheel faster than the base speed, and the other wheel slower, which will result in the mobile node turning in the correct direction as it moves forward. This type of controller has low runtime complexity and does not require a substantial amount of memory.

5. Error Analysis

In this section, we analyze the main sources of error in TripNav. We do this by generating a simulated setup and observing how various error sources affect the results. We also analyze the system assuming that a digital compass is not available. The simulation engine models the dynamics of the mobile node and computes the ideal bearings from each anchor at each timestep. Triangulation is then performed using the computed bearings. For the error analysis, Gaussian noise is added to the heading and position estimates, as described below for each source of error. In the simulation, we position anchors at the corners of a 20×20 meter region. The mobile node follows a path that takes around a 10×10 meter square within the sensing region. The desired speed of the mobile node is fixed first at 100 mm/s and then at 400 mm/s. This setup is identical to our real-world experimental evaluation, described in detail in Section 7 and illustrated in Figure 18.

5.1. Position Estimation Error. Although position error can reach as high as several meters in the worst case, it contributes relatively little to TripNav error. This is because position estimates are only used to recognize waypoint proximity. The rest of the time, the digital compass is used to maintain the desired trajectory. To analyze the effect of localization accuracy on TripNav, we simulate the system under ideal conditions, while adding Gaussian noise to the position with zero mean and varying the standard deviation between zero and five meters. Figure 8 shows the simulated paths of the mobile node with different localization accuracies. We see from the figure that even with large position error, the mobile node will still complete the circuit; however, the path it follows can be offset significantly from the desired path. Note that there are a greater number of data points for the 100 mm/s simulation because the mobile sensor is moving slower and can therefore perform more measurements.

5.2. Digital Compass Measurement Noise. In order to compute the heading error of the mobile node, its current orientation must be known. To determine this, we use a digital compass. To understand how noisy compass sensor data affects navigation, we performed 100 simulated runs under ideal conditions, introducing a Gaussian noise to the compass heading with zero mean and a standard deviation of 0.5°, 1°, 2°, 3°, 4°, and 5°. Figure 9 shows the average associated position error for each. From the figure, we see that even a compass heading error as high as 5° does not contribute significantly to the position error.

5.3. Latency. Because the mobile node is in motion while performing localization, the accuracy of the position estimate will depend on the speed of the mobile node and the latency of the localization algorithm. Bearings from each anchor are estimated sequentially. Triangulation is then performed to determine position by finding the intersection of these bearing vectors. However, even if all other sources of error were absent from this system, these bearing vectors would still not intersect at a common point because each measurement is made from a slightly different physical location. In addition, once all measurements have been taken, the mobile node continues to change its location while phase data is being transmitted from the anchor nodes and the position estimate is computed. Therefore, the faster the TripNav control loop runs, the more accurate the position estimates will be because the mobile node will not have had a chance to move far from the position where the localization algorithm was initiated.

To analyze how this affects the accuracy of TripNav, we simulate the system under ideal conditions while varying the number of anchors. We performed 100 simulated runs and averaged the position error for each localization. Figure 10 shows the average position error we can expect due to latency when we use two, three, and four anchors. From the figure, we can see that the latency incurred by increasing the number of anchors affects TripNav position accuracy on the order of centimeters.

5.4. Waypoint Navigation without the Digital Compass. The digital compass used in our implementation (see Section 6) is the Honeywell HMR3300 [32]. The compass has a compact footprint measuring 2.54 cm by 3.81 cm, is accurate up to 1° with 0.1° resolution and 0.5° repeatability, and has tilt compensation. However, the compass is expensive relative to the mobile sensor, and is therefore not an ideal solution.

Instead, it is possible to obtain our heading without the digital compass by computing the angle with respect to an arbitrary axis between the previous and current positions, as illustrated in Figure 11(a). The accuracy of

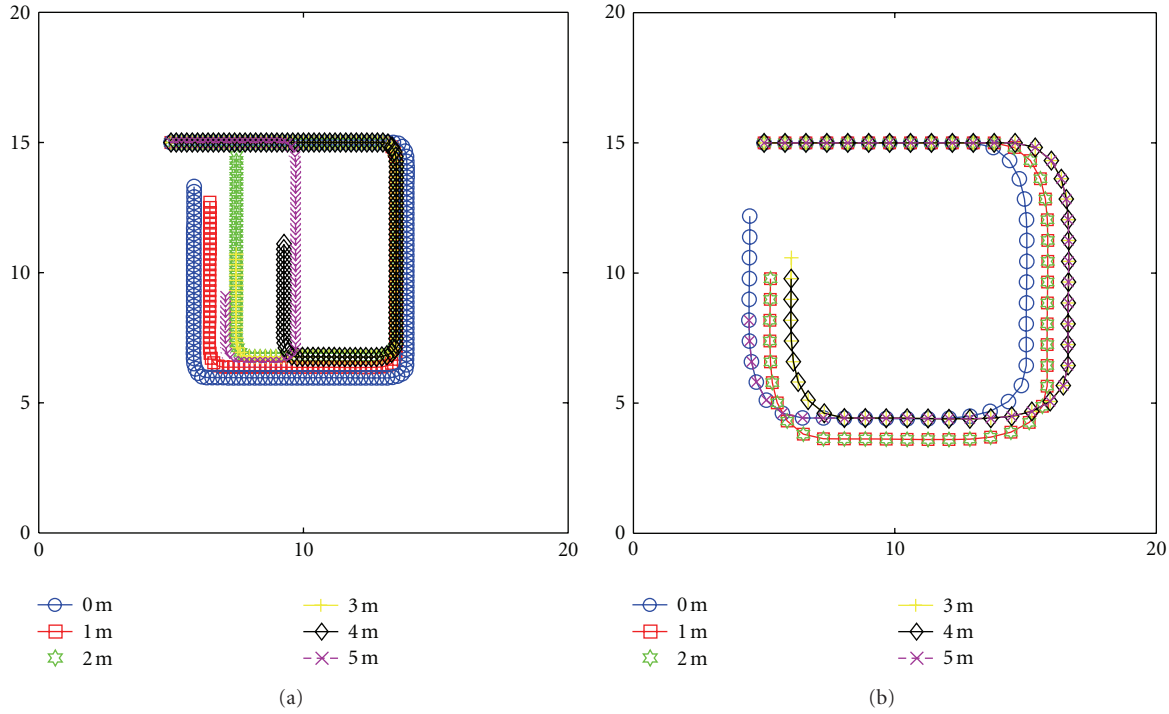


Figure 8: TripNav trajectories due to position error when the mobile node speed is (a) 100 mm/s and (b) 400 mm/s.

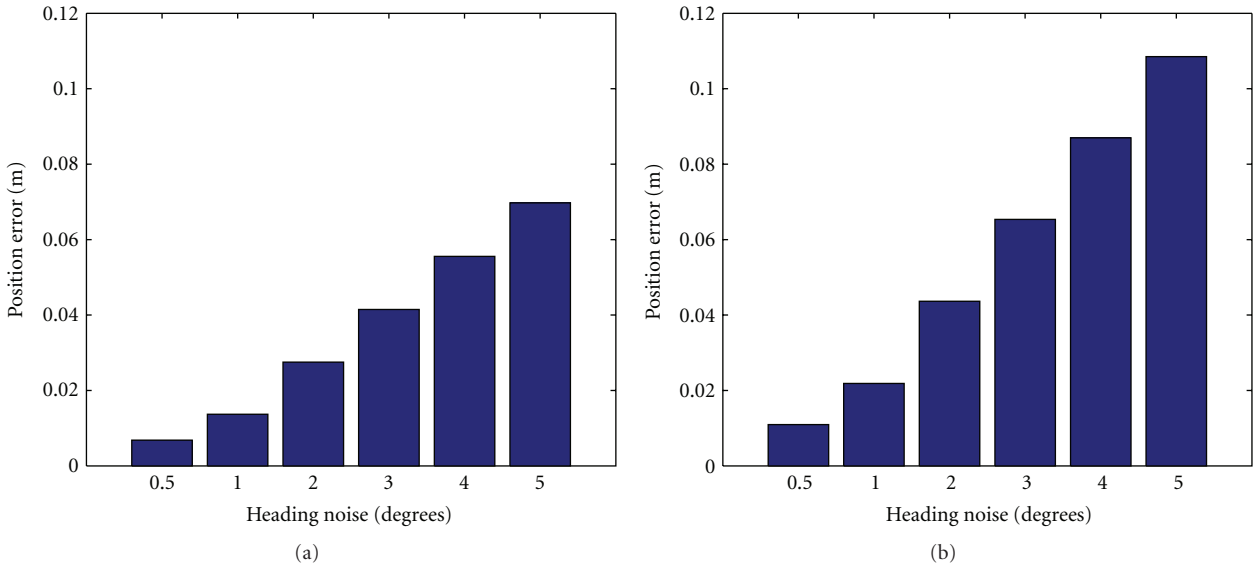


Figure 9: TripNav average position error due to digital compass sensor noise when the mobile node speed is (a) 100 mm/s and (b) 400 mm/s.

the heading estimate will depend on the accuracy of the position estimates and assume that the mobile device takes the shortest path between the two positions. If in fact the mobile device does not move between the two positions along a straight line (see Figure 11(b)), the heading estimate could be highly inaccurate. This can be caused, for example, by uneven terrain, drive motor wear and tear, and wheel slippage. However, successive position measurements taken relatively close in time will minimize the amount of heading inaccuracy that can occur. An extended Kalman filter (EKF) can further improve the heading estimate, thereby eliminating the need for the digital compass.

The EKF linearizes the estimation about the current state of the mobile node by applying the partial derivatives of the process and measurement functions, which take the form:

$$X_k = F(X_{k-1}, u_k, w_{k-1}),$$
$$z_k = h(X_k, v_k), \tag{16}$$

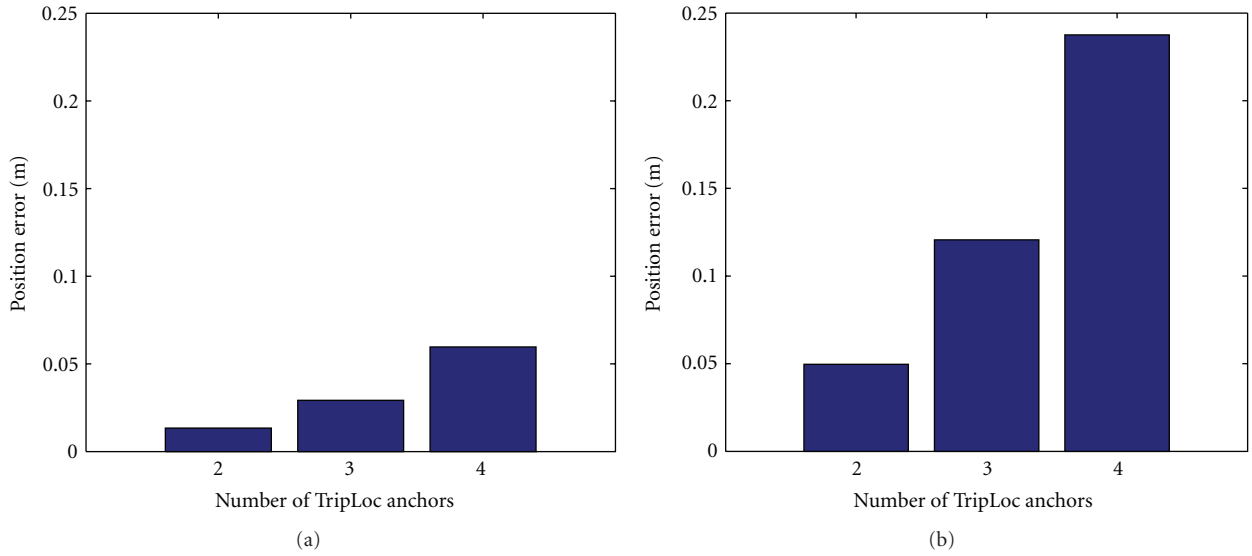


FIGURE 10: TripNav average position error due to latency using 2, 3, and 4 anchors, when the mobile sensor speed is (a) 100 mm/s and (b) 400 mm/s.

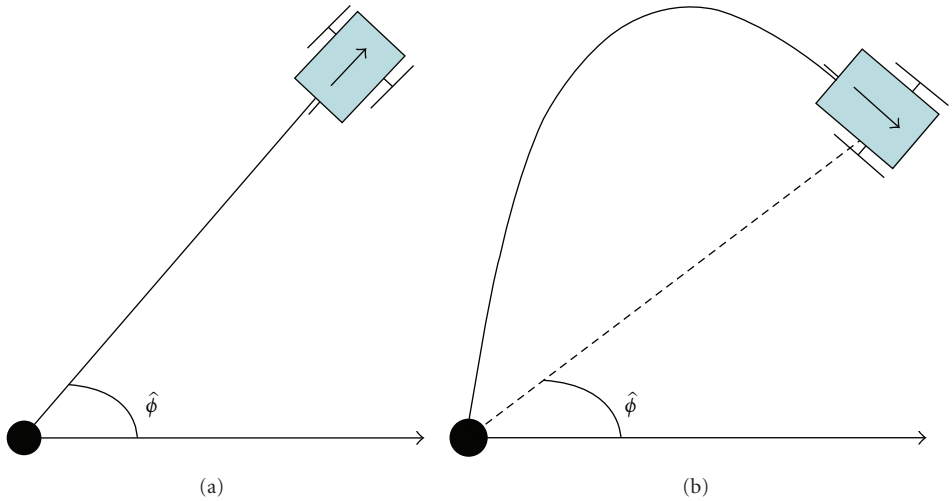


FIGURE 11: (a) Computing the heading of the mobile device. (b) Heading inaccuracy will result if the mobile device does not travel in a straight line.

where X_k is the state of the mobile node $\{x, y, \phi, \dot{x}, \dot{y}\}$, u_k is the process input $\{\omega_l, \omega_r\}$ obtained from the controller, w_k is the process noise with covariance Q, z_k is the set of position estimates obtained from the anchor nodes, v_k is the measurement noise with covariance R, and k is the timestep. The heading estimate $\hat{\phi}$ is computed as the angle of the velocity vector $\{\dot{x}, \dot{y}\}$ with respect to the x axis. Replacing the digital compass with the EKF in our waypoint navigation system results in the control loop pictured in Figure 12.

We ran simulations similar to our real-world experiments (see Section 7), both at 100 mm/s and 400 mm/s. For each set of simulations, we varied the bearing error of the localization algorithm (0, 2, 5, and 10 degrees). The wheel angular velocity error was fixed at 1 degree. The average position and

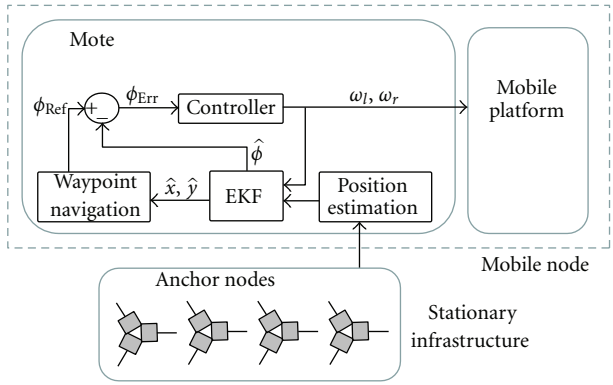


FIGURE 12: Control loop for waypoint navigation using the EKF for heading estimation.

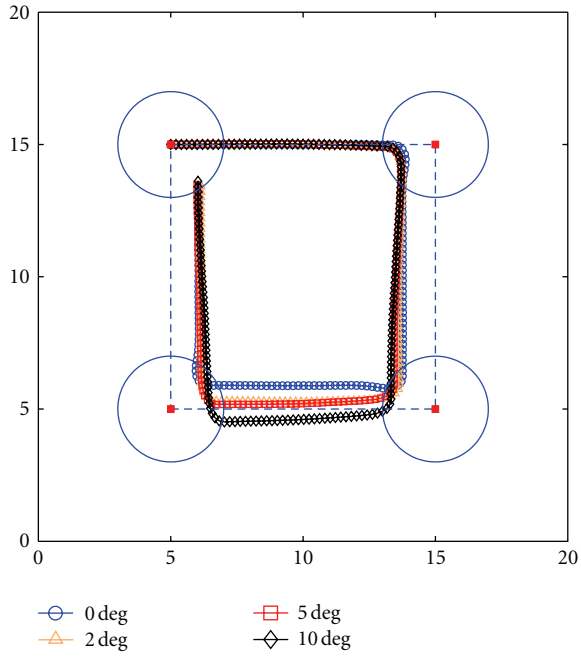


Figure 13: Simulations of mobile sensor waypoint navigation over a simple track at 100 mm/s using an EKF for heading estimation. Each track was run with different bearing errors of the localization algorithm.

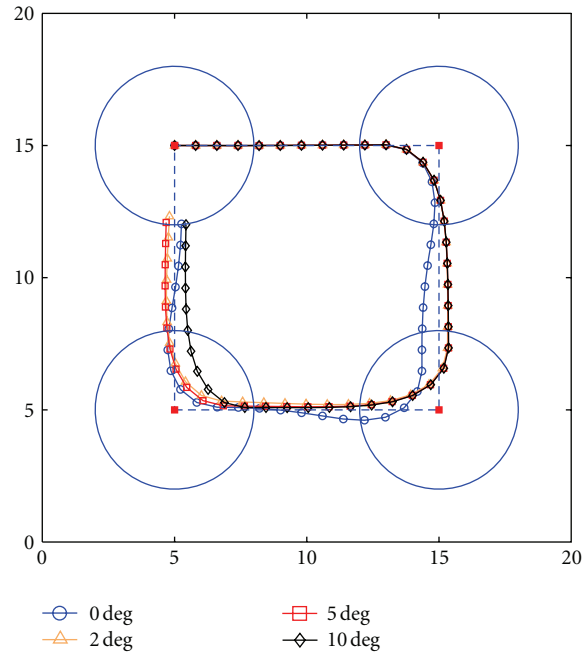


Figure 14: Simulations of mobile sensor waypoint navigation over a simple track at 400 mm/s using an EKF for heading estimation. Each track was run with different bearing errors of the localization algorithm.

heading errors for each simulation are listed in Table 1, and the tracks are pictured in Figures 13 and 14.

In addition, we performed simulations over a more complex track similar to that used in [33]. The average position and heading errors for each simulation are listed in Table 2, and the tracks are pictured in Figures 15 and 16.

It is clear from these simulation results that mobile sensor waypoint navigation can be achieved using the TripNav system without the need for a digital compass. This is beneficial to the overall system because we can still accurately maintain our desired trajectory without adding the weight, cost, and energy demand of additional hardware to the mobile platform.

6. Implementation

Our mobile sensor is comprised of an XSM mote [12] attached to an iRobot Create mobile platform [13], as pictured in Figure 17. All localization and control operations are performed on the mote, which communicates with the Create microcontroller over a serial interface. Mobile sensor heading is determined using a Honeywell HMR3300 digital compass [32]. The Create acts solely as a mobile platform and does not perform any computation or control independently of the mote. The anchor node implementation is described in [9].

The Create is a small-profile mobile platform, only 7.65 cm tall. Fixing the XSM mote to the Create body becomes problematic because the localization transmission signal is affected by ground-based reflections. We built a mount out of lightweight PVC pipe that places the mote 85 cm off the ground. We determined experimentally that 85 cm was sufficient to minimize the effect of ground-based reflections. The mount is fixed to the Create body, and houses the XSM mote, the digital compass, the connecting cable assembly for communicating with the Create and a battery pack.

One of the main implementation challenges for TripNav is designing an accurate rapid localization system as well as waypoint navigation and mobile control logic that is small enough to fit in the memory of a single mote. Our TripNav implementation consumes approximately 3.1 kB RAM and 60 kB of programming memory.

7. Experimental Evaluation

We place four anchors at the corners of a 20×20 meter region in a nonmultipath outdoor environment. The mobile node is given as a series of four waypoint coordinates within the region and instructed to drive along the square route that connects the waypoints. Once the mobile node reaches the last waypoint (i.e., completes the circuit), it is instructed to come to a stop. Figure 18 illustrates this setup.

7.1. Performance Analysis. There are several tunable parameters for waypoint navigation using TripNav. Since TripNav only controls the heading of the mobile node and not its speed, an important system parameter is the *target drive speed* (the translational speed of the mobile node). The maximum speed of the Create is 500 mm/s. However,

Table 1: Average position and heading errors for simple simulation tracks with varying localization bearing error.

Speed	0° localization bearing error	2° localization bearing error	5° localization bearing error	10° localization bearing error
100 mm/s	0 m 1.07°	0.08 m 1.30°	0.16 m 1.75°	0.24 m 2.42°
400 mm/s	0 m 0.63°	0.18 m 1.31°	0.32 m 1.65°	0.40 m 1.67°

Table 2: Average position and heading errors for complex simulation tracks with varying localization bearing error.

Speed	0° localization bearing error	2° localization bearing error	5° localization bearing error	10° localization bearing error
100 mm/s	0 m 0.44°	0.23 m 2.37°	0.47 m 3.01°	0.86 m 3.50°
400 mm/s	0 m 1.33°	0.26 m 1.36°	0.52 m 1.66°	0.87 m 1.71°

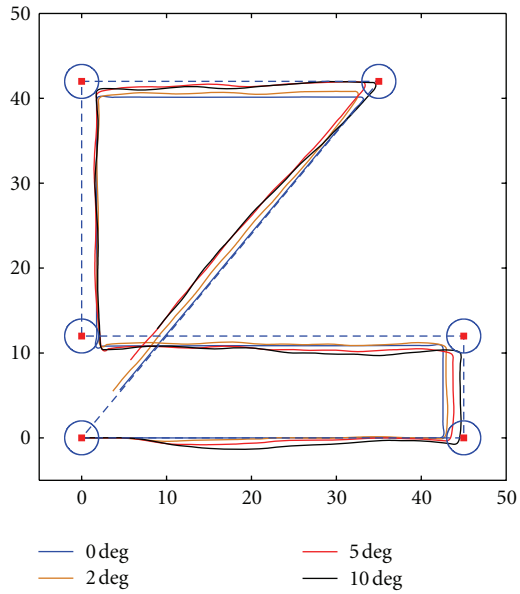


Figure 15: Simulations of mobile sensor waypoint navigation over a complex track at 100 mm/s using an EKF for heading estimation. Each track was run with different bearing errors of the localization algorithm.

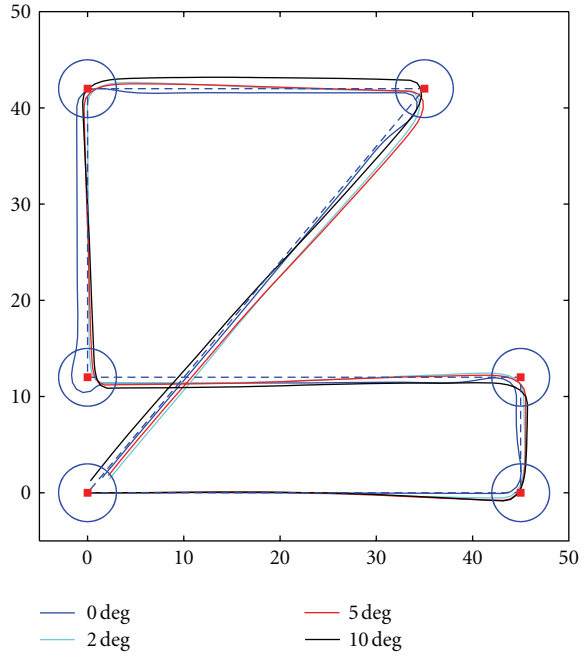


Figure 16: Simulations of mobile sensor waypoint navigation over a complex track at 400 mm/s using an EKF for heading estimation. Each track was run with different bearing errors of the localization algorithm.

because we attached a sensor mount to the body of the Create, the increased weight (as well as uneven terrain) limits the speed to about 450 mm/s. Because the controller specifies wheel speeds such that one wheel may rotate faster than the target speed and the other slower, we set our maximum target drive speed to be 400 mm/s. For our experiments, we performed waypoint navigation with target drive speeds of 100 mm/s and 400 mm/s.

Because of localization error and continuous movement, the mobile sensor will not always be able to land exactly on the waypoint. We, therefore, select a *waypoint range* that specifies how close the mobile sensor must be to a waypoint before being allowed to proceed to the next. The size of the waypoint range is adjusted based on the speed of the mobile node and the latency of the localization. If the mobile

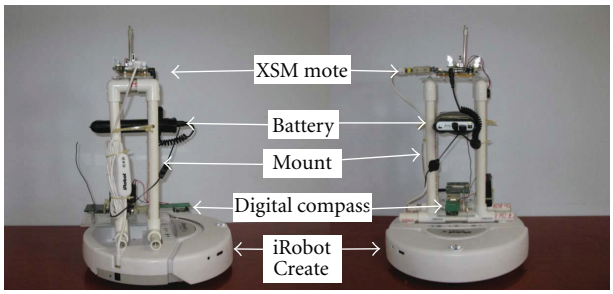


Figure 17: The TripNav mobile platform.

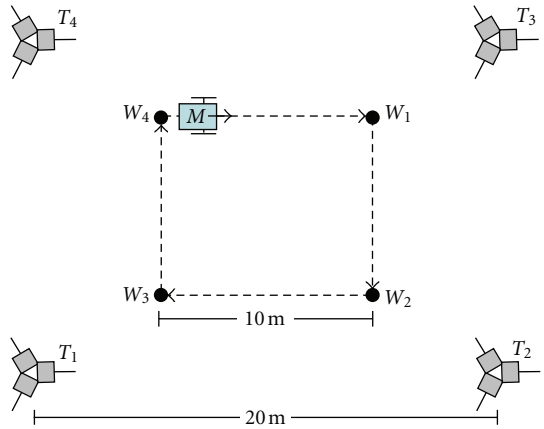


Figure 18: Waypoint navigation experimental setup. Anchors ($T_1 \cdots T_4$) surround the sensing region. The mobile node (M) is instructed to drive in a square, passing through each waypoint ($W_1 \cdots W_4$) before proceeding to the next.

node's speed is slower, we can reduce the size of the waypoint range. If the mobile node's speed is faster, we must increase the size of the waypoint range, otherwise the mobile node will not realize that it reached the waypoint. Because we make the design decision to not slow down as the mobile node nears the waypoint, or stop at the waypoint, turn, then start forward motion again, we must resort to using the waypoint range. For our experiments, we ultimately chose waypoint ranges of two meters when moving at 100 mm/s and three meters when moving at 400 mm/s. We found that if we increased the waypoint range beyond these values, the mobile node still completed its circuit; however, the path it followed had a high average position error.

Finally, to filter out inaccurate position estimates, we use a simple validation gate that approximates the distance traveled since the last position estimate by multiplying the elapsed time by the average wheel speed. If the distance difference between the current and previous position estimates is greater than the estimated travel distance plus a *position error constant* (to account for positioning and drive error), then the current position estimate is discarded. We chose a value of 2.5 meters for the position error constant.

We performed five waypoint navigation runs for both target drive speeds using TripNav. Figure 19 shows the average path of the mobile node over all runs. Note that the mobile node's path does not intersect with the waypoints and seems to stop short of the final waypoint. This is due to the waypoint range setting, where the mobile node considers the waypoint reached if it comes within the specified range. On average, position and heading accuracy with respect to the desired trajectory was 0.95 m and 4.75° when traveling at 100 mm/s and 1.08 m and 5.05° when traveling at 400 mm/s.

Figure 20 displays the outermost and innermost positions along the circuit of the mobile node over all runs. These are not individual paths, but bounds on the mobile sensor's movement over all five runs. This shows that one TripNav run does not significantly vary from another.

Table 3: Latency of TripNav components.

Component	Average latency (ms)	Maximum latency (ms)
Digital compass	49.61	89.48
Waypoint navigation	0.45	0.52
Controller	0.64	0.68
Localization (2 anchors)	888.72	956.22
Localization (3 anchors)	1283.81	1334.99
Localization (4 anchors)	1667.76	1734.48

7.2. Latency Analysis. Since the mobile sensor is moving while estimating its position, localization must be performed rapidly, otherwise the mobile node will be in a significantly different location by the time a result is returned. The speed of the entire localization process depends on the latency of each component within the TripNav system, and so we provide a timing analysis of those components here. A latency analysis of the individual components involved in bearing estimation is presented in [9].

Figure 21 shows a sequence diagram for each step in the TripNav control loop, in which two anchors (dotted boxes) and a single mobile node are used. Because phase difference is used to determine bearing, each node must measure the signal phase at the same time instant. This requires synchronization with an accuracy on the order of microseconds or better. A SyncEvent message [26] is broadcast by the primary transmitter and contains a time in the future for all participating nodes to start the first RIM. Each array then performs two RIMs, one for each primary-assistant pair. Signal transmission involves acquiring and calibrating the radio, transmitting the signal, then restoring the radio to enable data communication. The assistant nodes in the array store their phase measurements until both primary-assistant pairs have finished their RIMs, at which point they broadcast their phase measurements to the mobile node. The mobile node calculates its bearing from each array, determines its position using triangulation, obtains its heading from the digital compass, and then uses this information to move in the appropriate direction.

Table 3 lists the average and maximum execution times over 100 iterations for the components pictured in Figure 5. Note that TripNav execution time depends on the number of participating anchors because bearing from each anchor is estimated sequentially. A minimum of two anchors is required for triangulation; however, the accuracy of the localization will improve with the addition of more participating anchors. We, therefore, provide execution times for three scenarios, in which we vary the number of participating anchors between two (the minimum required) and four (the number we use in our real-world evaluation).

On average, the digital compass takes approximately 50 ms to estimate heading. This is in fact a limitation of the compass hardware, which provides heading estimates at a rate of approximately 8 Hz, or 125 ms. The 50 ms latency reflects the average time we must wait for the next heading estimate to be returned.

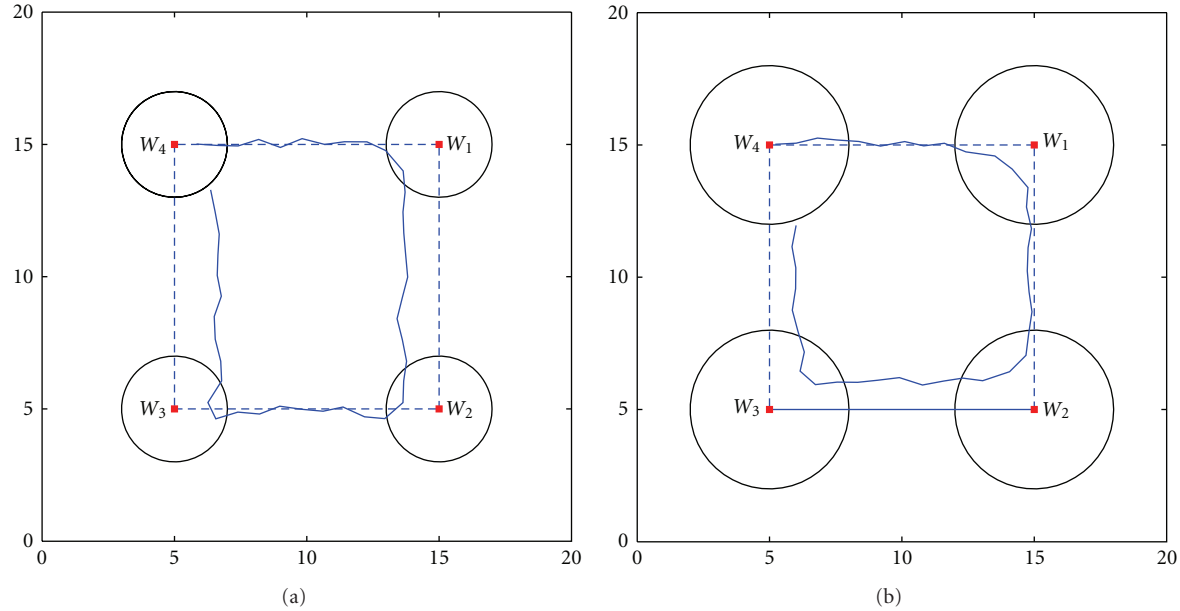


FIGURE 19: Waypoint navigation average position results when mobile sensor speed is (a) 100 mm/s, and (b) 400 mm/s. The dotted line represents the desired path. Waypoints are marked $W_1 \cdots W_4$, and the the surrounding circles represent the waypoint range of (a) 2 m and (b) 3 m.

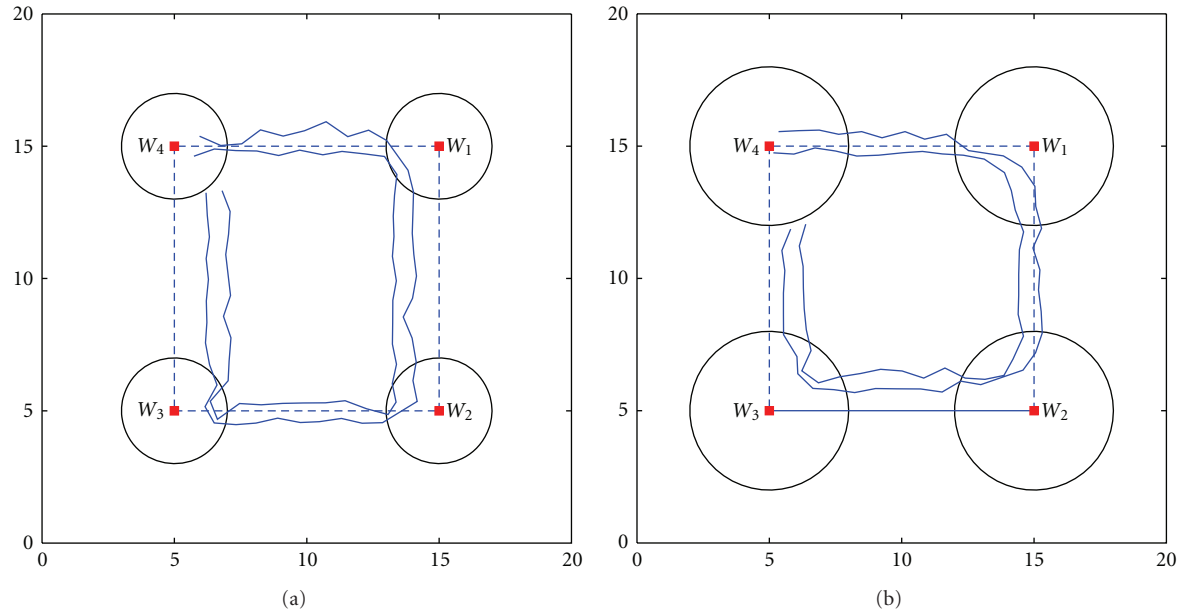


FIGURE 20: Waypoint navigation outermost and innermost path when mobile sensor speed is (a) 100 mm/s, and (b) 400 mm/s. The dotted line represents the desired path. Waypoints are marked $W_1 \cdots W_4$, and the the surrounding circles represent the waypoint range of (a) 2 m and (b) 3 m.

It is worth noting that because the mobile node acts solely as a receiver in this process, system latency is not affected by introducing more mobile nodes to the sensing region. TripNav is fully scalable in this respect; however, latency will increase as more anchors are employed, which will ultimately limit the size of the sensing region.

7.3. The Effect of TripNav Mobility on Position Accuracy. We performed our localization technique on a stationary sensor network deployment. Similar to the TripNav mobility experiments, four anchors were placed at the corners of a 20×20 meter region in an outdoor environment. Twelve stationary target nodes, placed at least 2.5 m inside the sensing region, performed 50 position estimates each. The average localization error was 0.62 m.

The experiment demonstrates the effect of TripNav mobility on the accuracy of our localization technique. When the mobile node is moving at a speed of 100 mm/s, the average position error due to mobility is 0.33 m. At a speed of 400 mm/s, the average position error is 0.46 m.

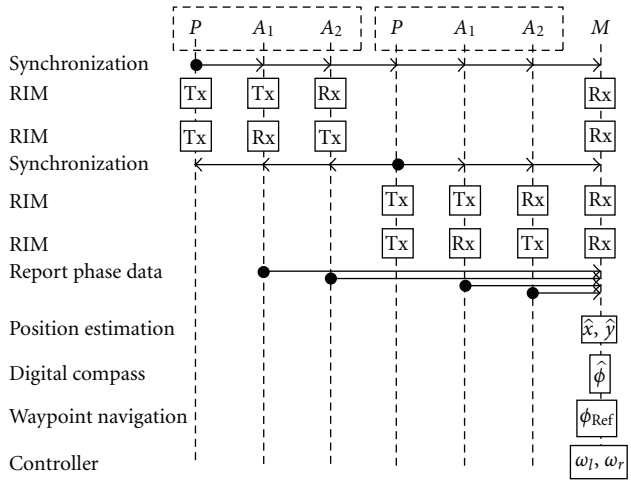


Figure 21: Sequence diagram of the TripNav control loop in which two anchors are used.

8. Conclusion

Spatiotemporal awareness in mobile wireless sensor networks entails new challenges that result from integrating resource-constrained wireless sensors onto mobile platforms. The localization methods and algorithms that provide greater accuracy on larger-footprint mobile entities with fewer resource limitations are no longer applicable. Similarly, centralized and high-latency localization techniques for static WSNs are undesirable for the majority of MWSN applications. In this paper, we presented a waypoint navigation method for resource-constrained mobile wireless sensor nodes. The method is rapid, distributed, and has submeter accuracy.

One of the biggest challenges we face with RF propagation is multipath fading. Currently, TripNav will not work acceptably in multipath environments. Outdoor urban areas and building interiors are both major sources of multipath, and yet these are places where MWSNs have the greatest utility. An RF-based localization system that provides accurate results in these environments would be a major step forward. This is a future direction for our MWSN localization and navigation research, and we have already obtained encouraging preliminary results. In [34], we were able to demonstrate that precise RF indoor 1-dimensional tracking is indeed possible, and we are currently investigating how we can extend this technique to two and three dimensions. Such fine-grained RF-based localization would enable mobile sensors to navigate through hallways of burning buildings, help to evacuate shopping malls in the event of an emergency, and monitor the health of patients in every room of their house.

Acknowledgments

This work was supported by ARO MURI Grant W911NF-06-1-0076, NSF Grant CNS-0721604, and NSF CAREER award CNS-0347440. The authors would also like to thank Peter Volgyesi for his assistance, and Metropolitan Nashville Parks and Recreation, and Edwin Warner Park for use of their space.

References

[1] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, "Robomote: enabling mobility in sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, p. 55, Los Angeles, Calif, USA, April 2005.

[2] J. Friedman, D. Lee, I. Tsigkogiannis et al., "Ragobot: a new platform for wireless mobile sensor networks," in *Proceedings of the 1st IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '05)*, vol. 3560 of *Lecture Notes in Computer Science*, p. 412, Marina del Rey, Calif, USA, July 2005.

[3] S. Bergbreiter and K. S. J. Pister, "CotsBots: an off-the-shelf platform for distributed robotics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)*, pp. 27–31, Las Vegas, Nev, USA, October 2003.

[4] M. B. McMickell, B. Goodwine, and L. A. Montestruque, "MICAbot: a robotic platform for large-scale distributed robotics," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '03)*, pp. 1600–1605, September 2003.

[5] H. B. Brown, J. M. Vande Weghe, C. A. Bererton, and P. K. Khosla, "Millibot trains for enhanced mobility," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 452–461, 2002.

[6] I. Getting, "The global positioning system," *IEEE Spectrum*, vol. 30, no. 12, pp. 36–47, 1993.

[7] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "Cricket location-support system," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 32–43, August 2000.

[8] M. Maroti, B. Kusy, G. Balogh et al., "Radio interferometric geolocation," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 1–12, November 2005.

[9] I. Amundson, J. Sallai, X. Koutsoukos, and A. Ledeczi, "Radio interferometric angle of arrival estimation," in *Proceedings of the 7th European Conference on Wireless Sensor Networks (EWSN '10)*, vol. 5970 of *Lecture Notes in Computer Science*, pp. 1–16, Springer, Coimbra, Portugal, February 2010.

[10] J. N. Ash and L. C. Potter, "Robust system multiangulation using subspace methods," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 61–68, April 2007.

[11] M. O. Franz and H. A. Mallot, "Biomimetic robot navigation," *Robotics and Autonomous Systems*, vol. 30, no. 1, pp. 133–153, 2000.

[12] P. Dutta, M. Grimmer, A. Arora, S. Bibykt, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 497–502, April 2005.

[13] iRobot, "Create programmable robot," http://www.irobot.com/.

[14] I. Amundson and X. D. Koutsoukos, "A survey on localization for mobile wireless sensor networks," in *Proceedings of the 2nd International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT '09)*, R. Fuller and

X. Koutsoukos, Eds., vol. 5801 of *Lecture Notes in Computer Science*, pp. 235–254, Springer, 2009.

[15] B. Kusy, A. Ledeczi, and X. Koutsoukos, "Tracking mobile nodes using RF doppler shifts," in *Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys '07)*, pp. 29–42, November 2007.

[16] B. Kusy, J. Sallai, G. Balogh et al., "Radio interferometric tracking of mobile wireless nodes," in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys '07)*, pp. 139–151, June 2007.

[17] L. Fang, P. J. Antsaklis, L. A. Montestruque et al., "Design of a wireless assisted pedestrian dead reckoning system—the NavMote experience," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 6, pp. 2342–2358, 2005.

[18] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: sensors and techniques," *Journal of Robotic Systems*, vol. 14, no. 4, pp. 231–249, 1997.

[19] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, pp. 775–784, March 2000.

[20] K. Lorincz and M. Welsh, "MoteTrack: a robust, decentralized approach to RF-based location tracking," *Personal and Ubiquitous Computing*, vol. 11, no. 6, pp. 489–503, 2007.

[21] M. W. M. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[22] C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue, "Simultaneous localization, calibration, and tracking in an ad hoc sensor network," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 27–33, April 2006.

[23] A. Verma, H. Sawant, and J. Tan, "Selection and navigation of mobile sensor nodes using a sensor network," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pp. 41–50, March 2005.

[24] Moog Crossbow, "Mica2," http://www.xbow.com.

[25] Texas Instruments, "CC1000: single chip very low power RF transceiver," http://focus.ti.com/docs/prod/folders/print/cc1000.html.

[26] B. Kusy, P. Dutta, P. Levis, M. Maroti, A. Ledeczi, and D. Culler, "Elapsed time on arrival: a simple and versatile primitive for canonical time synchronization services," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 1, pp. 239–251, 2006.

[27] C. D. McGillem and T. S. Rappaport, "Beacon navigation method for autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 38, no. 3, pp. 132–139, 1989.

[28] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1734–1743, April 2003.

[29] A. Nasipuri and R. El Najjar, "Experimental evaluation of an angle based indoor localization system," in *Proceedings of the 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '06)*, Boston, Mass, USA, April 2006.

[30] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 251–263, 1997.

[31] K. Dogancay, "Bearings-only target localization using total least squares," *Signal Processing*, vol. 85, no. 9, pp. 1695–1710, 2005.

[32] Honeywell, "HMR3300 digital compass," http://www.magneticsensors.com/.

[33] B. Kusy, I. Amundson, J. Sallai, P. Volgyesi, A. Ledeczi, and X. Koutsoukos, "RF doppler shift-based mobile sensor tracking and navigation," *ACM Transactions on Sensor Networks*, vol. 7, no. 10, 2010.

[34] J. Sallai, I. Amundson, A. Ledeczi, X. Koutsoukos, and M. Maroti, "Using RF received phase for indoor tracking," in *Proceedings of the Workshop on Hot Topics in Embedded Networked Sensors (HotEmNets '10)*, June 2010.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 741821, 7 pages
doi:10.1155/2012/741821

Research Article

Distributed Control of Mobile Sensor Networks under RF Connectivity Constraints

Yiannis Stergiopoulos, Yiannis Kantaros, and Anthony Tzes

Department of Electrical and Computer Engineering, University of Patras, Rio, 26500 Achaia, Greece

Correspondence should be addressed to Yiannis Stergiopoulos, stergiopoulos@ece.upatras.gr

Received 27 November 2011; Accepted 10 April 2012

Academic Editor: Hannes Frey

This paper addresses the problem of coordinating the motion of the nodes in a mobile sensor network for area coverage applications under RF communication limitations. During network evolution, the area sensed by the network increases until it reaches optimum configuration, while information for decision making is acquired distributively among the nodes via a prespecified number of hops. Unlike previous works, radio range is not demanded to be at least twice the sensing range, imposing an extra constraint in the overall problem setup. The proposed control scheme guarantees end-to-end RF connectivity of the network, while attaining optimum area coverage. Results are further verified via simulation studies.

1. Introduction

Distributed coordination of robotic swarms has been studied widely in the last years due to its direct application in missions where human interference may be risky or even prohibited. Mobile platforms with sensing, computational, and communication capabilities are in most cases spread in areas of interest in order to investigate various physical quantities and/or even take responsibility of surveying the area assuming intruder detection, patrolling, or exploration scenarios.

Although the agents in the group share a common objective, usually optimizing an *a priori* aggregate objective function [1–3], the way they collaborate is performed in a spatially distributed manner, rather than a global-coordinating one, due to the physical restriction in the acquired information via the antennas' range. Hence, connectivity preservation during the deployment stage is an issue of major importance, since this ensures information flow among the mobile nodes in order to cooperate for achieving their common goal.

Assuming both limited-range sensing and communication abilities of the platforms, it is evident that demand for connectivity preservation and area coverage optimality cannot be achieved simultaneously, and thus there is trade-off to be balanced [4, 5]. Distributed coordination of mobile networks via intuitive nearest neighbour rules has been proposed by the authors in [6]. Connectivity control of networks has been examined in [7, 8], while more generalized coordinate-free theoretical approaches have been developed in [9, 10].

Coverage of a region by a set of nodes has been examined in previous works neglecting sensing range and thus is approached via a more networked pointed of view. Recent works have utilized topology control techniques in order to reduce the number of redundant links in congested wireless sensor networks [11–14], while antagonistic approaches have also been proposed [15, 16]. Distributed connectivity preservation during the deployment stage has been examined in [17] via estimation of the eigenvalues of the network's Laplacian [18].

In most of the works in the existing literature on the field or distributed sensors deployment, the communication range of the nodes' antennas is assumed either variable but unbounded (in terms of no upper limit) [19, 20] or bounded but greater than twice the sensing range [1, 21, 22]. This dependence of the radio range on the sensing one, although not met in practice, remarkably lets us surpass any network

connectivity issues and concentrate on optimization of the covered area.

In this paper, though, the nodes' radio range is assumed to be fixed and can be less than the aforementioned bound (i.e., twice the sensing range). This restriction imposition, although met most often in practical scenarios (where the sensors' and antennas' ranges are uncorrelated), leads in inability to apply already presented area coverage-oriented coordination schemes [2, 6, 22, 23]. Despite the aforementioned radio range constraint, the algorithm proposed in this paper guarantees network connectivity in a finite predefined number of hops, while leading the nodes in an optimal state, considering coverage terms.

The rest of the paper is organized as follows. In Section 2 the problem of surveillance of a region by a mobile sensor network is introduced, along with the main preliminaries on Voronoi partitioning. In Section 3 the background concerning network connectivity is presented, while connectivity in a finite number of hops among two nodes is analysed from a graph perspective. The proposed coordination scheme that takes into account the network's coverage performance along with communication constraints imposed due to radio restrictions is presented in Section 4. Simulation results in Section 5 further confirm efficiency of the proposed scheme, while concluding remarks are provided in the last section.

2. Coverage Problem Formulation

Consider n in number mobile robotic agents responsible for the sensing coverage of an area of interest D, defined as a convex compact set in $\mathbb{R}^2$. Let $I_n = \{1, 2, \ldots, n\}$ be the set of unique identifiers of the nodes, while their positions on the Euclidean plane are denoted by x_i, $i \in I_n$.

The robots are considered to evolve in the interior of D in discrete time via the control inputs u_i, as

$$x_i^{k+1} = x_i^k + u_i^k, \quad u_i \in \mathbb{R}^2,\ x_i \in D,\ i \in I_n, \tag{1}$$

where the superscript k denotes the current time step, $k = 0, 1, \ldots$. In this paper, at each time step it is assumed that only one node can move; thus, at the first step, $k = 1$, node $i = 1$ will move, while afterwards the node to move is determined in a random manner.

Assuming surveillance purposes, sensors are embedded on the robotic platforms that sense the area in range of r around the nodes, denoted by B_i, that is,

$$B_i = \{x \in \mathbb{R}^2 : \|x - x_i\| \le r\}, \quad i \in I_n. \tag{2}$$

In an area coverage application, the aggregate objective function under optimization can be expressed as the area of the union of the nodes' sensing regions over the D domain, that is,

$$\mathcal{H} = \int_{D \cap \bigcup_{i \in I_n} B_i} \mathrm{dS}, \tag{3}$$

where dS is the elementary surface for integration purposes.

A quite common method to deal with such kind of problems in swarm robotics is to tessellate the space into subsets via a distance-based metric and assign them among the nodes. Voronoi diagram [24], $\mathcal{V} = \{V_i,\ i \in I_n\}$, is the most common partitioning among n distinct points $x_i,\ i \in I_n$, defined as

$$V_i = \left\{x \in D : \|x - x_i\| \le \left\|x - x_j\right\|,\ j \in I_n\right\}, \quad i \in I_n. \tag{4}$$

In other words, V_i is the set of the points of D that are closer to x_i than any other points in $\{x_j,\ j \in I_n\}$. We refer to V_i as the Voronoi cell of node i.

Let $\mathcal{G}_D$ be the Delaunay graph associated with the corresponding Voronoi partitioning. We assume that the reader is familiar with the main preliminaries on graph theory [18]. Two nodes that share an edge of their Voronoi cells are considered as neighbours in $\mathcal{G}_D$. The Delaunay neighbours $\mathcal{N}_i$ of an arbitrary node i are then defined as

$$\mathcal{N}_i = \left\{j \in I_n : V_i \cap V_j \neq \emptyset,\ j \neq i\right\}, \quad i \in I_n. \tag{5}$$

Apparently, if $j \in \mathcal{N}_i$, then $i \in \mathcal{N}_j$.

Utilizing the sets V_i and considering $\mathcal{H}$, one can define in an equivalent manner the r-limited Voronoi cell of an arbitrary node i, V_i^r, as the parts of the corresponding Voronoi cell that are simultaneously sensed by that node, that is,

$$V_i^r = V_i \cap B_i, \quad i \in I_n. \tag{6}$$

It is easily proven that via this definition, the total area sensed by the network, $\mathcal{H}$, can be expressed as the summation of the r-limited Voronoi cells of the nodes, that is,

$$\mathcal{H} = \sum_{i \in I_n} \int_{V_i^r} \mathrm{dS}. \tag{7}$$

Equivalently to the Delaunay graph $\mathcal{G}_D$, one can define the $2r$-limited Delaunay one, denoted by $\mathcal{G}_D^{2r}$, where the neighbours of a node i in this graph are the nodes whose r-limited Voronoi cells share an edge with V_i^r, that is,

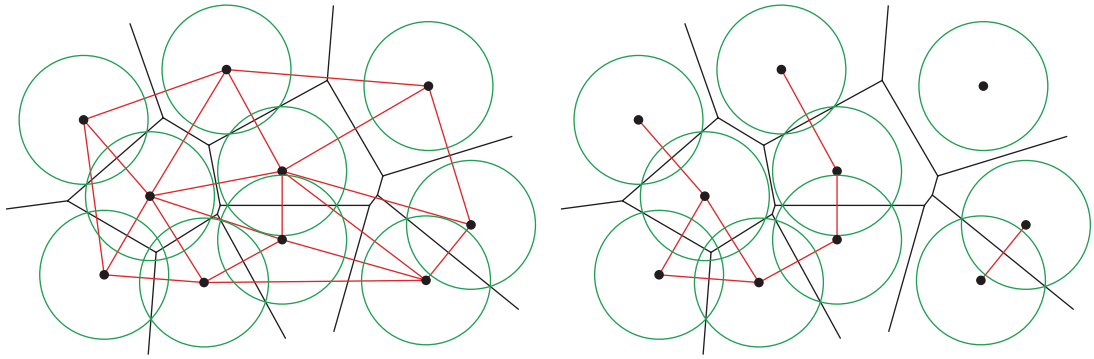
$$\mathcal{N}_i^{2r} = \left\{j \in I_n : V_i^r \cap V_j^r \neq \emptyset,\ j \neq i\right\}, \quad i \in I_n. \tag{8}$$

Figure 1 shows graphically the neighbouring relationships among the nodes in the Delaunay, $\mathcal{G}_D$, and $2r$-limited Delaunay graphs, $\mathcal{G}_D^{2r}$.

3. Radio Connectivity Issues

An issue of major importance in coordination of mobile sensor networks is the distributed nature of the designed control schemes. In other words, the nodes should organize their action without global knowledge of the network's state, but via local information from neighbouring nodes, instead.

Each node is assumed to be equipped with radio transceivers in order to be able to exchange spatial information with other neighbouring nodes in range. The antennas are assumed to transmit omnidirectionally around x_i up to a radius R. Unlike the majority of previous works



(a) (b)

Figure 1: Graphical representation of the Delaunay (a) and $2r$-limited Delaunay (b) neighbours in a sensor network.

in the existing literature, where the antennas' radii are considered variable, in this paper the latter is assumed fixed, same for all nodes, and not demanded to be at least twice the sensing range. In fact, the case $R \geq 2r$ is examined in detail in the literature, since network connectivity is trivially guaranteed that way, in the network's area-optimal state.

Apparently, a bidirectional communication link exists among any two nodes i and j if and only if $\|x_i - x_j\| \leq R$, for spatial information exchange purposes. Graphically, the neighbouring relationships among the nodes from a communication aspect (i.e., ignoring the nodes' sensing abilities) can be represented in the communication graph of the network, denoted by $\mathcal{G}_c$, where an edge exists among two nodes if and only if one is in radio-range of the other, and vice versa. It should be noted that although the Delaunay and $2r$-limited Delaunay graphs are somehow correlated, the communication graph is totally independent, since it does not rely on the nodes' sensory domains, B_i, but is determined only by the nodes' positions and the common communication radius R. As far as the neighbouring relationships in the graph $\mathcal{G}_c$ are considered, let us state the following definitions.

Definition 1. Two nodes i and j in the communication graph $\mathcal{G}_c$ associated with a wireless sensor network are called (directly) connected, while denoted by $i \sim j$, if and only if $\|x_i - x_j\| \leq R$, where R is the common radio range.

Definition 2. Given the communication graph $\mathcal{G}_c$ of a sensor network, a routing path of length ℓ among two nodes i and j is a sequence of $\ell + 1$ nodes $i, k_1, k_2, \ldots, k_{\ell-1}, j$ such that $i \sim k_1, k_1 \sim k_2, \ldots, k_{\ell-1} \sim j$.

Definition 3. Two nodes i and j in the communication graph $\mathcal{G}_c$ are called N-hops connected, while denoted by $i \sim_N j$, if and only if the minimum-length path among them (if there exists a path) has length N. When two nodes are 1-hop connected, we will simply refer to them as connected, that is, $i \sim j$.

Given a sensor network and the communication graph $\mathcal{G}_c$, one can conclude (global) connectivity of the latter if there exists a routing path from any node of the network to any other, a.k.a. end-to-end connectivity. This can also be expressed in algebraic terms via examining positiveness of the second smallest eigenvalue of the Laplacian matrix that corresponds to the aforementioned graph. For more information on algebraic graph theory the reader is encouraged to refer to [18]. The main issue, however, in controlling network connectivity via the graph's Laplacian is the fact that it is a centralized approach and thus inapplicable in cases of communication range constraints imposed by the nodes' antennas physical characteristics.

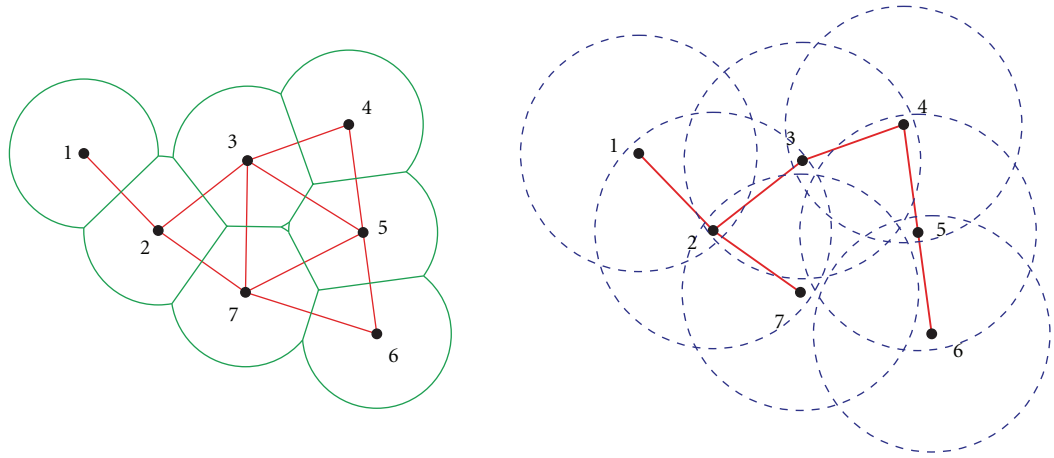
The motivation for introducing N-hop connectivity term among the nodes (Definition 3) is the fact that motion is performed in discrete time. Thus, from a practical point of view, one can assume that between two consequent motion time steps k and $k+1$, the nodes perform a transmit/receive action N times. That way, each node is informed at each time step k for the positions of the nodes that it is N-hops connected with, or equivalently, its N-hop neighbours in the $\mathcal{G}_c$ graph. Hence, distributed approaches can be developed that are based on the aforementioned set for the node to move at each step, without requiring global knowledge of the state of the network.

Assumption 1. Initially, each node is ℓ-hops connected with all its $2r$-limited Delaunay neighbours, where $\ell \in I_N$ for any *a priori* given $N \in I_{n-1}$, that is,

$$i \sim_{\ell} j, \quad j \in \mathcal{N}_i^{2r}, \ \ell \leq N \leq n-1, \ i \in I_n. \tag{9}$$

The aforementioned assumption becomes clearer via Figure 2, presenting a sensor network of 7 nodes with ratio R/r equal to 1.6. The network's $2r$-limited Delaunay graph along with the sensed domain is presented in Figure 2(a), while the communication graph along with the radii R is depicted in Figure 2(b). The two parts, sensing and communication ones, have been separated in order to avoid confusion, while they refer to the same network. Examining nodes 6 and 7, one can verify via $\mathcal{G}_D^{2r}$ that $6 \in \mathcal{N}_7^{2r}$ and vice versa. However, as concluded by the communication graph $\mathcal{G}_c$ of the network, these nodes are not directly connected but are 5-hops connected, instead, via the path $\{7, 2, 3, 4, 5, 6\}$, that is, $6 \sim_5 7$.

In fact, it is apparent that if $R \geq 2r$, then each node is directly connected (from a communication aspect) with all



(a) (b)

Figure 2: Graphical example in order to associate $\mathcal{G}_D^{2r}$ (a) and $\mathcal{G}_c$ (b) graphs.

of its $2r$-limited Delaunay neighbours, that is, $j \in \mathcal{N}_i^{2r} \Rightarrow j \sim i$, in general. The scope of the following section is the development of a distributed control law, such that the network is led in an area-optimal configuration, given communication range constraint R, and at most N-hop connectivity demand over the $2r$-limited Delaunay sets.

4. Connectivity-Aware Coordination Scheme

Recalling $\mathcal{H}$ as defined in (7), the primary objective of the nodes in the network should be to self-deploy themselves in a way that the area of the sensed part of D (expressed as the summation of the areas of the corresponding r-limited Voronoi cells) is as high as possible, assuming bounded sensing domains. Furthermore, coordination at this stage should be performed in a distributed manner via local information, as exchanged among the nodes considering their communicational capabilities. As far as the coverage-part of the objective is concerned, let us first state the following lemma which will form the basis of the coordination stage.

Definition 4. A set of n distinct points in D is called an r-limited centroidal Voronoi configuration if and only if each point lies at the centroid of its own r-limited Voronoi cell.

Lemma 5 (see [23]). *The area covered by a set of nodes in an r-limited centroidal Voronoi configuration is locally maximum.*

In this paper, since coordination of the network is assumed to be performed in discrete time-steps, the nodes are selected to move towards the centroid of their corresponding r-limited Voronoi cells via fixed step sizes, until they reach optimum configuration where the total sensed area is maximum. More specifically, considering (1), the control law u_i for the corresponding node to move at each step can be selected as

$$u_i = \sigma \frac{\text{centr}(V_i^r) - x_i}{||\text{centr}(V_i^r) - x_i||}, \tag{10}$$

where the sample instance superscripts have been omitted to avoid notation complexity. In the previous expression, $\text{centr}(\cdot)$ stands for the centroid of the compact-set argument, while $\sigma = \min(\varepsilon, ||\text{centr}(V_i^r) - x_i||)$, for any arbitrarily small $\varepsilon > 0$.

It should be noted that monotonicity of $\mathcal{H}$ is not guaranteed during the transition of the network towards the r-limited centroidal Voronoi configuration; however, since one node moves at each time step (as stated in Section 2), the network will reach asymptotically area optimal configuration.

However, in order to characterize the control scheme (10) as decentralized, in the sense that the node to move does not require global network knowledge to apply it, the aforementioned node should be able to evaluate its r-limited Voronoi cell via information from the nodes in range. In most of previous works in the literature, this communication issue is surpassed by allowing the nodes' communication range to be at least twice the sensing one, since only the nodes in range of $2r$ are needed for distributed evaluation of V_i^r [1].

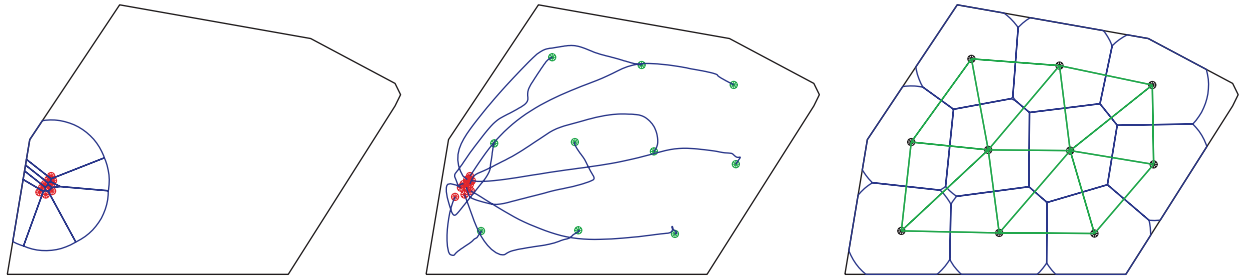
In this paper, however, an extra constraint is considered via limiting the communication range R to *any* arbitrary value, which in practice is imposed by the antennas' radio characteristics. The proposed control action is based mainly on preserving at most N-hop radio connectivity among the $2r$-limited Delaunay neighbours, where N is a fixed number of hops *a priori* defined and independent of the nodes' sensing radius, so that (10) is distributively evaluated.

Let us denote by i the node to move at an arbitrary time step k. The latter decides on making a move towards the centroid of its r-limited Voronoi cell, in order to increase network's coverage performance. However, node i is assumed to have spatial information acquired from the nodes that it is N-hops connected with. It is clear that if after the motion a new node joins the set of the nodes that i is connected with, then no issue arises, since this just provides an extra link in the $\mathcal{G}_c$ graph.

In the case, however, where a link is about to break if motion is performed, then the node to move checks if that

(1) [time-step k; node i is to move]
(2) identify N-hop neighbouring nodes
(3) evaluate V_i^r
(4) evaluate control law u_i
(5) estimate locally altered communication graph (with i at new position, but before performing the motion)
(6) **if** Assumption 1 does not hold (at new position) **and** bisection-max-depth not reached, **then**
(7) $\sigma \leftarrow \sigma/2$
(8) **go to** 4
(9) **end if**

Algorithm 1: Connectivity-aware control scheme.



(a) (b) (c)

Figure 3: Case Study I: coordination results derived via the proposed control scheme. (a) Initial network configuration. (b) Network evolution. The green (red) circles represent the nodes' final (initial) positions. (c) Final network state. Communication graph indicates 1-hop connectivity among the $2r$-limited Delaunay neighbours.

link was included in the shortest path in $\mathcal{G}_c$ to determine N-hop connectivity among any two nodes of the set of its radio-connected neighbours. If so, then motion is tried to be performed with half the step size σ, and so on, till a predefined search depth. In the case where neither of the examined cases is able to preserve connectivity among the aforementioned set of nodes, then the node stays idle and does not perform any motion. The previous procedure is summarized in Algorithm 1.

Although the control scheme described previously incorporates some kind of conservatism, it is still able to (i) increase network's coverage performance from one step to another, while (ii) preserving N-hop radio connectivity among the $2r$-limited Delaunay neighbours, as required for distributed evaluation of network's coverage increase. The fact that no motion is performed if a link that is about to break determines connectivity of the communication graph is the reason for the existence of overlapping among the nodes' sensory domains in the optimal state.

5. Simulation Results

Simulations were conducted in order to further verify efficiency of the proposed coordination scheme. The region D to be surveyed is considered as a convex compact planar set of total area $\int_D dS = 6.2015\,\text{units}^2$. The latter is identical to the one used in [25]. Two simulation studies are examined. In the first case, the network consists of $n = 10$ nodes with $r = 0.5$ units. The communication radius R was selected equal to $2r$, while the nodes were demanded to attain 1-hop network connectivity. In the second study, the network consists of $n = 20$ mobile nodes with sensing radii equal to $r = 0.2$ units. In order to emphasize in connectivity issues, the communication radius of the nodes' antennas was selected equal to r, $R = 0.2$ units, imposing a quite hard constraint in connectivity preservation during the coordination stage, while the nodes are *a priori* demanded to retain at most 3-hop connectivity.

The nodes are initially deployed randomly in D, so that Assumption 1 holds, for both cases. Given the set D, the maximum possible coverage ratio achieved that is evaluated as the summation of n circles (best case scenario) is equal to 100% and 40.5% of D, respectively, for each case. The network's initial configuration, evolution through time, and the final network's state, for the first case study are shown in Figure 3, in this order.

It is apparent that the network achieves optimum coverage, while the nodes retain radio connectivity with their $2r$-limited Delaunay neighbours. Figure 5(a) presents the network's coverage during evolution, where it is seen that almost maximum performance is attained.

As far as concerns the second scenario, $R = r$, the network's initial configuration, evolution, and the final network's state are presented in the top part of Figure 4, in this order.

Figure 5 depicts the evolution the normalized network's coverage, that is, $\mathcal{H}$ as a ratio of the area of D when

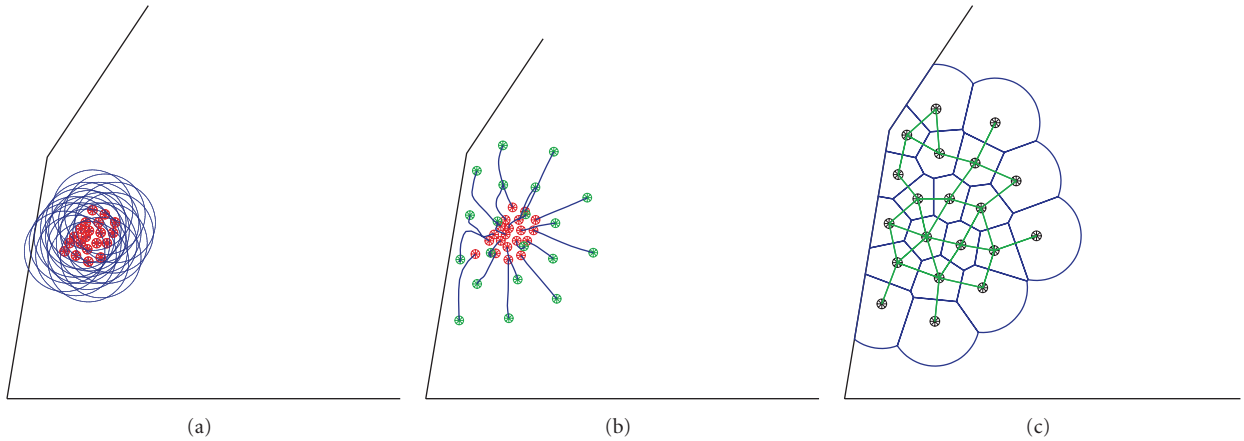


Figure 4: Case Study II: coordination results derived via the proposed control scheme. (a) Initial network configuration. (b) Network evolution. The green (red) circles represent the nodes' final (initial) positions. (c) Final network state. Communication graph indicates 3-hop connectivity among the $2r$-limited Delaunay neighbours.

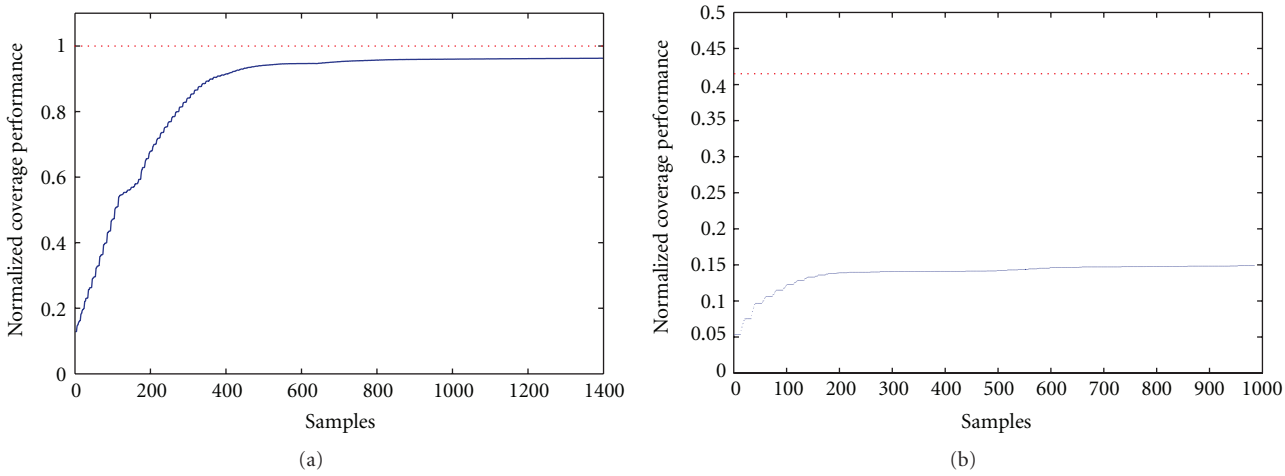


Figure 5: Percentage of sensed area during network evolution for Case Studies I (a) and II (b), respectively. The red upper-limit represents the maximum possible coverage ratio, in each case.

coordination scheme of Section 4 is applied (blue line). The red line represents the maximum possible coverage ratio, that is, 40.5%. More specifically, the sensed area percentage, starting from an initial value of 5% (dependent on the initial network configuration), increases as network evolves, until it converges to 14.8%, which is quite satisfactory considering communication constraints imposed.

One can see that the communication constraint imposition retains the network from achieving optimum area coverage, as evaluated via the best case scenario. However, the coverage performance attained is in fact quite satisfactory, taking into account that all nodes preserve 3-hop radio connectivity with all their $2r$-limited Delaunay neighbours.

6. Conclusions

In this paper, a spatially distributed approach was proposed for leading the mobile nodes of a sensor network towards an area optimal configuration, while simultaneously preserving radio connectivity among the nodes. The scheme was developed for the general case, where the radio range of the nodes' antennas can be less than twice the sensing range. Each node that moves is ensured to attain N-hop connectivity with the subset of nodes needed for decision making about the spot to move, so that the area coverage of the network is increased. Simulation studies were conducted verifying the efficiency of the proposed scheme.

References

[1] J. Cortés, S. Martínez, T. Karataş, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[2] J. Le Ny and G. J. Pappas, "Sensor based robot deployment algorithms," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC '10)*, pp. 5486–5492, Atlanta, Ga, USA, December 2010.

[3] A. Panousopoulou and A. Tzes, "On mobile agent positioning for wireless network reconfiguration," in *Proceedings of the 6th*

International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (Wiopt '08), pp. 108–115, Berlin, Germany, April 2008.

[4] A. Panousopoulou and A. Tzes, "RF-power overlapping control for connectivity awareness in wireless ad-hoc and sensor networks," in *Proceedings of the 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pp. 275–280, Annecy, France, 2010.

[5] M. M. Zavlanos, H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Hybrid control for connectivity preserving flocking," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2869–2875, 2009.

[6] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[7] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1416–1428, 2008.

[8] M. Zavlanos, V. Preciado, A. Jadbabaie, and G. Pappas, "Spectral control of mobile robot networks," in *Proceedings of the American Control Conference (ACC '11)*, San Francisco, Calif, USA, 2011.

[9] M. M. Zavlanos, A. Tahbaz-Salehi, A. Jadbabaie, and G. J. Pappas, "Distributed topology control of dynamic networks," in *Proceedings of the American Control Conference (ACC '08)*, pp. 2660–2665, Seattle, Wash, USA, June 2008.

[10] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, ""Graph theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525–1540, 2011.

[11] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. S. Ravi, "Algorithmic aspects of topology control problems for ad hoc networks," *Mobile Networks and Applications*, vol. 10, no. 1, pp. 19–34, 2005.

[12] J. Luo and J. P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *24th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1735–1746, March 2005.

[13] N. Burri, P. von Rickenbach, R. Wattenhofer, and Y. Weber, "Topology control made practical: increasing the performance of source routing," in *Proceedings of the 2nd International Conference on Mobile Ad-hoc and Sensor Networks*, Hong Kong, 2006.

[14] X. Y. Li, P. J. Wan, and O. Frieder, "Coverage in wireless ad hoc sensor networks," *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 753–763, 2003.

[15] Z. Han, D. Niyato, W. Saad, T. Basar, and A. Hjorungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*, Cambridge University Press, 2011.

[16] T. Basar and G. Olsder, *Dynamic Noncooperative Game Theory*, Academic Press, London, UK, 1995.

[17] F. Knorn, R. Stanojevic, M. Corless, and R. Shorten, "A framework for decentralised feedback connectivity control with application to sensor networks," *International Journal of Control*, vol. 82, no. 11, pp. 2095–2114, 2009.

[18] C. Godsil and G. Royle, *Algebraic Graph Theory*, Springer, New York, NY, USA, 2001.

[19] Y. Stergiopoulos and A. Tzes, "Decentralized swarm coordination: a combined coverage/connectivity approach," *Journal of Intelligent and Robotic Systems*, vol. 64, no. 3-4, pp. 603–623, 2011.

[20] J. Stergiopoulos and A. Tzes, "Decentralized communication range adjustment issues in multi-agent mobile networks," in *Proceedings of the American Control Conference (ACC '10)*, pp. 1629–1634, Baltimore, Md, USA, July 2010.

[21] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM: Control, Optimization and Calculus of Variation*, no. 11, pp. 691–719, 2005.

[22] S. Martínez, J. Cortés, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75–88, 2007.

[23] F. Bullo, J. Cortés, and S. Martinez, *Distributed Control of Robotic Networks*, Princeton University Press, 2009.

[24] F. Aurenhammer and R. Klein, "ch. 5: voronoi diagrams," in *Handbook of Computational Geometry*, pp. 201–290, Elsevier, 1999.

[25] J. Cortés and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 934240, 12 pages
doi:10.1155/2012/934240

Research Article

Remaining Energy-Level-Based Transmission Power Control for Energy-Harvesting WSNs

Guojun Dai, Jian Qiu, Peng Liu, Bing Lin, and Song Zhang

Institute of Computer Application Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Correspondence should be addressed to Jian Qiu, qiujianhdu@gmail.com

Received 24 November 2011; Accepted 18 March 2012

Academic Editor: Jiming Chen

The purpose of this paper is to introduce a transmission power control scheme based on the remaining energy level and the energy-harvesting status of individual sensor nodes to extend the overall lifetime of wireless sensor networks (WSNs) and balance the energy usage. Ambient energy harvesting has been introduced as a promising technique to solve the energy constraint problem of WSNs. However, considering the tiny equipment and the inherent low and unbalanced harvesting capability due to environmental issues, there is still a long distance from perfectly solving the problem. In this paper, a wind and solar power joint-harvested WSN system has been demonstrated, which uses ultracapacitor as energy storage. By analyzing the power recharging, leakage, and energy consumption rate, a novel energy-level-based transmission power control scheme (EL-TPC) is produced. In EL-TPC scheme, the transmission power is classified into various levels according to the remaining energy level. By adapting the nodes' operation pattern, hierarchical network architecture can be formed, which prioritizes the use of high energy level, fast charging nodes to save the energy of uncharged nodes. The simulation and demonstration results show that EL-TPC scheme can significantly balance the energy consumption and extend the entire network lifetime.

1. Introduction

Energy constraint has been one of the most important design issues of wireless sensor networks (WSNs) since last decade. For lifetime maintaining, many mechanisms have been produced either to minimize the energy usage or to expand the energy storage. Generally, these schemes can be classified into two main categories: energy management and energy harvesting [1].

It has been extensively studied that most of the energy is consumed by the radio transceiver, which indicates that an effective way is to reduce the time period when the onboard radio is on and minimize the amount of packet delivery. Therefore, sleep-wakeup duty cycle is introduced to allow sensor nodes to sleep when there are no operation requirements. In order to reduce the entire traffic load and achieve long sleep duration, energy-efficient MAC protocols [2], routing protocols [3], data aggregation schemes [4], and transmission power control schemes have been proposed as well as some cross-layer design [5].

However, these protocols still face some primary problems, such as the trade-off between scheme complexity and node simplicity, the trade-off between idle listening, overhearing, and control packet overhead in MAC protocols, the trade-off between energy-efficient routes and long latency in routing protocols, and also the trade-off between average number of hops and transmission power level as well as energy consumption balance problem. It is unavoidable that the sensor nodes that closer to the sink node have heavier traffic load than those farther away due to more routing relay. The sensor nodes that take more local computation, such as schedule assignment and broadcast, data aggregation, will also consume more energy. After a certain period of system operation, these nodes in critical positions may lose effect and result in the entire network invalid.

On the other hand, energy harvesting has now been introduced as a promising solution of energy constraint which allows the batteries of sensor nodes to be recharged using ambient energy resources, such as solar power, wind power, or even vibration power. In these designs, ambient

energy can be converted to electrical energy and used directly or stored in the means of energy storages, for example, batteries and ultracapacitor. However, using rechargeable sensor nodes also suffers from many problems, such as the technical challenges in tiny ambient energy collection device production and implementation. The extremely low recharging speed due to typical ultra-low-power ambient energy and the unbalanced recharging speed also generate problems. The lifetime of energy storage component is also a design challenge under the situation of frequently recharging and consuming.

The purpose of this paper is to solve the problem of the unbalanced energy consumption and harvesting speed in ambient powered WSNs, which applies a transmission power control scheme to enhance or reduce the communication distance of sensor nodes based on the energy level situation. By applying such a scheme, the sensor nodes with higher remaining energy resource or higher energy harvesting capability will take more responses to the network data packet delivery, while the other nodes will be in an idle state for longer time.

Our previous work has been presented in [6]; a building surface was mounted, wind power was collected, and wireless sensor network system has been demonstrated, which aims to monitors the usage pattern of air conditioners (ACs), the outdoor temperature. The idea of energy-level-based transmission power control scheme (EL-TPC) was firstly introduced in [6] as well. It achieves energy saving and balancing by modifying the transmission power level on different nodes according to the remaining energy level, power recharging, and leakage speed of the nodes. This paper is extended from [6]; more clear descriptions and more details of EL-TPC will be provided. Theoretical analyses are made, and the corresponding network simulations are also extended to highlight the outstanding performance of EL-TPC. Solar energy-harvesting sensor nodes are also introduced in the demonstration system.

The remaining of this paper is organized as follows. In Section 2, the related works of energy harvesting and the corresponding management schemes are introduced and compared. Our previous work of EL-TPC is also briefly introduced. The demonstration WSNs architecture and hardware component design are described in Section 3 with the measured experiment results of the device parameters. The transmission power control scheme EL-TPC is introduced in Section 4 in details. In the following Section 5, the network simulation and real BSMSN system demonstration are set up and carried out; the performances are evaluated. Finally, the conclusion is given in Section 6.

2. Related Works

Although the energy-harvesting WSNs are not mature in techniques, they indeed point out a sufficient way of the development of future WSNs. Some energy management schemes have also been proposed to schedule the energy usage pattern based on this background. Transmission power control has been carried out to coordinate MAC protocols previously. These methods as well as our original EL-TPC scheme are summarized as follows.

2.1. Existing Energy-Harvesting WSNs Systems. Many examples of energy-harvesting WSNs systems have been demonstrated since 2005, such as Trio [7], Fleck [8], Prometheus [9], AmbiMax [10], Everlast [11], and TwinStar [12]. In these systems, the resource of the ambient energy can either be solar power, wind power, or even mechanical vibration. Once the energy is collected by the sensor node, it will be stored either in NiMH batteries, Lithium batteries, ultracapacitors, or even directly supply the sensor nodes.

As a clean and safe ambient energy resource, solar power harvesting has been studied for a long time. However, as the power collecting board is relatively small which should fit the size of sensor nodes, the recharging rate of solar batteries is very slow. Another limitation of using solar power is that it cannot work during the night time or rainy days, which makes the network dependent on the weather. Wind power has also been well studied. Typically, it is not very stable when it is strongly constrained by the climate and the location of sensor nodes. Unless there is a constant wind resource, the performance is even worse than the solar power.

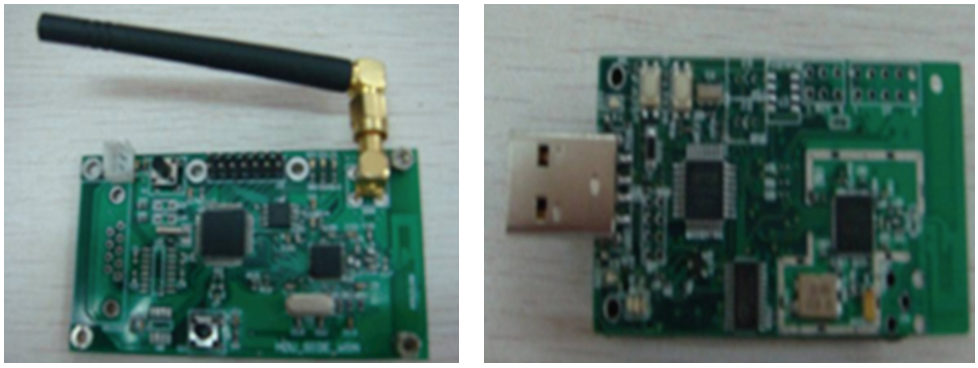
Compared with rechargeable batteries, the advantage of using ultracapacitor in WSNs is significant with the feature of fast recharging, unlimited recharging times, and invulnerable [12]. However, the leakage causes a big problem that the energy cannot be stored for long.

2.2. Energy Management Schemes in Energy-Harvesting WSNs. After designing and demonstrating a prototyping energy-harvesting WSN, how to use and store the harvested energy consists of more research topics. The efficiency of the management schemes significantly determines the performance of energy-harvesting WSNs. Recently, many energy management schemes have been introduced for better usage of harvested energy.

A solar energy-harvesting system has been introduced in [13] as well as a corresponding energy-neutral mode, which enables the nodes to use as much energy as it gains from the environment and results in the energy input and output balance in any individual node. In [14], a joint energy management and resource allocation scheme is proposed to schedule the energy usage by adapting the sampling rate, while in [15], the authors propose a novel cost metric for the randomized minimum path recovery time (R-MPRT) routing protocol which can achieve better performance than normal routing protocols considered in energy-harvesting WSNs.

These schemes can help to improve the usage efficiency of harvested energy in a certain level and particular WSNs. However, due to the deployment area and environmental situations, the energy-harvesting speed is unbalanced, which brings more challenges in management scheme design.

2.3. Transmission Power Control Protocols. Transmission power control (TPC) techniques have been proposed as a coordination of routing or MAC protocols, which enables nodes to have dynamic transmission range and thus results in multiple coverage areas [16, 17]. In [18], two TPC schemes



(a) (b)

Figure 1: HDU mote sensor and sink node.

are proposed to coordinate BMAC, which achieves up to 57% energy saving in their simulation. An experimental-based TPC research has been represented and evaluated in [19], which shows that the real performance of TPC schemes is not as well as simulated, but their dynamic TPC scheme can perform up to 37% energy saving in 10% duty cycle.

A traditional trade-off in TPC schemes is whether it is better to make larger transmission range which can reduce the number of hops in the routes or will it be more energy efficient to use smaller transmission power which reduces the energy consumption of each transmission and reception. Another problem of dynamic transmission power is that it is harmful to energy balance, which consumes the energy in the nodes with larger transmission power much faster than those using low power.

2.4. Previous Work of EL-TPC. As mentioned in previous subsections, both energy harvesting and energy consumption are unbalanced. Independently, they are both harmful in WSNs. But if they can be combined, the drawbacks can be converted to optimizing features. This also generates our original idea of EL-TPC [6]. By joint consideration of the remaining energy level, the energy-harvesting speed, sensor nodes are grouped, and different transmission power levels are assigned to each group. According to the given transmission power level, a specified radio connection topology is provided different from the location-based topology. The detailed description will be provided in the following sections.

3. System Architecture and Hardware Model

To verify all the proposed work, network simulation and real system implementation are both useful methods. In our earlier work, a project called Cyber IVY has been carried out which mainly supplies building surveillance functions. A building surface mounted sensor network (BSMSN) system has been implemented and published in [20]. However, at that stage, the energy resource of the sensor nodes was normal rechargeable batteries; simple broadcasting-based communication was applied without any channel and energy management. In the original work of EL-TPC [6], Cyber IVY has moved to a new stage by using wind powered sensor nodes, which is called building surface wind-power collected wireless sensor network (BSWPWSN) system. Recently, solar powered sensor nodes are also used as a sufficient support. The detailed design issues and system features are discussed as follows.

3.1. Application Study. The application potential of Cyber IVY is still being explored, which will shortly combine air pollution and noise monitoring in near future. The sensor boards with more functions are being developed. At the recent stage, the recent system aims to monitor the usage pattern of air conditioners (ACs), the outdoor temperature, and day light lamination information. The main purpose of doing these monitoring tasks is to provide some sensible information for environment protection and electrical energy saving. It is reported that the air conditioners consume huge amount of electrical energy during summer and winter time, especially in public buildings such as laboratory building in the universities or shopping malls. As in public buildings, privacy will not be a problem. By monitoring the AC usage pattern, it is easy to evaluate the total amount of energy cost. With the coordination of outdoor and indoor temperature information, unusual usage of AC will also be obtained and can help in further energy saving, which has been demonstrated and presented in [20].

In our plan, the demonstrated BSMSN is a prototype of a certain set of simple designed, multifunctional, self-powered, long lifetime WSNs. Scientific problems can be formed, and design challenges can be detected along with the progress. The possible solutions will also be proposed, developed, implemented, and evaluated.

3.2. Hardware Architecture and Network Topology. HDU mote (Figure 1) node is used in BSMSN with its Chipcon CC2420 transceiver component. A large choice of onboard sensors is also provided, such as CO_2 density, temperature, humidity, and light lamination. CC2420 can work on 31 different power levels and 14 orthogonal radio channels. Depending on the communication requirements, the transmission range can be adjusted by changing the power level.

Figure 2: Illustration of BSMSN system implementation.

In BSMSN, sensor nodes are densely deployed on the surface of external unit of AC, which covers the outside of the fan. This results in an advantage that the fan of the AC external unit can provide relatively stable wind power output to provide energy supply to sensor nodes when they are working. Figure 2 represents an example case of typical building surface and our sensor network deployment.

The transmission power of CC2420 transceiver can be adjusted with 31 levels, from level 0 to level 30. In our measurement results, if a node is operated with full-power state, all the nodes within a 150-meter range will be its one hop neighbor if the nodes are deployed in a pain floor. As the distance between two AC external units is relatively short in terms of 5–10 meters, a large number of ACs will be covered by a certain one as shown in Figure 3(a). In a typical hop-based packet collision avoidance MAC scheme, to guarantee a certain transmission to be successful, all the nodes other than the transmitter and receiver within such a large range have to be silent. In this case, very few nodes can be scheduled to transmit simultaneously, which causes low-efficiency problem. In another case shown in Figure 3(c), the transmission range is reduced to an unacceptable level in which no data delivery can be successful. In common knowledge or experience, the case shown in Figure 3(b) is much more reasonable than the other. The rest of this section will determine how to find a proper transmission range in our specific scenario.

3.3. Energy-Harvesting Component Design. In BSMSN system, wind power is the source of energy harvesting. As shown in Figure 4, the harvesting component is composed of four small fans facing the same direction. When the air conditioner is switched on, these small fans will follow the rotation of the fun of external unit. As the wind speed and strength of different area of the fan are not fixed, by trying various distances between four small fans and measuring the rotation speed of them, an approximately fastest rotation speed has been achieved. The simple generator behind the small fans will convert the wind power to electrical power.

Due to the feature of fast recharging and being invulnerable, Maxwell PC10 ultracapacitor is chosen to be the only power storage and resource of our sensor nodes. The rated voltage of PC10 is 2.5 volt, and the capacitance is 10 F. It can be recharged when the fans are rotating and

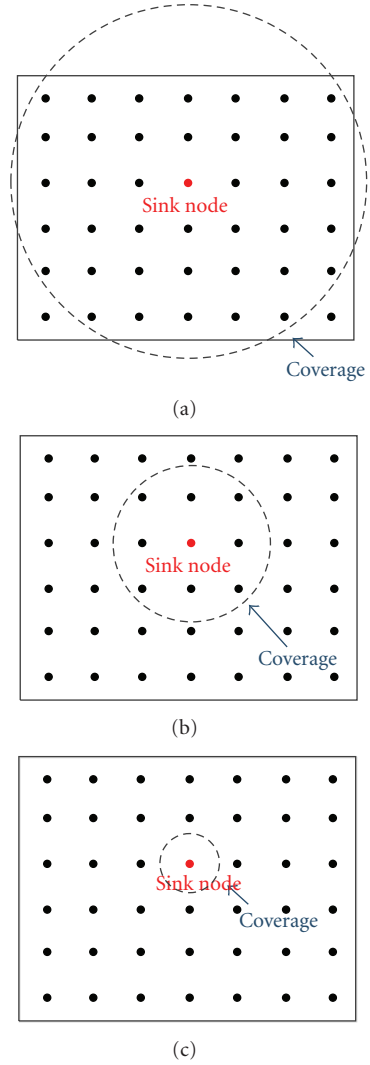


Figure 3: Illustration of logical network topology.

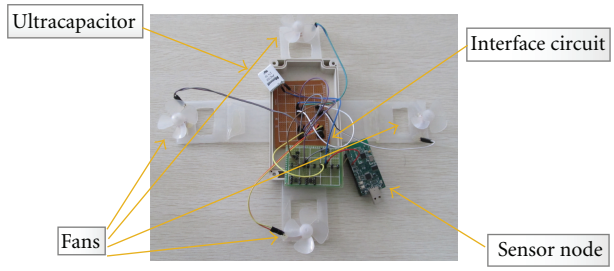


Figure 4: Illustration of wind energy-harvesting component.

supply sufficient energy for the nodes to maintain sensing and communication functions at the main time.

In our later work carried out in summer, solar power is also collected for sensor node recharging and operation. As shown in Figure 5, solar panel is used for energy harvesting, while the other components are the same as that of wind powered nodes. The energy collected is also stored in

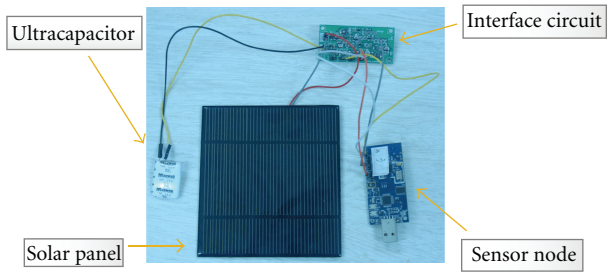


Figure 5: Illustration of solar energy-harvesting component.

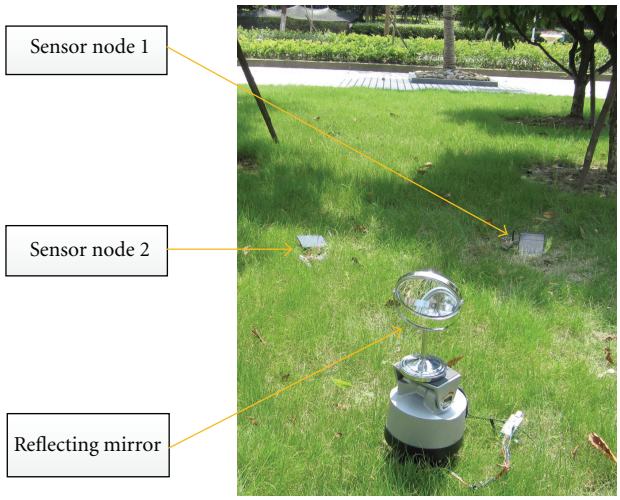
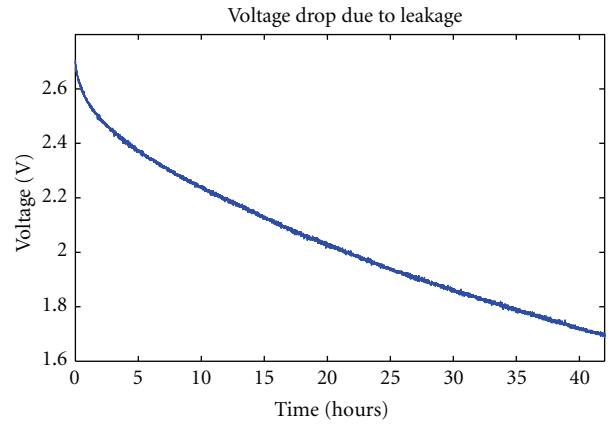
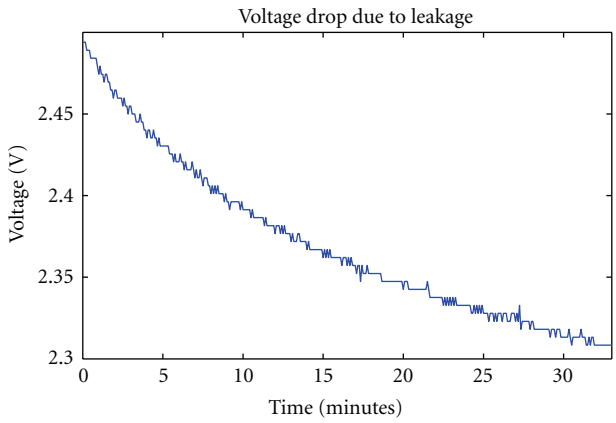


Figure 6: Illustration of solar powered sensor node deployment.



(a)



(b)

Figure 7: Long-term and short-term ultracapacitor leakage.

ultracapacitors, and the harvesting speed will be shown in Figure 8.

In our demonstration, the solar-powered sensor nodes are deployed between buildings as communication relays as well as environmental information sensing. To enhance the capability of solar energy harvesting and provide sufficient amount of energy supply in heavy traffic load situation, reflecting mirror is used for extra sunlight collection, which is shown in Figure 6. By this approach, the energy-harvesting speed is increased by approximately 50% in the demonstration.

3.4. Energy Recharging and Consumption Status. To know the status of energy recharging and consumption and the unbalanced situation which directs the design of the proposed EL-TPC scheme is very important. A set of experiments and measurements on related components and parameters have been carried out under various conditions.

As a natural feature of ultra-capacitor, leakage not only reduces the speed of charging, but also restricts the maximum duration of energy storage. The electrical energy will be lost even when the connected devices are powered off. The leakage in terms of voltage along with time is shown in Figure 7, in which Maxwell PC10 ultra-capacitor is used. The ultra-capacitor is firstly fully charged to 2.6 V. After approximately 45 hours, the voltage drops to an invalid level when the sensor node can no longer sufficiently work.

From Figure 7(a), it can also obtain that the leakage problem is more serious when the remaining energy level is higher. In Figure 7(b), the voltage reduction against time between 2.5 V to 2.3 V is illustrated, which shows that the stored energy is lost very fast in such high-voltage situation.

The curve of ultra-capacitor recharging is presented in Figure 8 under wind and solar power under various setups. It shows that with voltage increasing, the recharging speed will be slightly reduced. This is because ΔQ is proportional to ΔU^2 according to equation $Q = CU^2/2$. As the average recharging speed $\Delta Q/\Delta t$ is a constant value, the increasing of voltage $\Delta U/\Delta t$ will be reduced when U increases. In our special hardware devices and scenarios, as the measurement is carried out at noon in summer time, the solar power harvesting speed is faster than that of wind power, especially with direct sunlight combined with concave mirror reflection. However, solar power collection is not stable as the sunlight lamination varies significantly during the day. During the nights and rainy days, it will even lose effect. The wind power collection in our system is much more stable. When the mounted AC is powered on, the fan

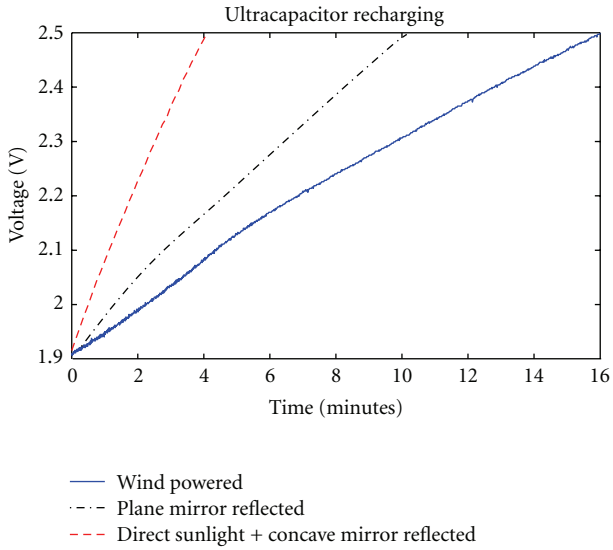


Figure 8: Energy-harvesting speed.

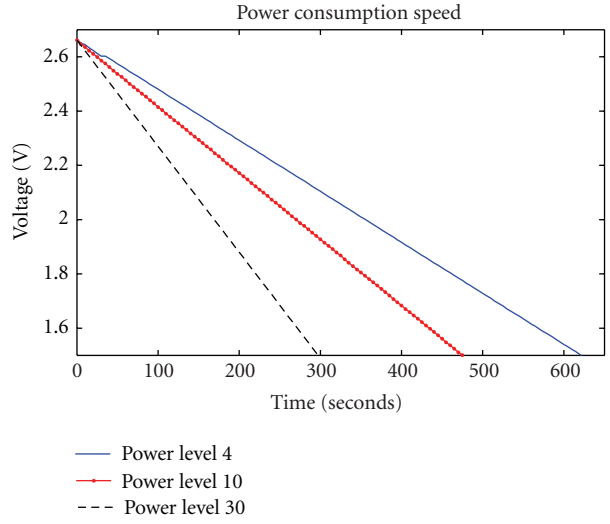


Figure 9: Power consumption of a sensor node.

in the external unit will supply constant wind power to our prototyping sensor node.

The energy consumption status is shown in Figure 9. In these measurements, the sensor node keeps the transceiver on all the time and broadcasts a "hello" message every second. The power level of the sensor node is set to be levels 4, 10, and 30, which can provide a radio-effective coverage for approximately 12 meters, 30 meters, and 100 meters, respectively. After 10.5, 8, and 4 minutes, the power level will be too low for the nodes to work, which is around 1.5 volts. The reason for choosing these three levels in the experiment is particularly due to the real application scenario as shown in the latter section, which specifies the coverage distance of the nodes. In different deployment environment and scenario, this can be varied.

Above all, it can be obtained that the sensor nodes consume energy a bit faster than the recharging speed. However, by considering a reasonable duty cycle which enables the sensor node to switch the radio transceiver off most of the time, it can achieve a balanced state in theory. The leakage is ignorable in most cases. But if a node resumes high energy level, it is a good choice to keep high-duty cycle. Otherwise, the energy will be lost rapidly. The remaining of this paper will focus on how to use transmission power control to make a negotiation considering all the above issues.

4. EL-TPC Scheme Description

According to the energy-harvesting design of sensor nodes, it is believed that the sensor node will no longer permanently lose effect due to power used up. The power-up nodes will go to a "long sleep" state when only recharging takes place.

In this section, a transmission power control scheme called EL-TPC will be introduced in detail on the purpose of balancing energy consumption to reduce the probability of the existence of "long sleep" state. In other words, the scheme will focus on how to keep most of the nodes in an "on" state to provide sufficient continuous service of WSNs. As mentioned in Section 2, the idea of EL-TPC was originally raised according to the unbalance of both energy consumption and harvesting. The fundamental goal is to make the nodes with more remaining energy and better harvesting capability to take more responses. With the development of our work, the scheme has been modified and is now wider applicable than that presented in [6].

4.1. Assumptions and System Parameters. There are many specified features of BSMSN including the system design and network topology. A list of assumptions and system parameters is summarized as follows.

Relatively Uniform Deployment. In BSMSN, the sensor nodes are mounted to external unit of ACs or other positions which have certain distance between each other. An example network topology is shown in Figure 2 where sensor nodes form a natural grid topology with a fixed distance between two neighboring nodes.

Time Perfect Synchronized. As synchronization is out of the scope of our EL-TPC scheme, in our experiment and simulation, sensor nodes are deployed after it is initialized and synchronized with the sink node. During the demonstration period, perfect synchronization is kept.

Packet Congestion Ignorable. There are 14 orthogonal radio channels available in CC2420 transceiver, and the transmission duration is relatively short in typical low sampling rate. By using a random back-off before each transmission, it is unlikely to have two nodes transmitting simultaneously and using the same channel.

Direct Connection Feasibility. In the demonstration of BSMSN system, the maximum diameter of the coverage area

is approximately 100 meters. Therefore, with power level 30, any two nodes can achieve direct connection.

Limited Energy Storage Capability. The leakage of the ultracapacitor makes the stored energy unable to provide the sensor nodes long time operation. It is necessary to increase the consumption rate when there is much energy left.

4.2. Multiple Transmission Range Choice. As mentioned in Section 2, a traditional trade-off in transmission power control schemes is between the low energy consumption rate in large quantity nodes and high energy consumption rate in low amount of nodes. It also limits the potential of TPC scheme in energy constraint WSNs as it may suffer from obvious energy unbalance.

In EL-TPC, this situation can be sorted out according to unbalanced energy-harvesting capability. Higher power and larger coverage can be applied in the nodes with more remaining energy and better harvesting capability, while the nodes short in energy can reduce the power level for energy saving. For example, three transmission power levels are determined in BSMSN system according to the measurement results as follows.

High Level (HL). Nodes use extremely high transmission power, (e.g., level 30 or a bit below in CC2420), which enables it to directly communicate with the sink node. In our demonstration and simulation, only the nodes with 2 volts or higher voltage and have sufficient power supply will work in this level.

Low Level (LL). Nodes without energy recharging reduce the power to a relatively low level by which it can only communicate with no more than one neighboring node in any direction. These nodes take very few sensing tasks and no communication relay functions. In the demonstration, the power level used is level 4 in CC2420.

Middle Level (ML). Nodes use medium power level, for example, level 10, which cover a hop distance approximately 20–30 meters and have two or three neighboring nodes on the main four directions (up, down, left, and right) within the transmission range. Any energy-harvesting node with a voltage lower than 2 volts will stay in this level.

In traditional TPC schemes, the main goal is to find out an optimal or near-optimal transmission power level which will be applied to all the sensor nodes. But in our EL-TPC, the nodes in the same network may use different power levels. Therefore, an interesting phenomenon occurs that the communication is not totally of dual direction. Node A may hear node B, while node B may not hear node A if the transmission power in node A is lower than in node B. To ensure that all the packets can be successfully forwarded to the sink node, handshake process is very important in scheme initialization and updating.

4.3. System Setup and Data Packet Delivery Strategies. Instead of routing and medium access control (MAC) protocols, a data packet delivery control mechanism has been proposed to coordinate the EL-TPC scheme. There are three packet delivery strategies described as follows, which inherently form the routes and schedule the channel access.

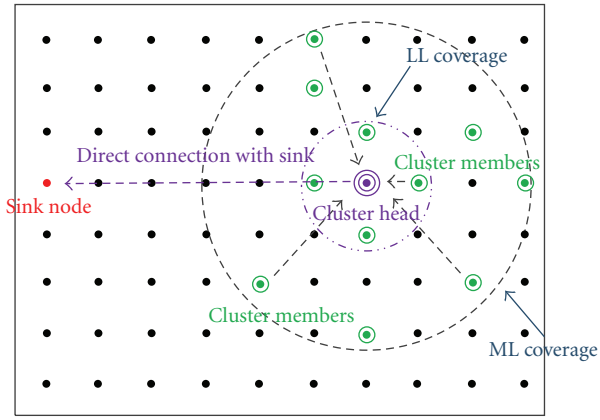


Figure 10: Illustration of cluster formulation.

The *first strategy* is direct or semidirect connection with the sink node. Once a node fulfills the requirements of *high level (HL)*, it will form a cluster by broadcasting an "invitation" packet to all the nodes in its coverage, which is the entire network. When *ML* and *LL* nodes receive this "invitation," they will return an acknowledgement addressed to this *HL* node after a random back-off. After receiving the ACK packet, the *HL* node will then reply with a "handshake" packet to the specific nodes to inform that they are cluster members.

However, as this ACK packet is transmitted in a lower power level, some of them may not be heard by the *HL* node. As shown in Figure 10, an *HL* node will form a cluster according to the connection status instead of geographic distances. During the data packet transmission, the sensed packet of the *HL* node will directly be received by the sink node and that from the cluster members will use the *HL* node as an intermediate, which forms semidirect connections to the sink node.

The *second strategy* is a multihop fashion. During the network initialization stage, if a node has not received any "handshake" packet, it will define itself as a noncluster member. Then, an "access request" packet will be sent to the nodes in its coverage by broadcasting. If there is any *middle level (ML)* neighbor who is a cluster member receives this packet, it will reply with an "access acceptance" packet. Then, a three-hop route is formed as shown in Figure 11, and this node will define itself as a cluster-access node.

If all the *ML* neighbors receive this "access request" are non-cluster members, there will be no "access acceptance" packet returned. After a timeout period, this node will broadcast a "routing request" packet. If there is any *ML* neighbor as a cluster-access node, a "routing acceptance" packet will be returned and a multihop route will be formed.

If there is no "routing acceptance" response returned after timeout, this node has to switch its radio transceiver off but keep ordinary sensing functions. When the voltage of

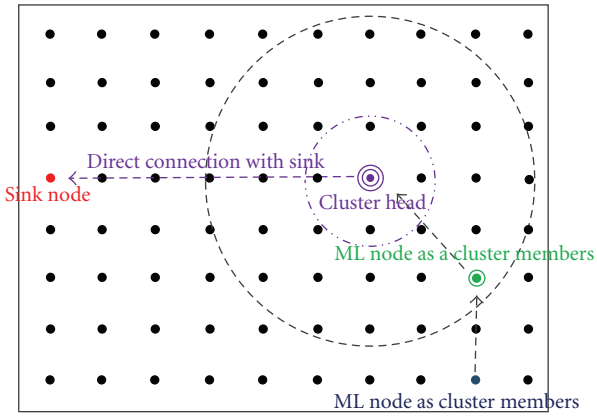


Figure 11: Three-hop routing formulation.

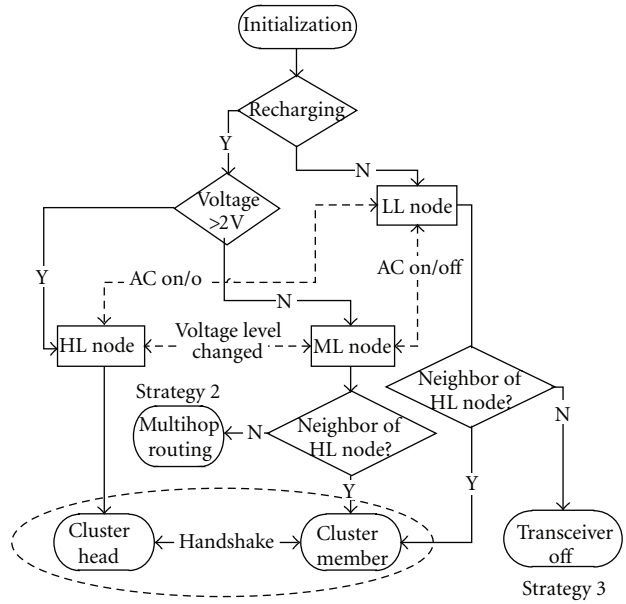


Figure 12: State machine of EL-TPC.

its onboard ultra-capacitor is recharged to above 2 volts, it will become an *HL* node and forms its cluster. The stored history data will then be directly transmitted to the sink node. Otherwise, it will wait until the next updating starts.

According to the above procedure, a maximum length of 4-hop route can be formed. In our small scope demonstration network, this is enough as it is very unlikely to leave any *ML* node to "off" state. But with the expansion of the network size, this thing will be changed. By allowing all the *ML* nodes to take the routing request process, longer routes will be available.

The *third strategy* is periodically waking up and updating the schedule information if there are still remaining nodes left in a *low level (LL)* transmission power state. In these nodes, the energy-harvesting component is not working, which indicates that the fans of the AC external unit are not rotating. Therefore, there is no necessity of AC operation monitoring; only other environmental information such as outdoor temperature will be sensed. To save energy, the *LL* node turns off its radio transceiver most of the time. It will periodically wake up and broadcast a "hello" packet every 10 minutes, which tells the neighboring nodes that it has no valid connection to the sink node. If there are any neighbors that become a cluster head, a "cluster invitation" packet will be returned addressing to this *LL* node, and a handshake procedure starts.

Above all, there are three main functions corresponding to the three power levels. Direct connection with the sink and cluster head functions are assigned to *HL* nodes, while *ML* nodes can either be a cluster member or a routing rely node. The *LL* nodes have no routing functions; it can either be a cluster member or wait to be a cluster member. Figure 12 represents the status of the entire procedure.

5. Simulation Setup and Performance Evaluation

In order to evaluate EL-TPC scheme and examine the parameter choice, a network simulation is carried out. In our simulator, the procedure is executed strictly following the scheme description. Except the given assumptions and assigned parameters in Section 4, a list of parameter setups are shown as follows.

Topology. Uniform grid topology as shown in Figure 10 and pseudorandom topology are both used in our simulation. In grid topology, the distance between each sensor nodes is 8 meters. In the pseudorandom topology, 80 sensor nodes are randomly deployed in a 50 m $\times$ 80 m area with a minimum distance of 5 meters between each other, which fits the real building situation and avoids redundancy in node deployment.

Energy Level. Energy level choice is essentially important in EL-TPC. Level 30 and level 4 in CC2420 are used for *HL* and *LL* nodes, respectively. In *ML* nodes, level 7 is mainly examined, which enables the nodes to cover 2 neighbors along the four main directions in the grid topology. An expanded energy level of 10 is also tested in some scenarios.

Sampling Rate. In our proposed BSMSN, the sampling rate is determined by the real monitoring requirements. Periodic sensing is set in each node with 20 seconds per packet for *HL* and *ML* nodes, and 100 seconds per packet for *LL* nodes. The reason is that the *LL* nodes only sense the environmental data, but the *ML* and *HL* nodes also sense the AC working status. To examine the performance of EL-TPC, we also show the results with 1/4, 1/2, 2, and 4 times of the traffic load.

AC Working Status. The feature of the sensing object also affects the performance of the proposed scheme. The on/off state of AC is not only the monitored event but also determines whether the sensor node can get sufficient energy supply. In our survey results, 80% of air conditioners are in

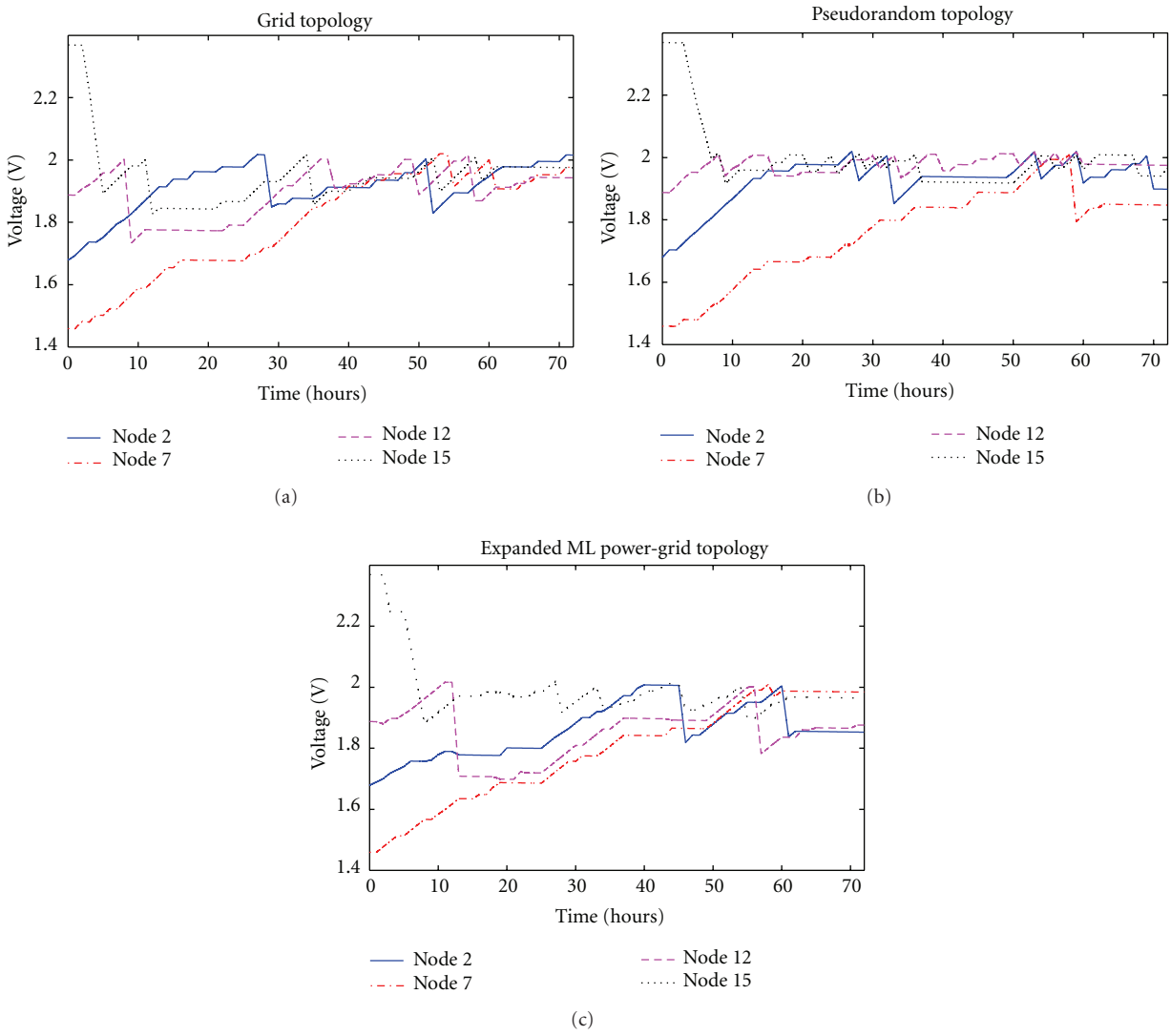


Figure 13: Average remaining energy in 4 different nodes.

an "on" state during the day time, while 20% of which are in an "off" state.

The most important result is represented in Figures 13 and 14, which shows the situation of remaining energy level in the nodes against time. Figure 13 clarifies that the nodes achieve a dynamic balance in energy consumption and harvesting along with the time under the given conditions by showing four nodes with different starting energy level. After approximately 10–30-hour operation, the average remaining energy in all the sensor nodes becomes very similar, and all of them keep a dynamic balance. After that duration, most of the nodes are in *ML* state or become *LL* state during the night. There are always certain *HL* nodes which can form clusters and achieve direct or semi-direct connections to the sink node. In grid and pseudorandom topologies, the scheme works relatively well. Figure 13(c) shows that by varying the *ML* nodes' energy level which changes the coverage of *ML* nodes, the energy performance is still stable.

Figure 14 provides the change of the remaining energy level of a randomly picked node along with time under different traffic loads. It shows that the average remaining energy level changes according the traffic load due the different transceiver energy consumption rate. If the traffic load is light enough, such as 25% of normal load, the energy level will increase continuously. With the increasing traffic load, the variation of the remaining energy level also increases, which means the node changes transmission power level frequently. It will finally break the balance as the traffic load is unaffordable for the node.

A natural feature of our proposed EL-TPC scheme is that there always exists certain isolated node which can neither be a cluster member nor achieve multihop relay. However, a node will not keep isolated for too long as the status of itself and the neighbors may change, which has been shown in Figures 13 and 14. In Figure 15, it is shown that the network also keeps a dynamic balance as the entire number of isolated

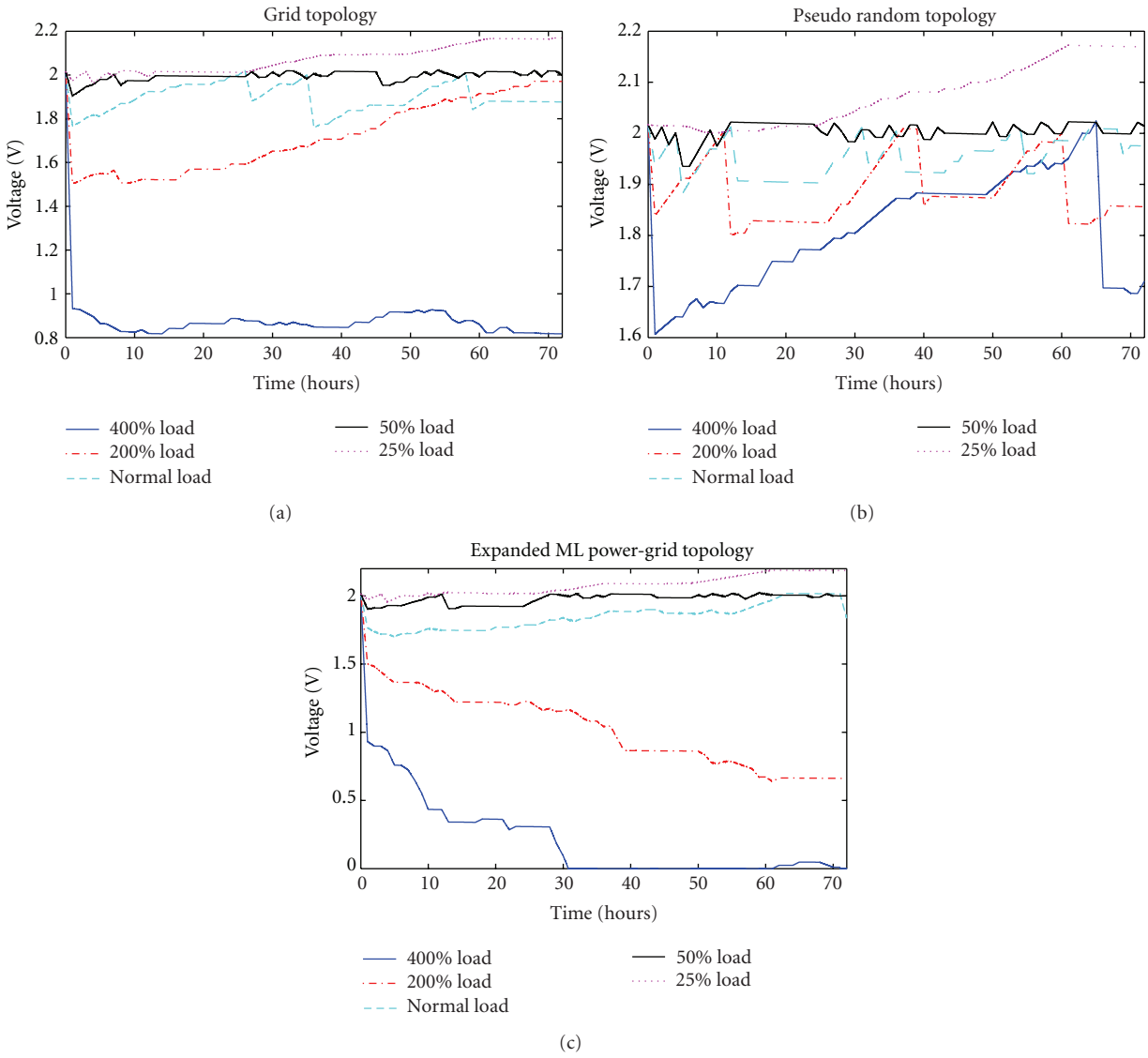


Figure 14: Average remaining energy against traffic load.

nodes will not be too large if the harvesting power resource keeps working. With the probability of "on" state of the AC increases, the amount of isolated node at any time decreases until zero. In our scenario, the system works stably if more than 50% of the air conditioners are switched on.

A demonstration of BSMSN system has also been built up and kept working for 3-day continuos monitoring. In this real implementation, uniform topology shown in Figure 2 is applied, and the nodes are synchronized before deployment. During the experiment, the system is shown to keep stable data packet output and displayed in the PC where the sink node is plugged on. The voltages of some sample nodes are real time monitored by adding the voltage information to the data packets.

Figure 16 represents the remaining energy level in terms of voltage against time for real implemented BSMSN system. By varying the traffic load and the average number of used air conditioner, the system works stably in which no nodes lose effect during the three days.

6. Conclusion

This paper has introduced a transmission power control scheme based on the remaining energy level and the energy-harvesting status of sensor nodes to maintain the lifetime of WSNs called EL-TPC. By achieving dynamic energy balance in sensor nodes and the entire WSNs, EL-TPC scheme has been proved to work excellently in energy-harvesting sensor networks, which keeps the WSNs robust and recoverable. The main contribution is that the unbalanced energy capability is used to solve the problem of unbalanced energy consumption. By applying EL-TPC scheme, the WSNs will no longer lose effect and keep high-efficiency operation. The simulation results compared the performance in different

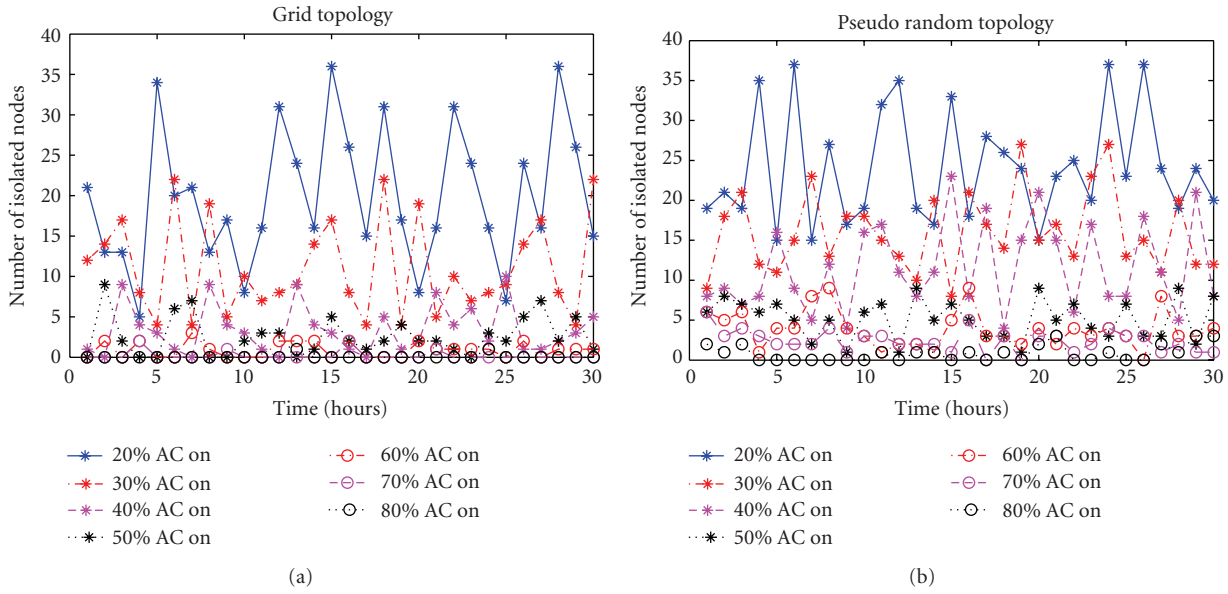


Figure 15: Isolated node identities against time.

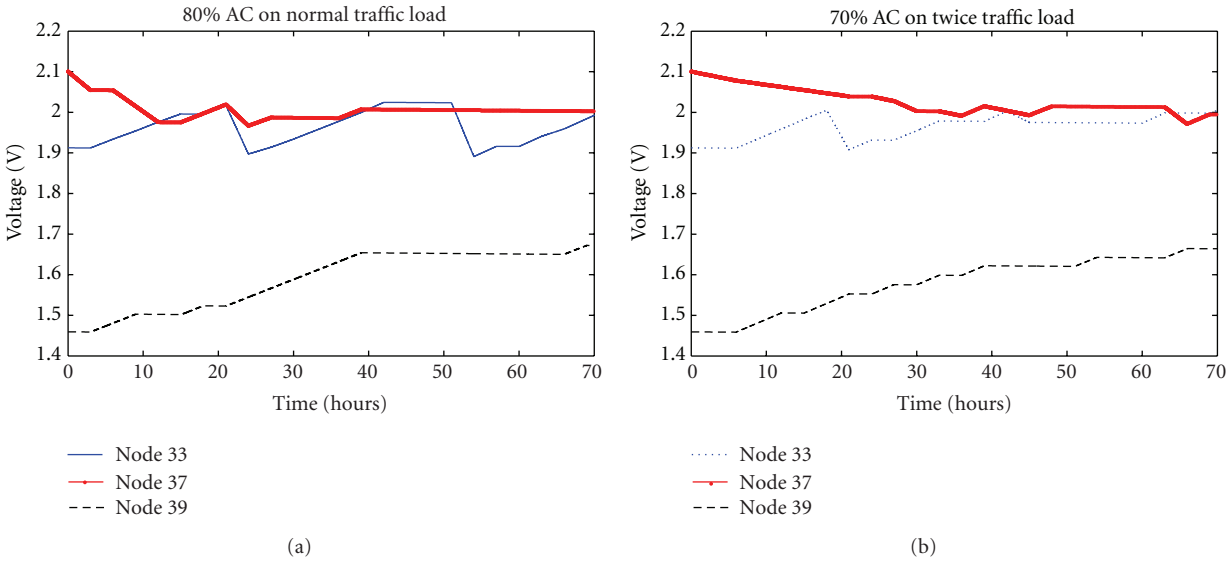


Figure 16: Average remaining energy illustration.

scenarios and showed the advantages and limitations of EL-TPC.

A demonstration of system call BSMSN has also been implemented, in which wind power is the only energy-harvesting resource, while energy is stored in and supplied by ultra-capacitors. It proves the idea of EL-TPC to be effective and sufficient during 3-day continuos working. This system can be used for air conditioner usage pattern surveillance and outdoor temperature monitoring at the recent stage, but it is scalable. More sensing functions can be extended to the system, and it will fit the requirement of more complex application environment deployment.

Acknowledgments

This work is supported by the Natural Science Foundation of Zhejiang Province under Grant no.Z1080979, Zhejiang Provincial Key Innovation Team on Sensor Networks under Grant no. 2009R50046-9, no. 2009R50046-1, no. 2009R50046-8, and no. 2009R50046-5, the Preresearch Scheme of National Basic Research Program of China (973 Program) under Grant no. 2010CB334707, the National Natural Science Foundation of China under Grant no. 60903167, and the Natural Science Foundation of Zhejiang Province no. Y1110831.

References

[1] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*, Wiley, Hoboken, NJ, USA, 2010.

[2] I. Demirkol, C. Ersoy, and F. Alagöz, "MAC protocols for wireless sensor networks: a survey," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115–121, 2006.

[3] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[4] J. N. Al-Karaki, R. Ul-Mustafa, and A. E. Kamal, "Data aggregation and routing in wireless sensor networks: optimal and heuristic algorithms," *Computer Networks*, vol. 53, no. 7, pp. 945–960, 2009.

[5] M. L. Sichitiu, "Cross-layer scheduling for power efficiency in wireless sensor networks," in *Proceedings of the IEEE 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 1740–1750, Hong Kong, China, March 2004.

[6] J. Qiu, B. Lin, P. Liu, S. Zhang, and G. Dai, "Energy level based transmission power control scheme for energy harvesting WSNs," in *Proceedings of the IEEE Global Communications Conference*, pp. 1–6, Houston, Tex, USA, December 2011.

[7] P. Dutta, J. Hui, J. Jeong et al., "Trio: enabling sustainable and scalable outdoor wireless sensor network deployments," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 407–415, Nashville, Tenn, USA, April 2006.

[8] P. Corke, P. Valencia, P. Sikka, T. Wark, and L. Overs, "Long-duration solar-powered wireless sensor networks," in *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets '07)*, pp. 33–37, Cork, Ireland, June 2007.

[9] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 463–468, Los Angeles, Calif, USA, April 2005.

[10] C. Park and P. H. Chou, "AmbiMax: autonomous energy harvesting platform for multi-supply wireless sensor nodes," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad hoc Communications and Networks (SECON '06)*, pp. 168–177, September 2006.

[11] F. Simjee and P. H. Chou, "Everlast: Long-life, supercapacitor-operated wireless sensor node," in *Proceedings of the 11th ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED '06)*, pp. 197–202, Tegernsee Germany, October 2006.

[12] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z. L. Zhang, "Leakage-aware energy synchronization for wireless sensor networks," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*, pp. 319–332, Kraków, Poland, June 2009.

[13] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastave, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 4, 2007.

[14] R. S. Liu, P. Sinha, and C. E. Koksal, "Joint energy management and resource allocation in rechargeable sensor networks," in *Proceedings of the IEEE (INFOCOM '10)*, Columbus, Ohio, USA, March 2010.

[15] D. Hasenfratz, A. Meier, C. Moser, J. J. Chen, and L. Thiele, "Analysis, comparison, and optimization of routing protocols for energy harvesting wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '10), IEEE International Workshop on Ubiquitous and Mobile Computing (UMC '10)*, pp. 19–26, June 2010.

[16] M. Cao, V. Raghunathan, S. Hanly, V. Sharma, and P. R. Kumar, "Power control and transmission scheduling for network utility maximization in wireless networks," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 5215–5221, December 2007.

[17] R. L. Cruz and A. V. Santhanam, "Optimal routing, link scheduling and power control in multi-hop wireless networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, vol. 1, pp. 702–711, April 2003.

[18] L. H. A. Correia, D. F. Macedo, A. L. dos Santos, A. A. F. Loureiro, and J. M. S. Nogueira, "Transmission power control techniques for wireless sensor networks," *Computer Networks*, vol. 51, no. 17, pp. 4765–4779, 2007.

[19] J. Jeong, D. Culler, and J. H. Oh, "Empirical analysis of transmission power control algorithms for wireless sensor networks," in *Proceedings of the 4th International Conference on Networked Sensing Systems (INSS '07)*, pp. 27–34, Braunschweig, Germany, June 2007.

[20] L. Peng, S. Zhang, J. Qiu, and G. Dai, "Building surface mounted wireless sensor network for air conditioner energy auditing," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM '11)*, pp. 738–743, Shanghai, China, April 2011.